BCL::SAS - Small Angle X-ray / Neutron Scattering Profiles to Assist Protein Structure Prediction

By

Daniel Kent Putnam

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Biomedical Informatics

May, 2016

Nashville, Tennessee

Approved:

Jens Meiler Ph.D
Loukas Petridis Ph.D
Douglas P. Hardin Ph.D
Martin Egli Ph.D
Thomas A. Lasko M.D., 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 and Department of Energy (DOE) SCGSR fellowship program. 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. They have encouraged, advised and led me along this path.

I specifically want to thank members of the Meiler Lab – Dr. Edward W. Lowe for his help with GPU acceleration, Axel Fischer for scientifically challenging my methods and conclusions. Further I would like to Loukas Petridis, John Eblen, Micholas Smith and Jeremy Smith for collaborating with me at Oak Ridge National Labs. 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.

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

APPENDIX

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

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
SANS	Small Angle Neutron Scattering
SAXS	Small Angle X-Ray Scattering
SSE	secondary structure element

CHAPTER I

RECONSTRUCTION OF SAXS PROFILES FROM PROTEIN STRUCTURES

Overview

The objective of my dissertation is to use small angle X-ray (SAXS) and Neutron scattering experimental data in conjunction with *in silico* structure prediction methods to produce protein models in agreement with the experimental restraints. To achieve this, I developed an algorithm, BCL::SAS to generate a replica of what an experimental SAS profile would look like from a rigid protein model. This algorithm can operate on complete protein models and models comprised of the backbone atoms of secondary structure elements (SSEs). This provides a method to compare models produced by BCL::Fold with experimental SAXS / SANS profiles.

Significance

I wrote a review article on how other groups have tackled the problem of SAXS profile reconstruction. This review article was praised as one of the best reviews on SAXS profile reconstruction by John Tainer and Robert Rambo. The reason this article was significant was because it was the first review in the field to explore multiple methods of SAXS profile construction and provide a history and rationale behind the methods. It succinctly explained the necessary theory and limitations of the proposed methods. This chapter is a reproduction of Reconstruction of SAXS Profiles written in 2013.[4].

Innovation

At the time of this writing, no other methods in the world were using the Debye method with GPU acceleration to reconstruction SAXS profiles from protein models.

Introduction

Small angle X-ray scattering (SAXS) is an experimental structural characterization method for rapid analysis of biological macromolecules in solution.[2, 3, 5-8] SAXS is inherently a low resolution method because samples move freely in solution during data acquisition resulting in spherically averaged scattering intensity curves. Because the samples do not need to be crystallized, they can be studied in different pH environments and concentrations leading to insightful structure-function relationships. 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. SAXS data has been used to filter a set of protein models by comparing the SAXS profile of each model with the experimental SAXS profiles [9, 10]. The SAXS profile has been incorporated as a term in the scoring function to obtain a protein model consistent with the experimental SAXS data [11]. An exciting feature in modern SAXS is identifying and modeling protein flexibility from an ensemble set of different conformers to fit experimental SAXS data [12, 13]. This requires a large library of starting conformers as input to the algorithm [14]. After a suitable library of conformers has been generated or found, the experimental SAXS data are used as a constraint in an algorithm to determine which combination of conformers optimally fit the data. The scattering intensity (I) is represented by a linear combination of the selected conformers. In this process the algorithm must decide 1) Which conformers to use and 2) How many conformers are required to accurately recreate the experimental SAXS profile. Critical to the success of this task are the underlying algorithms used to compute a SAXS profile from a proposed protein model. In this review we highlight different methods to accomplish this task. We recognize that these methods are not exhaustive of all methods, but represent a sampling of different approaches that provide insight to the process of computing SAXS profiles from atomic coordinates. For a more comprehensive review of small angle X-ray scattering theory we recommend several reviews [2, 3, 5, 15].

X-Ray Scattering Review

X-Ray scattering is observed when differences in electron density exist in a given sample and X-Rays generated from a source device pass through the sample. Although both coherent and incoherent scattering is possible, we will confine our considerations to coherent scattering because incoherent scattering is negligible weak at very small angles[2]. Elastic (without energy change) electron scattering is influenced by all atomic orbitals. Because atomic orbitals have different shapes according to their atomic group, the X-ray scattering provides information on the structure of the target sample.

The scattering process occurs as electrons resonate with the frequency of the X-rays passing through the object. As the electrons resonate, they emit coherent secondary waves which undergo both constructive and destructive interference. Because of destructive interference, the superposition of waves with all possible phases will lead to zero scattering at a scattering angle of 2θ [2]. The scattering maximum I(0) will be theoretically observed at a scattering angle of zero where all waves are in phase. Because of the high intensity of the incident X-Ray beam, a beam stop is placed between the detector and the beam to prevent it from distorting the scattering profile. I(0) must therefore be computed rather than experimentally observed.

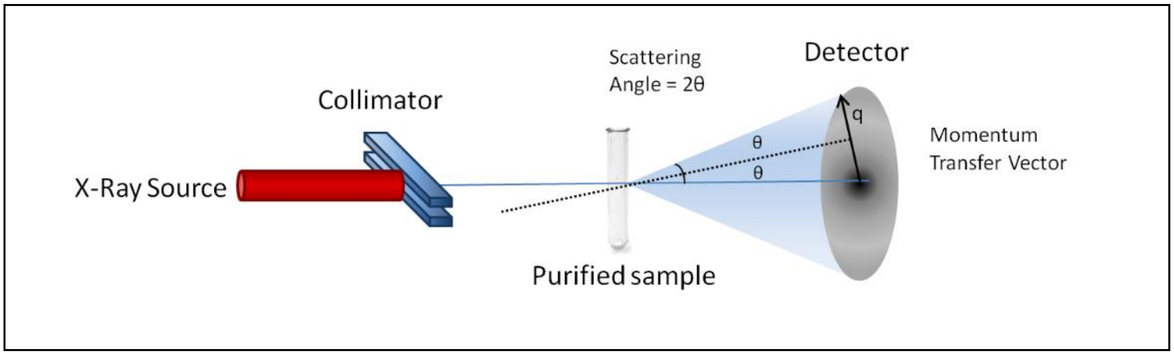


Figure 1: SAXS Experimental Setup. X-Rays with a constant wavelength λ are first focused by the collimator and then pass through the purified sample in solution. A small fraction of the X-Rays scatter as they encounter electrons in the sample. The detector captures these scattered X-Rays as intensity values. The final scattering profile is the difference between the profile of a blank buffer solution and a solution containing the purified sample**.**

To illustrate the scattering process, consider a linearly polarized monochromatic X-Ray beam incident on a single electron with charge e and mass m. The periodic electric field of the incident X-ray produces a force on the electron (**F**=q_e**E**) where **F** is the overall force the electron experiences, q_e is the charge of the electron and **E** is the electric field of the incident X-Ray. This force causes the electron to oscillate with the same frequency as the X-Ray. The equations governing this behavior are shown below beginning with the electric field equation:

$$\boldsymbol{E} = \boldsymbol{E_0} e^{i(\omega t - \delta)} \qquad 1.1$$

where **E** is the electric field, **E₀** is the maximum value of the electric field, ω is the frequency of oscillation of the wave-field, t is time, and δ is the phase constant. By Newton's second law of motion we equate the two equation of force:

$$\boldsymbol{F} = m\boldsymbol{a} = q_e\boldsymbol{E} = q_e\boldsymbol{E_0} e^{i(\omega t - \delta)} \qquad 1.2$$

Where m is the mass and **a** is the acceleration. The acceleration the electron experiences due to the periodic electric field is computed by dividing by the mass:

$$\boldsymbol{a} = \frac{q_e}{m}\boldsymbol{E_0}e^{i(\omega t-\delta)} = \boldsymbol{A_0}e^{i(\omega t-\delta)} \qquad 1.3$$

Where the amplitude **A₀** is:

$$\boldsymbol{A_0} = \frac{q_e}{m}\boldsymbol{E_0} \qquad 1.4$$

The electromagnetic radiation at a given distance with magnitude r from the charge q that experiences acceleration **a** has an electric field component:

$$\varepsilon = -\frac{q_e \boldsymbol{a}\sin\alpha}{c^2 r} \qquad 1.5$$

Where c is the speed of light, r is the magnitude of the position vector, q_e is the charge, a is the acceleration, and α is the angle between **a** and r. If the position of r is perpendicular to the incident beam (which is true for SAXS experiments) then α=90° and sinα = 1. Combining this simplification with the electric field component and substituting **A_0** for **a**:

$$\varepsilon = -\frac{q_e A_0}{c^2 r} = -\frac{q_e}{c^2 r}\frac{q_e}{m}\boldsymbol{E_0} = -\left(\frac{q_e^2}{mc^2}\right)\frac{\boldsymbol{E_0}}{r} \qquad 1.6$$

Now imagine instead of a single electron, we have an electron cloud. As incident X-rays pass through an electron cloud with the origin at the center, most of them travel through the cloud without scattering, while a small fraction (<1%) of the incident X-rays are scattered. This can be seen from the scattered to incident amplitude ratio:

$$\frac{\boldsymbol{\varepsilon}}{\boldsymbol{E_0}} = -\left(\frac{e^2}{mc^2}\right)\frac{1}{r} = -\frac{r_e}{r} \qquad 1.7$$

where e is the charge of an electron, r_e is the constant Thomson scattering length and r is the distance from the object to the detector.

$$r_e = \frac{e^2}{mc^2} = \frac{1}{4\pi\epsilon_0}\frac{q_e^2}{m_e c^2} = 2.818\ \times 10^{-15}\ m \tag{1.8}$$

Because r_e is small, the scattered-to-incident amplitude ration reveals that a single electron scatters a very small fraction of the incident X-Rays. For example, at a sample to detector distance of three meters (typical for SAXs experiments), the amplitude ration is:

$$\frac{r_e}{r} = \frac{2.818\ \times 10^{-15}\ m}{3m} \approx 10^{-15} \tag{1.9}$$

Table 1. Numerical values of critical constants in Thompson Scattering

Name		**Value**
q_e	Electron charge	1.602×10^{-19} C
m_e	Electron rest mass	9.107×10^{-31} kg
c	Speed of light	2.998×10^{8} m/s
ε_0	Permittivity of free space	8.854×10^{-12} $C^2/N{\cdot}m^2$

For a fuller description of the physics of X-ray scattering and the mathematics of waves we refer to the notes of Dr. Robert Blessing[16].

Because the scattered waves are coherent, the resulting amplitudes are added and the intensity is given by the absolute square of the amplitude[2].

$$A = \sum_{i=1}^{n} A_n\,, \qquad I = |A^2| \tag{1.10}$$

where A_n is the resulting amplitudes of all scattered waves and I is the scattering intensity. In Thompson elastic scattering all secondary waves have the same intensity and is given by:

$$I_s(\theta) = I_p \cdot \left(\frac{e^2}{mc^2}\right) \cdot \frac{1}{r^2} \cdot \frac{1 + \cos^2 2\theta}{2} \qquad 1.11$$

Where I_p is the primary intensity and I_s is the intensity of the secondary waves. The term e^2/mc^2 is the classical electron radius and r is the distance from the object to the detector. For small angles the polarization factor (1 _ $\cos^2 2\theta$)/2 is approximately one leaving:

$$I_s(\theta) = I_p \cdot \left(\frac{e^2}{mc^2}\right) \cdot \frac{1}{r^2} \qquad 1.12$$

The Momentum Transfer Vector

We will assume the amplitude and intensity of all secondary waves to be one for this discussion. With this framework, each secondary wave is represented by the complex function $e^{i\phi}$ where ϕ is the phase. Because the amplitude and intensity are one, all waves differ only by their phase. The phase of the scattered wave depends on the position of the oscillating electrons in space.

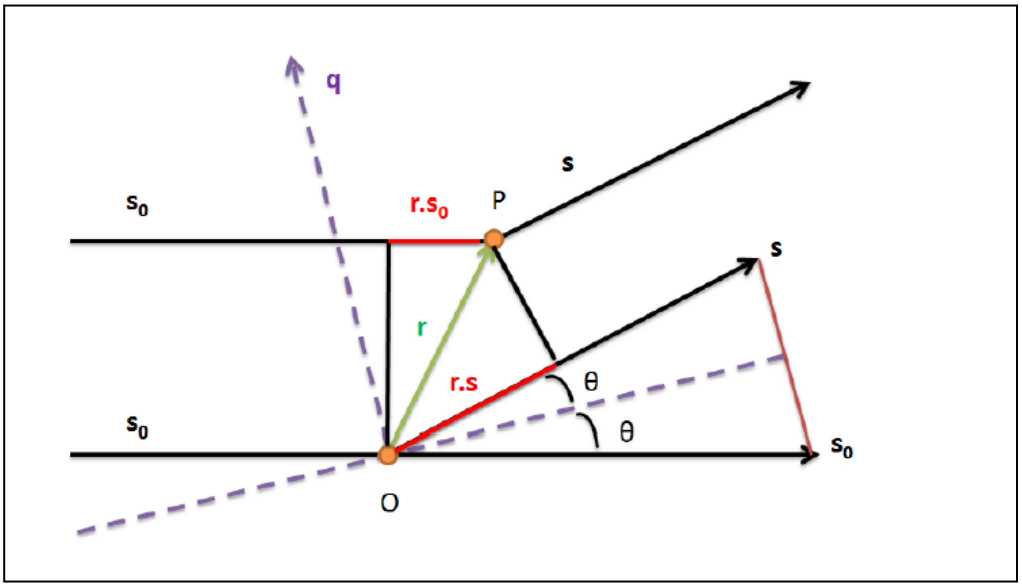


Figure 2. X-Ray Scattering: Adapted from Small Angle X-ray Scattering [2]. Incident (s_0) and Scattered – Rays (s) with the derivation of the momentum transfer vector q

The phase of the secondary waves is 2π/λ multiplied by the path difference between the scattered and incident waves. In the diagram, we let s_0 represent the direction of the incident beam and we let s represent the direction of the scattered beam. The path difference of a point P, specified by **r**, against the origin O is: -**r**·(**s**-$\mathbf{s_0}$). The phase is given by:

$$\varphi = -\frac{2\pi}{\lambda}\boldsymbol{r}(\boldsymbol{s} - \boldsymbol{s_0})\ ;\ \varphi = -\boldsymbol{qr} \qquad 1.13$$

The term (s-so) is symmetric to the incident and scattered beam with magnitude of 2sinθ. In this representation θ represents half the scattering angle. The momentum transfer vector q is independent of the distance to the detector and the wavelength (⍰) and defines the scattering curve in reciprocal space with units of Å-1. The momentum transfer vector has the same direction as (s-so) and the magnitude is given by substituting 2sinθ for (s-so):

$$|\boldsymbol{q}| = \frac{4\pi \cdot \sin(\theta)}{\lambda} \qquad 1.14$$

where 2θ is the scattering angle. We refer to q as the magnitude of the momentum transfer vector q. In the literature, this term has been defined multiple ways and one must be aware of the convention used. For example the symbols h and s have been used in place of q. Sometime s is defined as s = (2sinθ)/λ with q = 2πS. Others define θ rather than 2θ as the scattering angle. In this review we use the convention for q shown above with 2θ as the scattering angle. Large interatomic distances contribute primarily to the scattered X-ray intensity at small scattering angles, whereas short interatomic distances primarily contribute to X-ray intensity at large scattering angles. The information content of a SAXS profile is small compared to other high resolution experimental techniques because the overall scattering profile represents the orientationally averaged contribution of all atoms in all orientations. The

SAXS scattering curve contains information related to the overall shape of the molecule and is routinely used for the validation of structural models [17, 18].

The Scattering Intensity Curve can be derived from the Electron Density Function

The term electron density is frequently used in the literature in the place of electron density difference or contrast. The electron density ρ is the number of electrons per unit volume. In SAXS experiments only the electron density difference ρ2 – ρ1 (ρ2 is the electron density of the sample, ρ1 is the electron density of the solvent) is measurable. If ρ2 = ρ1, then scattering is not observed because the waves scattered in any direction will cancel out. During a SAXS experiment the electron density of the buffer solution is subtracted from the density of the combined sample and buffer solution leaving the electron density of the sample without background solution. The electron density function ρ(r) is defined in real space for non-negative values. It is a histogram of equivalent pairwise atomic distances in a given sample. Because of the solution subtraction, the electron density it is zero everywhere except for defined electron distances in the sample where identical distances add together.

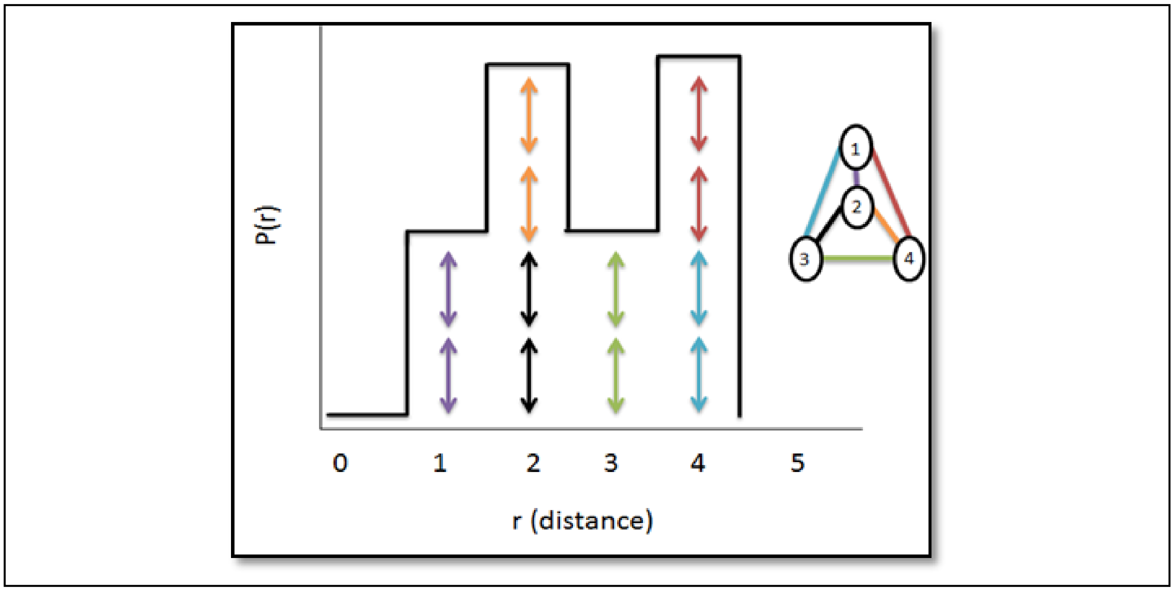


Figure 3: The pairwise distance distribution function adapted from X-ray solution scattering (SAXS) combined with crystallography and computation: defining accurate macromolecular structures, conformations and assemblies in solution. Pair-wise distances between each atom are represented. The distances are symmetric and are represented twice by the double arrows. The P(r) function will be zero whenever a particular distance is not defined by the geometry of the sample.

$$\mathrm{I(q)} = 4\pi \int_0^\infty \rho(\mathrm{r}) \frac{\sin(\mathrm{qr})}{\mathrm{qr}} \cdot \mathrm{dr} \qquad 1.15$$

Likewise the distance distribution function ρ(r) can be calculated by Fourier inversion of the scattering curve[2]:

$$\rho(\mathrm{r}) = \frac{1}{2\pi^2} \int_0^\infty \mathrm{I(q)} \cdot \mathrm{qr} \cdot \sin(\mathrm{qr}) \cdot \mathrm{dq} \qquad 1.16$$

Theoretical scattering curves can be computed for a model of a given shape and compared with experimental data using either the intensity calculation I(q) or the distance distribution function p(r). The distance distribution function allows the deduction of the largest particle dimension dmax and is the distance at which the p(r) drops to zero.

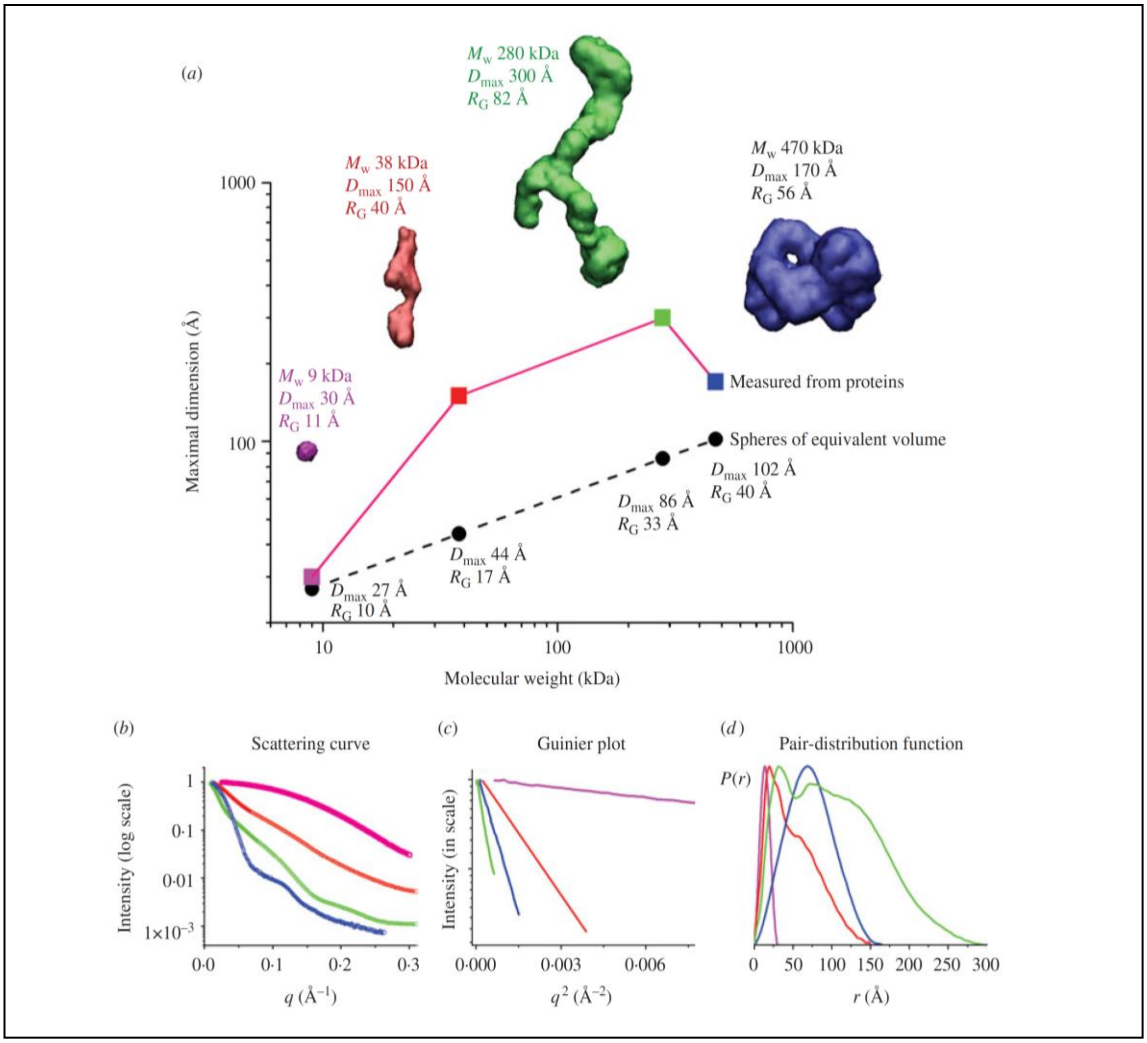


Figure 4: Originally from SAXS combined with crystallography and computation[3]. This figure depicts the experimental SAXS curves and parameters measured for Pyrococcus furiosis PF1282 rubredoxin (magenta), a 'designed' scaffoldin protein S4 (red), a 'designed' minicellulosome containing three catalytic subunits (green), and the DNA-dependent protein kinase (blue). (a) Dmax of the scattering particle is a simple function of molecular weight for perfect spheres, but not for proteins that adopt different shapes. Envelopes correspond to ab-initio models calculated from experimental curves using GASBOR. (b) The experimental scattering curves for each protein show that the intensity of scattering falls more slowly for rebredoxin (RG 11 Å ; magenta) than the minicellulosome (RG 82 Å; green). (c) The linear region of the Guinier plot, from which RG and I(0) can be derived, is a function of the RG. (d) Each protein has both a substantially different Dmax as well as pair-distribution function, reflecting the different atomic arrangements.

Debye Formula for computing scattering profiles from Atomic Coordinates

Proteins are built up from the arrangement of amino acids which are built up from the arrangement of atoms differing by side chain arrangement. Imagine a protein sample in a fixed orientation. The centers of mass of each atom may be designated by r1, r2, …, rn, and their amplitudes with respect to each mass center by f1, f2, …, fn. The total amplitude is[2]:

$$f_{\mathrm{protein}}(q) = \sum_{j=1}^{N} f_j\,(q) \cdot e^{-iqr_j} \qquad 1.17$$

where the additional phase factor describes the position of the atom and fj(q) is the amplitude. The intensity is the absolute square of the amplitude, averaged over all orientations:

$$I(q) = ff^* = \langle \sum_{j=1}^{n} \sum_{k=1}^{n} f_j f_k^* \cdot e^{-iq(r_j - r_k)} \rangle \qquad 1.18$$

When j=k the phase factor reduces to one. This situation represents the contribution to the intensity diffracted by the atoms alone. The situation j≠k represents the interference between the atoms, according to the relative distance (rj-rk). Each amplitude f has a phase:

$$f_j = \left\| f_j \right\| \cdot e^{i\varphi_j} \qquad 1.19$$

Splitting the atomic diffraction (j=k) from the interference between atoms (j≠k) yields:

$$I(q) = \sum_{j=1}^{N} I_j(q) + 2 \cdot \langle \sum \sum_{j \neq k}^{n} \left| f_j \right| \left| f_k \right| e^{i(qr_{jk} + \varphi_k - \varphi_j)} \rangle \qquad 1.20$$

In SAXS experiments there is no fixed origin because particles are sampled in all orientations. The phase is dependent on a fixed origin. By averaging over all orientations and restricting atoms to be spherical, the phase vanishes, (ϕk - ϕj) = 0 and fj becomes independent of orientation. Furthermore, spherical averaging of all orientations is given by:

$$\langle e^{iqr_{jk}} \rangle = \frac{\sin(q \cdot r_{ij})}{q \cdot r_{ij}} \qquad 1.21$$

This representation of the spherical averaging is known as the Debye factor[19]. The final Debye formula is:

$$\mathrm{I(q)} = \sum_{\mathrm{j=1}}^{\mathrm{N}} \mathrm{I_j(q)} + 2 \cdot \sum \sum_{\mathrm{j \neq k}}^{\mathrm{n}} \mathrm{f_j(q) f_k(q)} \frac{\sin \mathrm{qr_{jk}}}{\mathrm{qr_{jk}}} \qquad 1.22$$

In this format the amplitudes f are calculated by computing the atomic structure factors. The atomic diffraction and interference between atom sums can be combined together to give the form of the Debye equation frequently cited in the literature:

$$\mathrm{I(q)} = \sum_{\mathrm{i=1}}^{\mathrm{N}} \sum_{\mathrm{j=1}}^{\mathrm{N}} \mathrm{f_i(q) f_j(q)} \frac{\sin(\mathrm{q} \cdot \mathrm{r_{ij}})}{\mathrm{q} \cdot \mathrm{r_{ij}}} \qquad 1.23$$

where rij = | ri –rj | are the x,y,z positions of atoms i and j. The Debye formula given above takes the atomic x,y,z coordinates as input and returns the intensity as a function of momentum transfer q. This double sum of all atoms in a given system for each computed q value has a computational cost of O(N^2). The quadratic cost is a prohibitive barrier for atomic level application of the Debye formula for large systems (N > 10,000). In the case of structural refinement for SAXS, the scattering profile must be computed from all pairs of interactions with atoms in the molecule. In high-throughput applications the profile must be computed thousands of times, while in an iterative ensemble analysis, the profile must be computed hundreds of thousands of times. Because of the high computational cost, different methods have been developed to reduce the number of necessary calculations to compute intensity. Before we discuss the approximations to the Debye formula, we must first understand the structure factors fi(q) and fj(q).

Structure Factors and Form Factors

The atomic form factor is a fundamental physical quantity in solid state physics. It is the Fourier transform of an electron distribution around a nucleus of a given atom and carries information on the electron wave function. The X-ray scattering power of a given atom will depend on the number of electrons it contains. As the number of electrons contained in an atom increases (higher atomic number), the scattering power increases. As the scattering angle increases, the scattering power decreases. A scattering angle of zero results in the maximum scattering factor for a particular atom which is equal to Z – the atomic number. The form factor approximations are based on the combination of relativistic Dirac-Slater wave functions and numerical Hartree-Fock wave functions [20-23]. These Hartree-Fock structure factors were computed from q = 0 to q = 1.5 at intervals of 0.01Å-1. For convenience, they were fit to a 5-gaussian (Cromer-Mann) analytic function:

$$f_{v,i}(q) = \sum_{i=1}^{4} a_i \cdot e^{-b_i(\frac{q}{4\pi})^2} + c \qquad 1.24$$

where fv,i (q) is the structure factor of a particular atom at a given q-value in vacuo. The constants a1, a2, a3, a4, b1, b2, b3, b4, and c are the Cromer-Mann coefficients for a given atom, and q is the momentum transfer in inverse angstroms. Tables for the Cromer-Mann coefficients are found in the International Tables for X-Ray Crystallography[24]. This approximation is valid in the q-ranges for SAXS scattering experiments from 0 to ≈ 0.33Å-1 [3, 5]. For larger q-ranges, a 6-gaussian approximation must be used which is valid from 0 to ≈ 6.0Å-1 [23].

In addition to the vacuo contribution to the form factors, the solvent makes a critical contribution to the overall scattered intensity. The solvent effect is considered by modeling the solvent as an electron gas with density equal to the average electron density of the solvent[25]. Taking the solvent effect into account, the overall structure factor of the atom is the combination of the structure factor representing the excluded solvent subtracted from the form factor for a given atom:

$$F_i(q) = f_{v,i}(q) - f_{s,i}(q) \qquad 1.25$$

where fs,i is the structure factor of the hypothetical atom that represents the displaced solvent. The displaced solvent scattering term fs,i is given by:

$$f_{s,i}(q) = \rho V_i e^{-\frac{q^2 V_i^{2/3}}{4\pi}} \qquad 1.26$$

where ρ is the electron density of the solvent. For pure water this is 0.334e Å-3. Vi is the solvent volume V displaced by atom i and is calculated from the van der Waals radius of the atom.[25, 26]. The exponential term is the normalized Fourier transform of the Gaussian sphere. This sphere corresponds to the excluded volume around the atom.

The electron density surrounding the scattering body is calculated by computing the number of electrons per liter of solvent and then converting that to the number of electrons in a cubic angstrom. This excess electron density is then added to the density of pure water. Proteins have an electron density around 0.44e Å-3[5]. The electron density of the solvent should maximize difference between itself and the electron density of the sample to maximize contrast in SAXS experiments. The derivation for the electron density of pure water with a density of 1g/mL is shown below:

$$\left[\frac{6.02 * 10^{23} \mathrm{H_2O\ Molecules}}{1\ \mathrm{mol\ H_2O}}\right]\left[\frac{10\ \mathrm{electrons}}{1\ \mathrm{H_2O\ Molecule}}\right]\left[\frac{1\ \mathrm{mol\ H_2O}}{18\mathrm{g}}\right]\left[\frac{1\mathrm{g\ H_2O}}{1\mathrm{cm^3\ H_2O}}\right]\left[\frac{1\mathrm{cm^3\ H_2O}}{10^{24}\ \mathrm{Å^3}}\right] \approx 0.334\mathrm{e}\ \mathrm{Å}^{-3}$$

Now that we have reviewed the theory of X-ray scattering and have an idea of the Debye equation with a costly double sum over all atoms, we are ready to review methods using the Debye equation designed to maximize accuracy while minimizing computation time.

Fast approximation of the Debye Formula by Pantos and Bordas

In 1994, Pantos and Bordas used an approach to simulate SAXS patterns of large molecules by building models of closely packed spheres that are much larger than individual atoms thereby reducing N for the calculation. This was incorporated into the software package DALAI. They used the Debye formula to compute an intensity profile of the proposed model[27]:

$$\mathrm{I(q)} = \sum_{\mathrm{j=1}}^{\mathrm{N}} \mathrm{I_j(q)} + 2 \sum_{\mathrm{j=1}}^{\mathrm{N}} \sum_{\mathrm{k=1}}^{\mathrm{N}} \mathrm{F_i(q)F_j(q)} \frac{\sin(\mathrm{q} \cdot \mathrm{r_{ij}})}{\mathrm{q} \cdot \mathrm{r_{ij}}} \text{ ,} \mathrm{j} \neq \mathrm{k} \qquad 1.27$$

The first sum gives the intensity for spheres in isolation, while the double sum give the contributions from density-density correlations. To reduce the computational task in the double summations of the Debye equation, all spheres were given the same radius and mass density. The structure factor product Fi(q)Fj(q) is now constant for each value of q and can be pulled out of the double sum. The Debye formula becomes:

$$\mathrm{I(q)} = \sum_{\mathrm{j=1}}^{\mathrm{N}} \mathrm{I_j(q)} + 2\,\mathrm{F^2(q)} \sum_{\mathrm{j=1}}^{\mathrm{N}} \sum_{\mathrm{k=1}}^{\mathrm{N}} \frac{\sin(\mathrm{q} \cdot \mathrm{r_{ij}})}{\mathrm{q} \cdot \mathrm{r_{ij}}} \text{ ,} \mathrm{j} \neq \mathrm{k} \qquad 1.28$$

At this point in the formulation, Pantos and Bordas have not compromised the accuracy of the calculation for the reduced sphere model. They moved the bulk of the computation to the initial state of the algorithm. The calculation of rij is still O(N2). To model large structures requiring a large number of spheres, they approximated pairwise distances between atoms. In this approach pair distances are grouped into a histogram of bin sizes based on the experimental data resolution. Without binning, the number of pairwise distance terms is equal to N(N-1)/2. In this method the distances were quantized to multiples of dmax/100 where dmax is the maximum particle dimension. The resolution increases with decreasing bin size and decreases with increasing bin size. The resolution adjustment blurs the sampling grid by an undetectable amount in the resolution range of the simulation. The pair distance matrix of rjk values are now a vector of distances weighted by the number of distances occurring in a given bin. The scattering formula becomes:

$$\mathrm{I(q)} = \sum_{\mathrm{j=1}}^{\mathrm{N}} \mathrm{I_j(q)} + 2\,\mathrm{F^2(q)} \sum_{\mathrm{k=1}}^{\mathrm{Nbins}} \mathrm{m(r_k)} \frac{\sin(\mathrm{q} \cdot \mathrm{r_k})}{\mathrm{q} \cdot \mathrm{r_k}} \qquad 1.29$$

where m(rk) is the bin population at pair distance rk and the limits of the sum are the number of distance bins.

This method is valid when protein structures are modeled with multiple spheres of constant radii and mass density. When this condition is met, the structure factor calculation can be brought out of the double sum. The Debye calculation can then be binned leading to change of an O(N2) calculation to O(N). Prior to this calculation the pairwise distances must be pre-computed and binned which is still an O(N2) calculation. The speed increase by this algorithm is dependent on the number of spheres used to model the system. An advantage of this method is that the pairwise distance matrix must only be computed once and can then be reused during the course of analysis.

Calculation of SAXS profiles with the Debye formula from coarse-grained protein models

In 2010, Stovgaard et al, used the Debye formula for calculating the scattering curve combined with a coarse-grained representation of protein structure to address the high computational cost[28]. This approach led to a significant speed-up in computational time when compared with the all atom calculation. In this approximation, amino acids were represented by two scattering bodies or dummy atoms – one representing the backbone, and the other representing the side chain. These dummy atoms were placed at the respective center of mass of the atomic group they represented. They had to estimate 21 form factor values for this approximation – one for alanine, one for glycine, one for the backbone, and 18 for the remaining side chains. They recreated these functions for each of the 21 form factors by binning the q-range into intervals of equal width (0.015 Å-1) and then computing a form factor estimate for each of the 21 form factor types in each of the q-bins. They sampled form factor values from a training set of 297 structures with lengths between 50 and 400 residues and calculated a form factor estimate from the centroid in each bin. The SAXS curves generated through the Debye formula with dummy atom form factors for 50 proteins were compared with SAXS curves generated for the same proteins through CRYSOL with high agreement.

This method is contingent upon the accuracy of the form factor estimates for the dummy atoms and relies on a training set of 297 proteins to represent amino acids in nature. Amino acid residues behave differently in different environments, and caution must be used to ensure the training set accurately represents the environment of the protein of interest. The authors state that two additional developments with this method are needed: 1) a proper description of the hydration layer and 2) a probabilistic description of the experimental errors associated with a SAXS experiment. This is currently under development in the PHAISTOS software package.

The incorporation of the hydration layer into the Debye Formula via the form factor equations

In the same year that PHAISTOS was published, the Sali Lab published their approach to the Debye formula and made their web server FoXs publically available[29]. To account for the displaced solvent and hydration shell, the structure factor contribution for a given atom is given by:

$$\mathrm{F_i(q) = f_{v,i}(q) - c_1\, f_{s,i}(q)\ +\ c_2 S_i f_{w,i}(q)} \qquad 1.30$$

where fv,i (q) is the form factor of a particular atom at a given q-value without the effects of excluded volume and a water shell, and fs,i is the structure factor for the excluded volume, and the last term is the structure factor of the hypothetical molecule that represents the displaced solvent. Si is the solvent accessible surface area for a given heavy atom and fw,i is the form factor of water. This approach is novel because it models the hydration shell as a function of the solvent accessible surface area of a given atom. The parameter c1 is used to adjust the electron density contrast while the parameter c2 is used to adjust the hydration shell thickness. The form factor of water is given by the sum of all atomic form factors in water:

$$\mathrm{f_{w,i}(q) = 2 * f_{v,i}(q)_{hydrogen} +\ f_{v,i}(q)_{oxygen}} \qquad 1.31$$

The computed profile was fit to a given experimental SAXS profile by minimizing the chi function with respect to c, c1, and c2:

$$\chi = \sqrt{\frac{1}{\mathrm{M}}\sum_{\mathrm{i}=1}^{\mathrm{M}}\left(\frac{\mathrm{I}_{\mathrm{exp}}(\mathrm{q}_{\mathrm{i}}) - \mathrm{cI}(\mathrm{q}_{\mathrm{i}})}{\sigma(\mathrm{q}_{\mathrm{i}})}\right)^{2}} \tag{1.32}$$

where Iexp(q) and I(q) are the experimental and computed profiles, σ(q) is the experimental error of the measured profiles, M is the number of points in the profile, and c is the scale factor. The minimum value of chi was found by a computing c1 on the interval of [0.95, 1.12] and c2 on the interval of [0, 4.0] in steps of 0.005 and 0.1. Linear least squares minimization was performed to find the value of c that minimized chi for each c1 and c2 combination.

Similar to DALAI, FoXs has the form factor calculation moved out of the double sum of the Debye formula. Instead of modeling uniform space filling spheres, they assumed an identical modulation of fi(q) for different atoms i:

$$\mathrm{f}_{\mathrm{i}}(\mathrm{q}) = \mathrm{f}_{\mathrm{i}}(0) \cdot \mathrm{E}(\mathrm{q}) \tag{1.33}$$

where the modulation function E(q) is equal for all atoms. This approximation creates a system of different scattering masses but equal shape. The pairwise distance distribution function represents population at a given distance r and is given in this approximation as:

$$\rho(\mathrm{r}) = \sum_{\mathrm{i,j}} \mathrm{f}_{\mathrm{i}}(0)\mathrm{f}_{\mathrm{j}}(0) \cdot \delta(\mathrm{r} - \mathrm{d}_{\mathrm{ij}}) \tag{1.34}$$

where ▯(r-dij) is the Dirac-Delta distribution and r is a given pairwise distance. In this representation, only the form factor with a constant q = 0 is considered, which reduces the value to the atomic number Z of the given value. The intensity is given by:

$$\mathrm{I}(\mathrm{q}) = \mathrm{E}^{2}(\mathrm{q}) \cdot \int_{0}^{\infty} \rho(\mathrm{r}) \frac{\sin(\mathrm{qr})}{\mathrm{qr}} \mathrm{dr} \tag{1.35}$$

The modulation function E2(q) is parameterized as:

$$E^2(q) = e^{(-b*q^2)} \tag{1.36}$$

The parameter b was determined by computing the SAXS profile with the original Debye formula using the non-approximated form factors and then computing the SAXS profile with the approximated form factors and initial guess of the b parameter. The parameter b=0.23±0.01 Å-1 was chosen to minimize the difference between both profiles from 30 random protein structures from the Protein Data Bank. This approximation typically speeds to calculation of the Debye formula by two orders of magnitude.

The explicit incorporation of the hydration layer into the Debye Formula

In 2011, the Zhang lab at the University of Michigan introduced SAXSTER, an online tool to improve protein template recognition by using SAXS profiles[30]. In their approach they also simulate the SAXS intensity profile according to the Debye equation. Instead of summing over all atoms, they sum over all atoms plus the explicit water atoms. The equation is:

$$I(q) = \sum_{i=1}^{N+W} \sum_{j=1}^{N+W} F_i(q) F_j(q) \frac{\sin(q \cdot r_{ij})}{q \cdot r_{ij}} \tag{1.37}$$

where W is the number of “dummy” water molecules around the protein representing the hydration shell. The initial structure factor equations are identical to equations previously shown. To account for the explicit water molecules around the model, they started from a face-centered cubic (FCC) lattice system with edge length Lcell. Each point in the lattice represents a water molecule. The overall structure factor is given by subtracting the excluded solvent from the atomic form factor and adding the explicit water contribution from the lattice. The protein structure is projected onto the FCC system and only water molecules in the range of 3.5-6.5 Å to any Cα atoms are kept. The density of the water molecules in the lattice system is defined by:

$$\rho_{FCC} = \frac{N_{FCC}}{V_{FCC}} = \frac{4k^3}{L^3} \qquad 1.38$$

where N is the number of points in the FCC lattice system, V is the volume of the system, k is the number of unit cells in the x,y,z directions and L = k * Lcell. L represents the maximum length for each direction. In a FCC lattice system, the water contribution from each corner of the cubic cell is 1/8 and the contribution from each face is 1/2. There are eight corners and six sides yielding an effective water contribution of four (8(1/8) + 6(1/2)). Each water molecule consists of 10 electrons yielding 40 (water contribution of four * 10 electrons) electrons per cubic cell. The number of excess electrons per volume in the hydration shell relative to the bulk water is:

$$\delta\rho = \frac{40 \text{ electrons}}{L_{cell}^3} = \rho_{shell} - \rho_{bulk} \qquad 1.39$$

The thickness of the hydration shell is thus controlled by the edge Length of the FCC system. The threading-based models are composed of α-carbons only and the SAXS computations are given by:

$$I(q) = \sum_{i=1}^{N+W} \sum_{j=1}^{N+W} F_{eff}^{i}(q) F_{eff}^{j}(q) \frac{\sin(q \cdot r_{ij})}{q \cdot r_{ij}} \qquad 1.40$$

$$\rho(r) = \sum_{i=1}^{N+W} \sum_{j=1}^{N+W} F_{eff}^{i}(q=0) F_{eff}^{j}(q=0)\, \delta\,(r - r_{ij}) \qquad 1.41$$

This form of the ρ(r) function is very similar to FoXs. The difference is that the water molecules are explicitly summed over. In the approximation, a new structure factor must be derived to represent the α-carbons:

$$F_{eff}(q) = \langle \sum_{i=1}^{N+W} \sum_{j=1}^{N+W} F_i(q) F_j(q) \frac{\sin(q \cdot r_{ij})}{q \cdot r_{ij}} \rangle^{1/2} \qquad 1.42$$

where ⟨… ⟩ denotes the average over all residues of the same type calculated from 200 randomly selected PDB structures. The term f(q) is computed by the initial structure factor equations previously shown. This procedure produces 20 effective structure factors for each amino acid type. In the case of water, its scattering factor is calculated by the modified Debye equation with n = 3, rij = 0 and Fi(q) being the vacuum form factors for either hydrogen or oxygen.

Spherical Harmonics - A second widely used approach to address the computational cost of SAXS profile reconstruction

In the methods previously described, the orientational averaging of the scattered waves was computed analytically using the Debye relation[19]:

$$\langle e^{iqr} \rangle = \frac{\sin(q \cdot r)}{q \cdot r} \qquad 1.43$$

Instead of analytically computing the orientational averaging, another method is to use a mathematical representation of the scattering body (or protein) that uses the rotational properties of spherical tensors. In this formulation the scattering body is expanded in terms of an infinite series of spherical harmonics. The orthogonality properties of the basis functions simplify the averaging of the harmonic series from which an overall scattering intensity can be computed. These basis functions are built from spherical Bessel functions, and normalized spherical harmonics of degree m and order L. This approach reduces the computational complexity from O(N2) to O(N).

The scattering amplitude in vacuo of a particle with N atoms is:

$$A_{\mathrm{vacuo}}(q) = \sum_{j=1}^{N} f_j(q) e^{iqr_j} \tag{1.44}$$

where rj = (rj,ωj) = (rj,θj,ϕj) and fj is the corresponding atomic form factors. Spherical averaging is simplified by multipole expansion[31]:

$$e^{iqr} = 4\pi \sum_{L=0}^{L_{\max}} \sum_{m=-L}^{L} i^L j_L(qr) Y_{Lm}^{*}(\omega) Y_{Lm}(\Omega) \tag{1.45}$$

where jL(qr) are the spherical Bessel functions of order L and YLm(Ω) are the spherical harmonics of order (L,m). The angular symmetry of YLm is related to the symmetry of the multipoles: L=0 (monopole) L=1 (dipole), L=2(quadrupole), etc. Substituting the multipole expansion with spherical harmonics for the exponential term yields:

$$A_{\mathrm{vacuo}}(q) = \sum_{L=0}^{L_{\max}} \sum_{m=-L}^{L} 4\pi i^L Y_{lm}(\Omega) \sum_{j=1}^{N} f_j(q)\, j_L(qr_j) Y_{Lm}^{*}(\omega_j) \tag{1.46}$$

where (rj,ωj) are the polar coordinates of the jth atom. The partial amplitudes can be separated from the proceeding equation:

$$A_{\mathrm{vacuo}}(q) = \sum_{L=0}^{L_{\max}} \sum_{m=-L}^{L} A_{Lm}(q) Y_{Lm}(\Omega) \tag{1.47}$$

where ALm(q) are the partial amplitudes and are given by:

$$A_{Lm}(q) = 4\pi i^L \sum_{j=1}^{N} f_j(q)\, j_L(qr_j) Y_{Lm}^{*}(\omega_j) \tag{1.48}$$

Because of the orthogonality properties of spherical harmonics, the cross terms cancel and the intensity calculation is reduced to[32]:

$$I_{vacuo}(q) = \int |A_{Lm}(q)|^2 dS(u) = \sum_{L=0}^{L_{max}} \sum_{m=-L}^{L} |A_{Lm}(q)|^2 \tag{1.49}$$

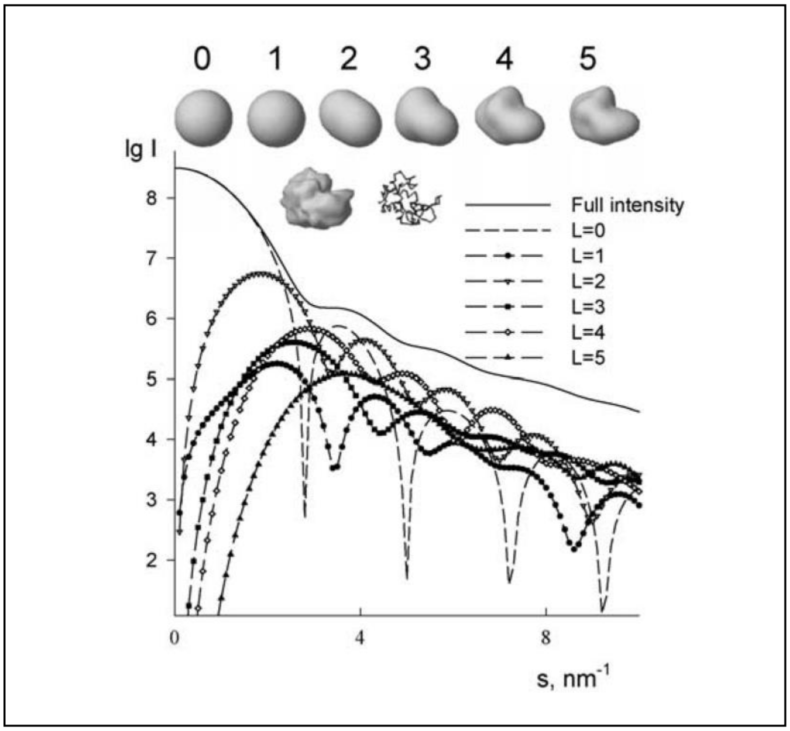


Figure 5: Originally from Models, structures, interactions and scattering. Accuracy shape representations using spherical harmonics. Top row: surface representations of truncated envelope functions of lysozyme. Second row: high-resolution envelope functions and Cα trace of the protein. The shape scattering intensity from lysozyme is shown along with the contributions from different multipoles.

The huge advantage of spherical harmonics is that the complexity is reduced from O(N^2) to O(N). The integrand for averaging over the sphere in the proceeding equation is approximated by an L = O(qD) band limited function in a spherical harmonic basis where q is the momentum transfer vector and D is the maximum dimension of the sample. It is insufficient to use L smaller than qD/2 because any value less than this violate Nyquist Shannon sampling[33] for periodic functions. At least L2 = O(q^{2}D^2) sampling points are needed to provide an accurate integration of bandwidth L. Any index above L does not improve the fit for a given qmax, while any index below L will result in systematic errors in the calculation[34].

CRYSOL – The incorporation of the hydration shell using spherical harmonics with multipole expansion to compute SAXS profiles from atomic coordinates

By the early 1990s many there were many studies showing the importance of modeling the water molecules surrounding a given macromolecule when recreating SAXS profiles from atomic coordinates. For example, Grossman et. al compared experimental SAXS profiles with SAXS profiles computed from different configurations of dimers, trimers, and tetramers. They optimized the agreement between experimental and simulated scattering profiles by placing solvent molecules on a diamond-shaped grid surrounding the structure[35]. In their results, the computed SAXS profile with the best fit to the experimental SAXS profile consisted of a solvent shell of 716 water oxygens up to a maximum distance of 3.15 Å from the protein surface. Their results suggested that the water shell very close to the surface of a protein differs in electron density from the remaining bulk water and thus contributes to x-ray scattering.

In 1995 Svergun et. al released CRYSOL – a program to compute SAXS intensity profiles from atomic coordinates while considering the hydration shell surrounding the target sample[26]. There were lingering questions concerning the true cause of the electron density contrast conditions surrounding a sample in solution. Was the density contrast caused by a water layer or could the contrast be explained by side chains moving freely on the protein surface? Three years later in 1998 Svergun et al. confirmed in a combined X-ray and neutron scattering study that the differing electron contrast conditions were more likely caused by a denser hydration shell rather than a higher mobility of the side-chains on the protein surface[36]. Water modeling is critical to the correct interpretation of SAXS profiles and computational methods are under development today to improve chemistry constraints, improve geometric constraints (surface curvature), and incorporate experimental data from high-angle SAXS[37].

Currently, popular approaches for modeling the hydration shell are to: 1) place water molecules on the surface of the protein, 2) simulate the solvation shell by surrounding the protein with a continuous outer envelope, 3) simulate the solvation shell and excluded volume by computing a modified scattering factor.

CRYSOL employed the second approach to model the hydration shell and extended the multipole expansion and spherical harmonics formulation to handle not only the vacuo scattering, but the excluded volume and hydration shell.[11]

In this formulation the intensity is given by:

$$\mathrm{I(q)} = \langle |\mathrm{A_a(q)} - \rho_0 \mathrm{A_c(q)} + \delta\rho \mathrm{A_b(q)}|^2 \rangle_\Omega \qquad 1.50$$

where Aa(q) is the in vacuo scattering, $A_c(q)$ is the excluded volume scattering and $A_b(q)$ is the border layer scattering, $\delta\rho = \rho_b - \rho_0$, where ρ_0 is the average scattering density of the solvent surrounding the particle and ⍰b is the average scattering density of the border layer around the particle with thickness Δ. $\langle\ \rangle_\Omega$ stands for the average over all particle orientations and Ω is the solid angle in reciprocal space, q = (q, Ω). Each of the three amplitudes is represented via its multipole components. Because of the orthogonal properties of the spherical harmonics, all cross terms cancel in the average over Ω, leading to:

$$\mathrm{I(q)} = \sum_{\mathrm{l}=0}^{\mathrm{L}} \sum_{\mathrm{m}=-\mathrm{l}}^{\mathrm{l}} |\mathrm{A_{lm}(q)} - \rho_0 \mathrm{C_{lm}(q)} + \delta\rho \mathrm{B_{lm}(q)}|^2 \qquad 1.51$$

The value L defines the resolution of the particle. This approach works best with shapes that can be described using spherical harmonics which include most globular and extended proteins. Spherical harmonics is less adept at handling shapes that contain internal cavities such as shells and donuts.[26] Additionally this method uses by default a harmonic order of 15, with a maximum value of 50. This gives the method a complexity of O(MN) with M=q2D2. This can lead to errors when a harmonic order greater than 50 is necessary based on the size of the protein and desired qmax.

In CRYSOL there are several adjustable parameters used when calculating predicted data that best match the experimental curve. These parameters are: the effective atomic radii multiplier which scales the solvent volume displaced by each atom (vi), the electron density contrast of the surface solvent layer (c_2) and the total displaced solvent volume (c_1), approximately equal to the variation of the electron density of the displaced solvent relative to bulk water. The need for adjustable parameters in CRYSOL becomes clear when studying SAXS profile reproducibility for distinct samples of the same protein on different instruments. The characteristic features of the experimental scattering profiles are conserved between experiments, but the experimental variation of the scattered intensity at higher q-values depends on the extrapolated intensity at I(0)[38]. Because of the beamstop in a SAXS experiment, I(0) cannot be directly observed. One method to extrapolate this value is to compute the slope of the intensity profile in the initial linear region of the scattering profile (the Guinier region) and extrapolate to the y-intercept. The adjustable parameters in CRYSOL absorb this variability by changing the level of the higher-q features of the predicted data relative to the low-q intensities.

Extension of CRYSOL to improve accuracy

Fifteen years after the introduction of the original CRYSOL program, Alexander Grishaev, Liang Guo, Thomas Irving and Ad Bax introduced AXES in 2010 – a program for fitting SAXS data to macromolecular structure and ensembles of structures[38]. The program AXES was designed to be more discriminating than CRYSOL when evaluating poorly or incorrectly modeled protein structures. On a set of small well-studied proteins that had X-ray crystallography and solution NMR data they reported an improvement in fit by 10-50% by χ score. This set was comprised of four proteins – hen egg white lysozyme, cytochrome c, the B3 domain of protein G (GB3) and ubiquitin.

They reformulated the approach to fitting SAXS data by explicitly taking into account the sources of experimental data variability:

$$I_{exp}(q) = I_{sample}(q) - \alpha I_{buffer} + c \qquad 1.52$$

where α accounts for the uncertainty in the measurements and c accounts for the variability of the detector and X-ray fluorescence. These uncertainties appear responsible for the systematic difference between repeated experimental data sets. Taking these uncertainties into account, the computed scattering intensity is:

$$I(q) = \langle\langle\langle |A_a(q) - \rho_0 A_c(q) + \delta\rho A_b(q)|^2 \rangle_{\Omega}\rangle_{solv}\rangle_{ens} \qquad 1.53$$

where Ω is the average taken over a discrete set of molecular orientations relative to the incident beam, solv is the average taken over the displaced and surface water sets, and ens is the average over the ensemble of macromolecular structures. The program AXES models the hydration shell directly by using explicit water molecules from a pre-equilibrated water box.

Using this approach they tested how well they could discriminate different models of the same protein. They generated 2000 models of GB3 using Rosetta and fit the experimental SAXS data to all of the models using both CRYSOL and AXES. The CRYSOL fits yielded χ values that were much lower for poor models (models with a high RMSD relative to the native structure) than the native structure. This behavior is indicative of overfitting. Using AXES, they did not observe significantly better fits for the poor Rosetta models. Furthermore, when provided chemical shift guided Rosetta models with the correct fold, AXES correctly assigned higher χ values to non-native structures.

The cost of this higher precision comes at the price of computation time. AXES is more than an order of magnitude slower than CRYSOL due to the averaging of the scattering amplitudes of the displaced and surface solvent sets over 20 different configurations. Among these configurations are: 6 elementary scattering functions averaged over angular orientations, macromolecular conformers, and molecular solvent configurations for a given electron density contrast of the surface solvent layer. Currently several avenues for computation speedup are under development.

The use of Zernike polynomials to compute SAXS scattering profiles

We previously mentioned three popular approaches for treating the hydration shell and excluded solvent. They were: 1) to place water molecules on the surface of the protein and compute scattering profiles with explicit water molecules, 2) simulate the solvation shell by surrounding the protein with a continuous outer envelope, 3) simulate the solvation shell and excluded volume by computing a modified scattering factor. The drawback to the first approach is the computational cost to construct the explicit solvent model. The drawback of the second approach occurs for proteins containing cavities. Assuming a uniform layer around a cavity or hole will introduce artificial areas without any electron density. The drawback of the third approach is the appearance of non-uniformities in the electron density by overlapping dummy atoms.

In 2012, Liu et al proposed a new method to address the limitations of excluded solvent and hydration shell modeling[32]. In their approach they parameterized the Fourier transform of the electron density distribution function p(r) by a Zernike polynomial expansion with spherical harmonics. Zernike polynomials are orthogonal functions on the unit ball. They reformulated the SAXS intensity calculation as:

$$\mathrm{I(q)} = 16\pi^2 \sum_{n=0}^{\infty}\sum_{n'=0}^{\infty} b_n(qr_{max})\, b_{n'}(qr_{max}) F_{nn'} \tag{1.54}$$

$$b_n(qr_{max}) = \frac{j_n(qr_{max}) + j_{n+2}(qr_{max})}{2n+3} \tag{1.55}$$

where jn is the spherical Bessel function of order n.

$$F_{nn'} = \sum_{l=0}^{n} k_{nn'l} \sum_{m=-l}^{l} c_{nlm}\, c^{*}_{n'lm} \tag{1.56}$$

where cnlm is the Zernike moments from three-dimensional objects and knn'l is either a positive or negative coefficient given by:

$$k_{nn'l} = (-1)^{\frac{n+n'}{2-l}} \tag{1.57}$$

The Zernike moments are computed by a linear combination of the geometric moments of the object:

$$c_{nlm} = \frac{3}{4\pi} \sum_{r+s+t \leq n} \overline{\chi_{nlm}^{rst}} M_{rst} \qquad 1.58$$

where Mrst is the geometric moment and $\chi_{nlm}^{rst}$are the coefficients. The procedure to compute the coefficients are given by the Novotni and Klein algorithm[39].

$$M_{rst} = \int_{|r| \leq 1} \rho(r) x^r y^r z^r dr \qquad 1.59$$

The geometric moments are computed from a scattering object that has been segmented into a series of small volume cubes called voxels. Voxels are used in 3D graphics for the visualization and analysis of medical and scientific data. In this case the voxelization process maps electron density from the scatterer (or protein) into voxels from which the geometric moments can be computed. From this process, multiple sets of voxels are created: 1) P – the set of non-zero electron density voxels, 2) S+B – the set of voxels representing the excluded solvent and surface bound solvent, and 3) S – the set of voxels representing the excluded solvent.

The Zernike moments of all three voxelized objects are combined by a weighted sum to produce one set of Zernike moments from which the scattering intensity is computed. The computational complexity of this algorithm is O(N), but prior to computation, the voxelized object must be created in a preprocessing step.

The advantage of the Zernike expansion method is that it can model holes or cavities of structures that spherical harmonics traditionally has difficulty with. This approach also incorporates all solvent-accessible surfaces into the overall scattering profile. When compared on a set of ten experimental proteins with high resolution crystal structures, this method had similar results with the spherical harmonic expansion method. This method offers an extension to spherical harmonic expansion methods that may improve the fit to experimental data by improved hydration shell and excluded volume treatment of structures with cavities or holes. It is included in the SASTBX software package.

Table 2: Summary of Techniques to reconstruct SAXS profiles from Atomic Coordinates

Year	Method	Complexity	
		Big O	M
1994	DALAI (Debye with binned pairwise distance) $I(q) = \sum_{j=1}^{N} I_j(q) + 2\,F^2(q) \sum_{k=1}^{Nbins} m(r_k) \frac{\sin(q \cdot r_k)}{q \cdot r_k}$	$O(N^2)$	-
1995	CRYSOL (Multipole expansion and spherical harmonics) $\sum_{L=0}^{L_{max}} \sum_{m=-L}^{L} \left\lvert 4\pi i^L \sum_{j=1}^{N} f_j(q)\, j_L(qr_j) Y^*_{Lm}(\omega_j) \right\rvert^2$	O(MN)[37]	(q^2D^2)[34]
2010	PHAISTOS (Debye with Bayesian modeling of form factor) $I(q) = \sum_{i=1}^{N} \sum_{j=1}^{N} f_i(q) f_j(q) \frac{\sin(q \cdot r_{ij})}{q \cdot r_{ij}}$	$O\left(\left[\frac{M}{k}\right]^2\right)$[40]	M: number of atoms in the structure K: number of atoms described by a dummy body. Kave = 4.24
2010	FOXS (Debye with approximated structure factor) $I(q) = \sum_{i=1}^{N} \sum_{j=1}^{N} f_i(q) f_j(q) \frac{\sin(q \cdot r_{ij})}{q \cdot r_{ij}}$	$O(N^2)$[37]	-

2010	AXES (multiple averaging with spherical harmonics and explicit water molecules) $\langle\langle\langle\lvert A_a(q) - \rho_0 A_c(q) + \delta\rho A_b(q)\rvert^2\rangle_\Omega\rangle_{solv}\rangle_{ens}$	O(MN)[37]	M: number of spherical grid points
2011	SAXSTER (Debye with explicit water molecules) $I(q) = \sum_{i=1}^{N+W}\sum_{j=1}^{N+W} F_i(q)F_j(q)\frac{\sin(q \cdot r_{ij})}{q \cdot r_{ij}}$	O([N+W]2)	-
2012	SASTBX (3D Zernicke polynomials) $\left\lvert\sum_{n=0}^{n_{max}}\sum_{l=0}^{n}\sum_{m=-l}^{l} i^l(-1)^{(n-1)/2}c_{nlm}Y^*_{lm}(w_q)b_n(q)\right\rvert^2$	O(MN)[37]	(Nmax + 1)2

Recent developments for SAXS profile reconstruction using GPU acceleration

In 2012, the SAXS algorithm in PHAISTOS was accelerated using general purpose graphical processing units (GPGPUs)[40]. This method utilizes Bayesian probability statistics to compute the form factors in the Debye equation for protein models built from either one or two scattering bodies. The speed up using GPU's was measured from protein sizes ranging from 64 to 8192 scattering bodies. They reported a 16x speed up for proteins with 64 scattering bodies. As the proteins increased in size the speed up increased to a maximum speed up of 394x for proteins with 8192 scattering bodies.

Because of the uncertainty introduced into the accuracy of the Debye equation by approximation methods, we devised a method to compute the intensity directly without approximating structure factor calculations[41]. Furthermore, we model the hydration shell as a function of the solvent accessible surface area of a given atom analogous to FoXs. Our method BCL::SAXS offsets the high computational cost of the Debye formula by simultaneously computing multiple pieces of the equation using the parallel architecture of graphical processing units (GPUs). The Debye formula can be framed as an NxN square matrix of N-atom rows by N-atom columns where N is the number of atoms in a given protein. The pairwise Euclidean distances (rij) are calculated from the upper triangle of the matrix. The diagonal is set to zero and the lower triangle is a symmetric mirror of the upper triangle. Each GPU thread computes a partial Debye sum.

$$\mathrm{I_{partial}(q)} = \sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{F_i(q)F_j(q)} \frac{\sin(\mathrm{q} \cdot \mathrm{r_{ij}})}{\mathrm{q} \cdot \mathrm{r_{ij}}} \qquad 1.60$$

This results into a matrix of q rows by N-atom columns where q is the momentum transfer and N is the total number of atoms. These partial values are summed across each column to complete the intensity computation:

$$\mathrm{I_{total}(q)} = \sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{I_{partial,n}} \qquad 1.61$$

This approach removes the uncertainty introduced by form factor approximation while maintaining the efficiency of methods with form factor approximations. The speed up using GPU’s was measured from protein sizes ranging from 1832 atoms (PDB ID: 1O26) atoms to 91,846 (PDB ID: 1VSZ). Using a GTX680 GPU card, we observed a 5x speed up for the smaller protein (1O26). For the largest protein in our set (1VSZ) we observed a speed up of 1707x for protein 1VSZ using the same graphics card. By leveraging GPU’s, we absorb the O(N2) cost while achieving substantial reduction in computation time without sacrificing accuracy by introducing approximations to the Debye formula.

Conclusion

In this review we focused on proteins as a scattering body, but RNA and DNA can be studied as well using SAXS. These algorithms represent a sampling of methods for SAXS profile reconstruction and are not representative of all the approaches that exist. Another approach that expands these ideas was published in 2012. In this work, Gumerov et. al proposed a Hierarchal algorithm based on a fast multipole method (FMM) to compute SAXS profiles[34]. For a review of timing and accuracy for protein of varying sizes and shapes with either spherical harmonic or Debye implementations we refer to their work. In each of the algorithms presented, there was a trade-off between speed and accuracy. In order to use the Debye formula for protein structure analysis, approximations were made to the equation to move terms out of the double sum. The uncertainty introduced by this approach is a subject for further study. In order to model with spherical harmonics, the correct harmonic order must be set and the shape complexity of the scattering body must be considered. We expect that more algorithms in the near future will take advantage of the parallelizable form of the Debye equation and use GPU acceleration to obtain the necessary computational speed without the uncertainty introduced by structure factor approximation and momentum transfer binning.

Furthermore, to standardize testing of SAXS algorithms we echo the suggestion of Rambo and Tainer and believe a reference dataset should be created with experimental SAXS profiles and PDB models[37]. This dataset would be comprised of proteins of varying sizes and shapes and folds. All new and existing methods should be benchmarked against this set to identify strengths and weakness of any given algorithm.

CHAPTER II

BCL::SAXS: GPU ACCELERATED DEBYE METHOD FOR COMPUTATION OF SAXS PROFILES

Overview

This chapter is a reproduction of BCL::SAXS: GPU accelerated Debye method for computation of small angle X Ray scattering profiles [41]. Brian Weiner and I wrote the CPU implementation, Edward W. Lowe Jr. wrote the GPU implementation and performed the timing of the GPU algorithm, Nils Woetzel and Jens Meiler provided insight and guidance during algorithm development and manuscript production.

Significance

Small angle X-ray scattering (SAXS) is an experimental technique used for structural characterization of macromolecules in solution. Here, we introduce BCL::SAXS – an algorithm designed to replicate SAXS profiles from rigid protein models at different levels of detail. We first show our derivation of BCL::SAXS and compare our results with the experimental scattering profile of Hen Egg White Lysozyme. Using this protein we show how to generate SAXS profiles representing: 1) complete models, 2) models with approximated side chain coordinates, and 3) models with approximated side chain and loop region coordinates. We evaluated the ability of SAXS profiles to identify a correct protein topology from a non-redundant benchmark set of proteins. We find that complete SAXS profiles can be used to identify the correct protein by receiver operating characteristic (ROC) analysis with an area under the curve (AUC) > 99%. We show how our approximation of loop coordinates between secondary structure elements improves protein recognition by SAXS for protein models without loop regions and side chains. Agreement with SAXS data is a necessary but not sufficient condition for structure determination. We conclude that experimental SAXS data can be used as a filter to exclude protein models with large structural differences from the native.

Innovation

This is the only algorithm in the world to Reconstruct SAXS profiles for models comprised of the backbone atoms of SSEs**.** To perform this comparison we developed a novel method to rapidly approximate the loop regions between secondary structure elements. We developed a novel scoring method that used the derivative of the SAXS profiles for ranking the agreement of two given SAXS profiles.

Introduction

Protein structure determination remains a major challenge in the field of structural biology[42]. While X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy can provide high resolution structures, these techniques can be limited by size[43], high flexibility[44], and membrane environment[44]. Computational de novo protein structure prediction methods have been developed, but are limited by the vast conformational search space that needs to be searched when no template structure is available[45]. To overcome these experimental and computational limitations, hybrid methods – i.e. the combination of multiple techniques – can be utilized to gain structural insights of proteins[9, 46, 47].

SAXS offers an alternative to traditional structure determination techniques

Small angle X-ray scattering (SAXS) is an experimental structural characterization method for rapid analysis of biological macromolecules in solution[3, 5-8]. During data acquisition in SAXS, macromolecules move freely in solution while a beam of X-Rays with constant wavelength λ irradiate the sample. At the point of interaction between X-Rays and electrons in the sample, both elastic and inelastic scattering occur. This work considers the case of elastic scattering by electrons. The intensity of the scattered X-Rays captured on the detector is proportional to the Fourier Transform of a pairwise distance function ρ(r) that gives the probability of finding two atoms a certain distance apart. This distance function is weighted by the excess scattering density of the respective scattering volume compared to the solvent. For a more comprehensive review of SAXS theory we recommend several reviews[2-5, 15, 48]. SAXS profiles are reported by intensity (I) as a function of momentum transfer vector (q). Large interatomic distances contribute to the intensity profile at small q, while short interatomic distances contribute to the intensity at large q. Several parameters can be extracted directly from the scattering profile including: the molecular mass (MM), radius of gyration (Rg), hydrated particle volume (Vp) and maximum particle diameter (Dmax). The state of the protein (folded vs. unfolded) can be observed from the Kratky representation of the scattering data plotting q vs. q2I(q). The scattering profile can be transformed into the pairwise distance density function which is a histogram of distances between pairs of points in a particle. This shape information has been used for the validation of structural models[17, 18].

Use of SAXS experimental data in computation

The experimental SAXS profile has been used to filter a set of proposed models by comparing the computed SAXS profile of each model with the experimental data[9, 10]. Furthermore, the experimental profile has been incorporated into an energy function for protein folding to obtain a model consistent with experimental data[11]. More recently SAXS has been used to identify and model protein flexibility from an ensemble set of conformers [12]. In this approach a large library of initial conformers are given as input. After a sufficient library of conformers has been found, the experimental SAXS data are used to ascertain which combination of conformers optimally fit the data. In this case, the scattering intensity (I) is represented by a linear combination of the selected conformers. The crucial step in this analysis is computation of a SAXS profile from a proposed protein model.

Protein Structure Prediction

De novo protein structure prediction methods have two major components – a sampling algorithm and a scoring function. During the sampling phase, the protein model is perturbed. The protein is then scored, using a scoring function designed to identify native-like topologies. This process is iterated in order to minimize the scoring function. The challenge in this process is sampling the large conformational space of a protein densely enough so that one model approaches the native conformation. To be time-efficient, the protein model is often simplified to remove conformational degrees of freedom (coarse grained sampling) and the scoring function is therefore rapid but inaccurate. Sampling for larger proteins is further complicated by non-local contacts, amino acids in contact in Euclidean space (< 8Å), that are far apart in sequence (> 12 residues). As the number of non-local contacts increase, the accuracy of de novo protein structure prediction methods drastically decreases [49]. Atomic detail is added in a later stage of the protocol and the model is rescored / optimized with a higher accuracy scoring function. The accuracy necessary to identify the correct topology by its superior energy at this stage is a RMSD value of approximately 2Å when compared with the native structure.

BCL::Fold is designed to address the sampling bottleneck

BCL::Fold is a protein structure prediction method that rapidly assembles secondary structure elements (SSEs) into topologies.[50, 51] This approach provides a means to focus sampling on long range contacts between amino acid pairs. To begin, a pool of predicted SSEs is generated from an input FASTA sequence of amino acids. SSEs are randomly selected from the pool and assembled using a Monte Carlo Metropolis (MCM) assembly protocol to produce a coarse grained representation of the protein without side chain atoms and loop region residues. During assembly the model is evaluated using a consensus knowledge-based scoring function. This process is repeated 10,000 to 100,000 times. The underlying hypothesis of BCL::Fold is that the interactions between SSEs determine the majority of the protein core and give rise to its thermodynamic stability. Once the models have been generated, they are clustered by RMSD100 into N cluster centers. The medoid from each cluster center is selected for loop construction and side chain addition using Rosetta[52] to produce a set of proposed conformations for a given protein sequence in the absence of experimental data.

BCL::SAXS is a GPU accelerated Debye implementation for profile reconstruction

The use of experimental SAXS profiles during the construction of protein models with BCL::Fold would provide additional constraints on the sampling space of a given protein sequence. To incorporate experimental SAXS restraints into BCL::Fold, we must first develop a method to compare experimental SAXS profiles with profiles generated from protein models produced by BCL::Fold, i.e. missing loop region and side chain residues.

Here we describe our newly developed algorithm BCL::SAXS. It computes complete SAXS scattering profiles for complete protein models and an approximate scattering profile for protein models that consist of secondary structure elements only as used in BCL::Fold[50, 51, 53-55]. The main methods to calculate a SAXS scattering profile from atomic coordinates are spherical harmonics with multipole expansion, Monte Carlo methods, and the Debye formula [26, 28, 29, 53, 56]. Multipole expansion methods have been shown to be highly accurate, but difficult to modify for incomplete protein models. The Debye formula is easy to modify, but comes with a high computational cost. Ultimately we want to compare SAXS Profiles generated from BCL::Fold models [50, 51] – i.e. protein structure that lack loops and side chains – with experimental 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 graphical processing unit (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[19]. It models the hydration layer based on the solvent accessible surface area of each atom. To maximize the fit to experimental data BCL::SAXS optimizes the hydration layer density and the excluded volume of the protein. 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 consistent with the BCL::Fold folding protocol[50]: 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 set of 455 proteins with SAXS profiles computed *in silico* and experimental data from Hen Egg White Lysozyme. Furthermore, our work introduces a new approximation of the coordinates of residues in loop regions for crude protein models missing these residues. BCL::SAXS is available to the scientific community via the BCL::Commons user interface (www.meilerb.org). It is free for academic use.

Materials and Methods

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[19-21, 25, 28, 29]. The form factors are continuous functions of the magnitude of the momentum transfer vector $\vec{q}$. Using the Euclidean atomic coordinates from structures stored in the protein data bank (PDB)[57], scattering profiles are reconstructed. The following equations, starting with the Debye formula, depict the method:

$$I(q) = \sum_{i=1}^{M}\sum_{j=1}^{M} F_i(q)F_j(q)\frac{\sin(qr_{ij})}{qr_{ij}} \qquad 2.1$$

where the intensity, I(q) is a function of the magnitude of the momentum transfer vector $\vec{q}$. It is given by $|\vec{q}|$ = (4πsinθ) / λ, where θ is given by a scattering angle of 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 and added the contribution of the hydration layer:

$$F_i(q) = f_{v,i}(q) - c_1 f_{s,i}(q) + c_2 S_i f_{w,i}(q) \qquad 2.2$$

where $f_{v,i}(q)$ is the atomic form factor in vacuo, $f_{s,i}(q)$ is the form factor of the hypothetical atom that represents the displaced solvent[26] , and $f_{w,i}(q)$ is the contribution from the hydration layer. S_i is the solvent accessible surface area of the given atom. C_1 is used to modify the total excluded volume of the atoms and C_2 is used to modify the water density in the hydration shell. The atomic form factor in vacuo approximation is based on the combination of relativistic Dirac-Slater wave functions and numerical Hartree-Fock wave function [20-23, 53]. These Hartree-Fock scattering factors were previously computed from q = 0 to q = 1.5 at intervals of 0.01Å-1 [24]. For convenience, these scattering factors were previously fit to the 5-gaussian (Cromer-Mann) analytic function:

$$f_{v,i}(q) = \sum_{i=1}^{4} a_i \cdot e^{-b_i(\frac{q}{4\pi})^2} + c \qquad 2.3$$

where a, b, and c are the constants for each atom, and q is the momentum transfer vector. This approximation is only valid with a q range from 0 to 2.0Å[20-22] which is sufficient for SAXS scattering experiments where the valid scattering angle range is from 0 to ~0.33Å [3, 5]. For larger scattering angles, a 6-gaussian approximation must be used which is valid from 0 to ≈ 6.0 Å[23]. The displaced solvent scattering fs,i(q) was approximated by Vi [26], the excluded solvent volume V displaced by atom i:

$$f_{s,i}(q) = q_s V_i e^{\frac{-q^2 V_i^{2/3}}{4\pi}} \qquad 2.4$$

where qs is the solvent density of 0.334e Å-3 [25]. The combination of these equations yields a SAXS scattering profile from rigid body data stored in a pdb file.

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 q evaluated, where N represents the number of 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 circumvent this computational limitation, alternative approaches for this calculation including multipole expansion methods for spherical harmonics[26] and approximation of the individual form factors have been developed[28]. In contrast, to directly compute the SAXS profile using the Debye formula we leverage here the parallel architecture of graphical processing unit (GPU) threads using OpenCL and computed SAXS profiles directly.

GPU Implementations of the Debye Formula for SAXS Profile Reconstruction

In 2013, Antonov et al. showed how to use GPU acceleration to evaluate SAXS profiles in a Markov Chain Monte Carlo framework [58]. From a protein structure created in silico, they reconstructed the SAXS profile using the Debye formula and GPU Acceleration. To address the $O(N^2)$ complexity of the Debye formula they created a coarse grain representation of the protein model with a one or two-body "dummy atom" approximation for each residue. The two body representation required the development of 21 form factors to represent each new atom type – one for Alanine, one for Glycine, one for the Backbone, and 18 for the remaining side chains. These form factors were derived using a Monte Carlo simulation of a set of 297 high resolution crystal structures from the Protein Data Bank (PDB)[57, 59, 60] This algorithm was benchmarked on problem sizes ranging from 64 to 8192 scattering bodies. The speed up ranges from 16x to 394x. A protein represented by 1888 bodies with 51 discrete q values took 2408 ms on a central processing unit (CPU) and 9 ms with GPU acceleration.

BCL::SAXS GPU Implementations of the Debye Formula for SAXS Profile Reconstruction

To build upon the previous work we parameterize the excluded volume and hydration shell in the form factor calculation and operate on individual atoms. For full atom representations of proteins we can account for deviations in electron density and hydration shell thickness. 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 are 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 with the only limitation being memory. To address memory requirements, the algorithm was restructured to have each thread calculate a Debye partial sum for a current atom i:

$$I_{\mathrm{partial}} = F_i(q) \sum_{j=1}^{M} F_j(q) \frac{\sin(qr_{ij})}{qr_{ij}} \qquad 2.5$$

This technique enables the application of this accelerated algorithm to very large multimeric systems in excess of 90,000 atoms with the current GPU memory constraints 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 momentum transfer vector 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.

Generation of SAXS scattering profile from atomic coordinates with CRYSOL

To measure the time the algorithm takes on different types of GPUs, experimental scattering curves were approximated from high resolution protein structures in the PDB using the program CRYSOL[26]. This program computes the scattering profile using spherical harmonics and multipole expansion for fast calculation of the spherically averaged scattering profile.

Approximate SAXS scattering profiles for protein models without side chain and loop regions

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 Cβ position of the respective amino acid. This approach is analogous to how the form factors for hydrogen are folded into their respective heavy atom in CRYSOL[26]. The loop regions were approximated by removing atomic coordinate data between secondary structure elements (SSEs) and computing a path from the c-terminus of the first SSE to the n-terminus of the second SSE. The amino acid residues in the loop regions were placed at points along the path (Figure 6). While crude, this approach is much more rapid than actual construction of loops.

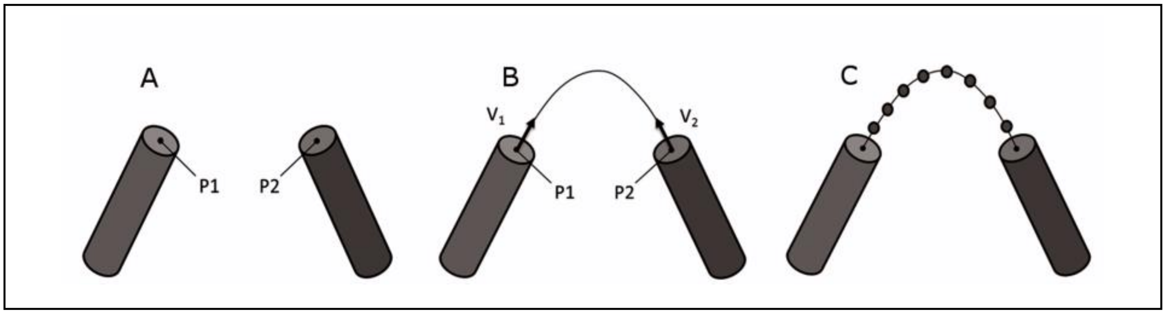


Figure 6: Construction of curvilinear path and placement of residues in region between two SSEs (A) Protein model with two α-helical structures, p1 and p2. (B) Approximated path with unit vectors v1 and v2 pointing in the helical direction of SSE1 and the helical direction of SSE2 (C) Residues placed equidistant along the curvilinear path between SSEs.

Vector calculations to approximate the loop path between two secondary structure elements

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

$$\overrightarrow{P_{1,n}} = [x_1, y_1, z_1] \qquad 2.6$$

$$\overrightarrow{P_{2,c}} = [x_1, y_1, z_1] \qquad 2.7$$

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

$$\overrightarrow{CP_{1,n}} = [x_2, y_2, z_2] \qquad 2.8$$

$$\overrightarrow{CP_{2,c}} = [x_2, y_2, z_2] \qquad 2.9$$

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

$$\overrightarrow{V_n} = \overrightarrow{P_n} - \overrightarrow{CP_n} \qquad 2.10$$

where n is the index of the point. The direction of the vectors V1 and V2 were computed by dividing them by their magnitude.

$$\overrightarrow{\mathrm{D_n}} = \frac{\overrightarrow{\mathrm{V_n}}}{\sqrt{\mathrm{V_{nx}^2} + \mathrm{V_{ny}^2} + \mathrm{V_{nz}^2}}} \qquad 2.11$$

The scalar 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 percentage to move from P_1 toward P_2 at each step (L) along path (S) was computed by dividing one by one more than the number of amino acids in the loop region.

$$\mathrm{L} = \frac{1}{\mathrm{N_{aa}} + 1} \qquad 2.12$$

The predicted Euclidean loop length (P) was computed by multiplying the number of amino acids by the Cα – Cα spacing of 3.2 Å. The 3.2 Å term is the average distance between amino acids in the coil region of a protein. It was computed by averaging the Cα distance between residues in the engrailed homeodomain (pdb id: 1ENH)[61].

$$\mathrm{P} = \mathrm{N_{aa}} \times 3.2\text{Å} \qquad 2.13$$

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:

$$\overrightarrow{\mathrm{l(L)}} = (1 - \mathrm{L})\vec{\mathrm{P}}_1 + \mathrm{L}\vec{\mathrm{P}}_2 \qquad 2.14$$

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

$$\mathrm{p(L)} = \mathrm{N} \times \mathrm{L}(1 - \mathrm{L}) \qquad 2.15$$

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

$$\overrightarrow{\mathrm{d(L)}} = [(1-\mathrm{L})\overrightarrow{\mathrm{d_1}} + \mathrm{L}\overrightarrow{\mathrm{d_2}}\,] \qquad 2.16$$

where d_1 and d_2 are unit directional vectors pointing in the direction of SSE_1 and SSE_2 respectively. The complete parabolic approximation function is:

$$\overrightarrow{\mathrm{P(L)}} = (1-\mathrm{L})\overrightarrow{\mathrm{P_1}} + \mathrm{L}\overrightarrow{\mathrm{P_2}}\; + \mathrm{NL}(1-\mathrm{L}) \times [(1-\mathrm{L})\overrightarrow{\mathrm{d_1}} + \mathrm{L}\overrightarrow{\mathrm{d_2}}\,] \qquad 2.17$$

Normalization Factor and Path Length Calculations

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 (Figure 7). With these approximations, the normalization factor (N) is given by the Pythagorean Theorem:

$$\mathrm{N} = \frac{1}{2}\sqrt{\mathrm{P}^2 - \mathrm{D}_{\mathrm{sse}}^2} \qquad 2.18$$

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.

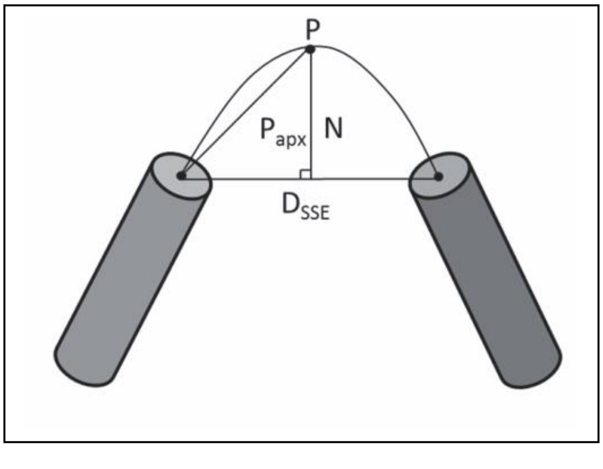


Figure 7: Depiction of the parabolic height approximation method. D_{sse} is the Euclidean distance between SSEs, P_{apx} is the estimated length of the hypotenuse side of a right triangle. N is the normalization factor and controls the height of the parabola.

Model quality was assessed by the χ agreement between the calculated and experimental SAXS curves

To compare the scattering profiles, we first normalized the experimental and calculated scattering intensities to be between (0, 1]. To magnify the effects of small distances, (higher q values), the scattering intensities (I) for both data sets were converted to a log10 scale. To account for concentration differences in experimental data, the calculated curve was multiplied by a scaling weight (c) that minimizes the χ score[26, 29].

$$c=\left[\sum_{k=1}^{Q}\frac{I_{cal}(q_k)\cdot I_{exp}(q_k)}{\sigma_{exp}^{2}(q_k)}\right]\left[\sum_{k=1}^{Q}\frac{I_{m}^{2}(q_k)}{\sigma_{exp}^{2}(q_k)}\right]^{-1} \qquad 2.19$$

where I_{cal} is the intensity of the calculated curve, I_{exp} is the intensity of the experimental curve, σ is the experimental error and q is the momentum transfer vector. 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[8, 53, 62, 63], we score a model based on the χ score between the experimental profile and the profile computed by our algorithm BCL::SAXS.

$$\chi=\sqrt{\frac{1}{Q}\sum_{i=1}^{q}\left(\frac{I_{exp}(q_i)-cI_{cal}(q_i)}{\sigma(q_i)}\right)^{2}} \qquad 2.20$$

where Q is the number of entries in the data set and σ is the experimental error of the measured profile. In cases where no experimental error is provided it is simulated. We compute the χ score from different states of the experimental and calculated scattering profiles. The first state on the absolute scale is to compute the χ score right after the initial profile reconstruction with the Debye formula and scaling. The second state is to compute the χ score after converting the both experimental and computed data to the log10 scale. The third state is to compute the

χ score after taking the derivative of the log10 representation of the experimental and calculated curves.

For complete models, we identify the optimal χ values by optimizing combinations of the excluded volume parameter, C_1 and the hydration layer parameter, C_2 inside a boundary ($0.8 \leq C_1 \leq 1.2$ and $0 \leq C_2 \leq 4.0$). Using these parameters we compute the scaling parameter c that minimizes χ for each C_1, C_2 combination.

Results

To illustrate the use of BCL::SAXS, we show the results using hen egg white lysozyme (PDB ID: 6LYZ, molecular weight 14 kDa). The X-Ray scattering results for this protein were obtained from an open access database, BIOISIS, containing experimental SAXS data for hen egg white lysozyme (BIOSIS ID: LYSOZP). The SAXS profile for this protein was collected at the SIBYLS Beamline ASL BL12.3.1 and the experimental setup has been previously described [64]. To account for uncertainty in the PDB definitions of secondary structure of 6LYZ, we added additional SSEs by taking the consensus prediction of the secondary structure server 2Struc[65]. This meta server runs secondary structure prediction using the Dictionary of Secondary Structure of Proteins (DSSP)[66], DSSPcont[67], Stride[68], P-SEA[69], PALSSE[70], STICK[71], KAKSI[72]and TM-Align[73]. The final SSE definitions used for analysis are shown in Table 3. The final model with loop approximations is shown in figure 8.

Table 3 SSE Definitions for Hen Egg White Lysozyme

Type	SSE Number	Start Residue	Sequence Location	End Residue	Sequence Location
Helix	1	ARG	5	HIS	15
Helix	2	LEU	25	SER	36
Helix	3	CYS	80	LEU	84
Helix	4	ILE	88	ASP	101
Helix	5	VAL	109	CYS	115
Helix	6	ASP	119	ARG	125
Strand	1	LYS	1	PHE	3
Strand	2	PHE	38	THR	40
Strand	3	ALA	42	ASN	46
Strand	4	SER	50	GLY	54
Strand	5	GLN	57	SER	60

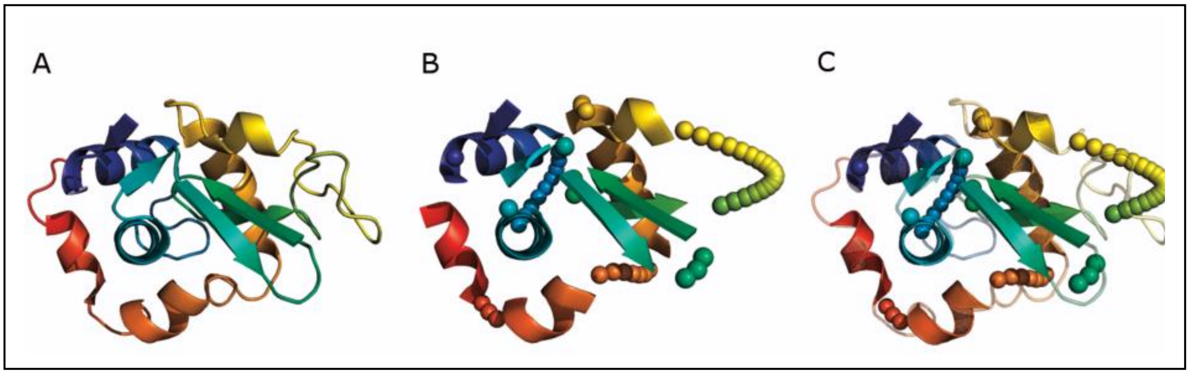


Figure 8: Depiction of Hen Egg White Lysozyme PDBid: 6lyz (A) The crystal structure of Lysozyme with the n-terminal region colored blue and the c terminal region colored red. (B) Depiction of the native structure with the loop regions removed and approximated by pseudo atoms along the curvilinear path between SSEs. (C) Overlay of the native and approximated version of Hen Egg White Lysozyme.

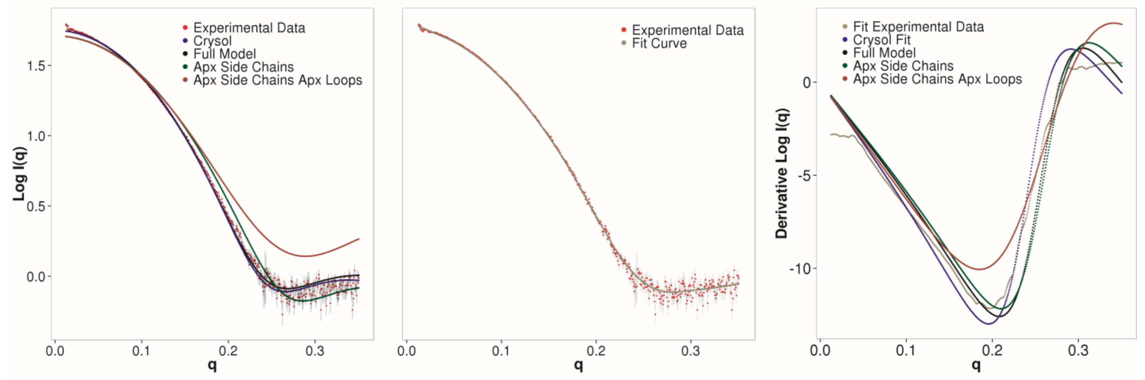


Figure 9: Depiction of the Experimental SAXS profile for Hen Egg White Lysozyme and SAXS profiles computed with BCL::SAXS for different protein states Panel A (left) represents the fit on a log10 scale with Experimental data being the SAXS profile or Hen Egg White lysozyme, Crysol is the curve generated through Crysol from 6lyz and fit to the experimental data. Full Model is the curve generated through BCL::SAXS from 6lyz. Apx Side Chains is the curve generated through BCL::SAXS using Backbone atoms only and summing the form factors for all side chain atoms at the Cβ coordinate of the residue. Apx Side Chains Apx Loops is the curve generated through BCL::SAXS using loop approximation and side chain approximation. Panel B (middle) shows the locally weighted scatterplot smoothing (LOESS) of the experimental SAXS data points. Panel C (right) shows the fit of previous data types from panel A using the derivative of the log10 profiles.

Table 4: Chi Scores comparing experimental SAXS data for Hen Egg White Lysozyme with profiles generated from the crystal structure (6LYZ) for CRYSOL and BCL::SAXS.

Type	$\text{Log}_{10}\,\chi$	Derivative χ
Crysol	2.81	0.96
BCL::SAXS Full Model	2.32	1.01
BCL::SAXS Apx Side Chains	9.16	1.17
BCL::SAXS Apx Side Chains and Loops	19.83	1.25

The SAXS comparison derivative χ score

When comparing SAXS profiles between two distinct proteins, the common method is to use the χ formula previously shown [26, 29, 30]. However, when computing a SAXS profile for models with approximate the side chain atoms and loop regions, we observe a systematic upward shift from the original I(q) profile (Figure 9A). This shift between the experimental and approximated profiles increases the rate of false positive identification by SAXS scores (Figure 8). We observe also that minima and maxima of the I(q) profile are less affected. Therefore, by comparing the derivative of the profiles, we take the shape of the SAXS profile into account which decreases the rate of false positive identification by SAXS score.

For this derivative comparison, a curve was fit through the experimental data points using locally weighted scatterplot smoothing (LOESS)[74, 75] using a span of 0.2 and a polynomial degree of 1 in R. The span variable determines how much of the data is used to fit each local polynomial. A large span produces the smoothest function while the smaller the span, the closer the regression will conform to the data. Splines were used to numerically differentiate the fit profile. The derivative results and scores are shown in figure 9 and table 4. To measure the similarity between an experimental SAXS profile and complete protein models, we use the standard χ score. By using this score, we can easily compare our method with other established methods in the field such as CRYSOL. The user can specify what metric to use during analysis.

Non-redundant dataset for protein discrimination benchmark

To determine how well the SAXS score can distinguish protein folds from each other, we evaluated a representative subset of 455 proteins with a 20% identify cutoff, 1.6 Å resolution cutoff, and 0.25 R-factor cutoff from the PISCES databank[76, 77]. 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.

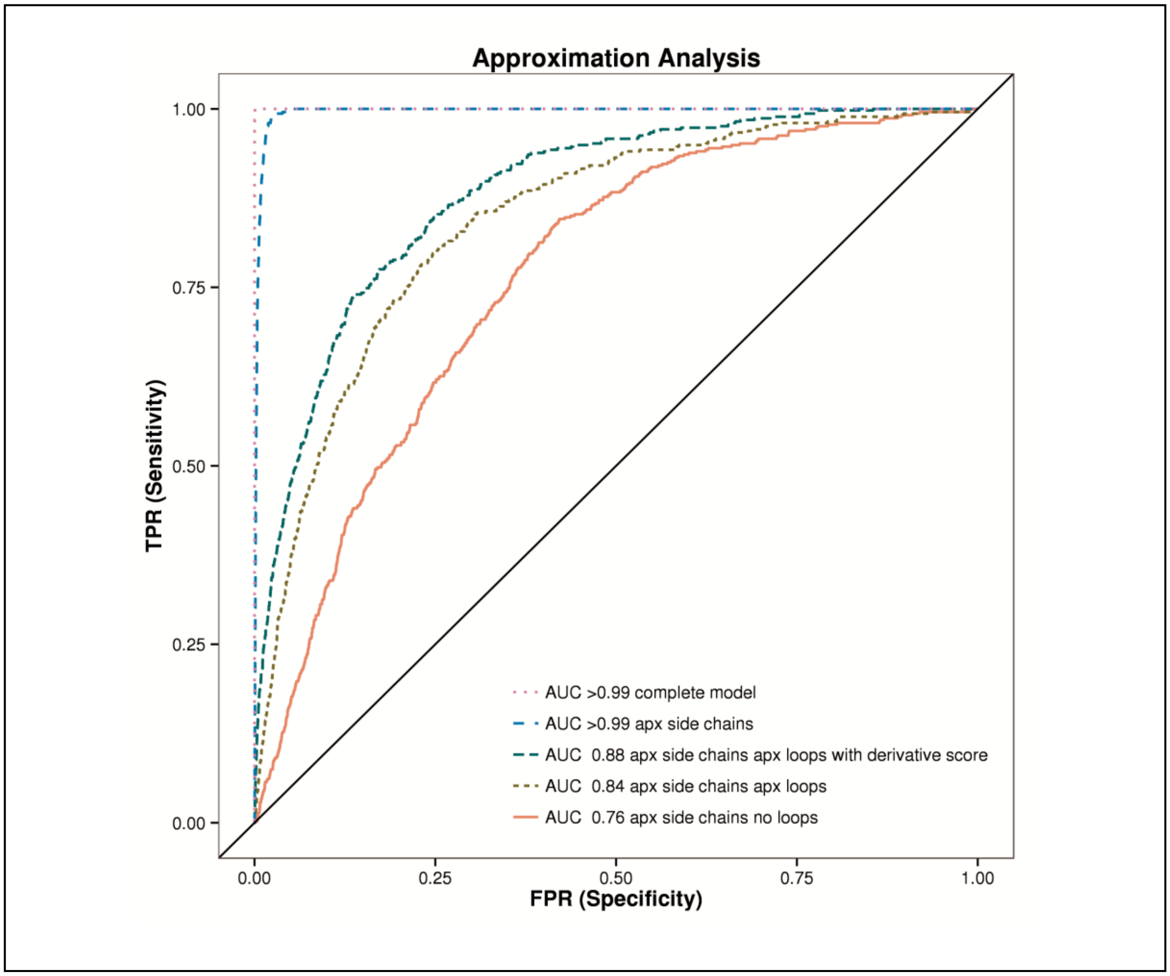


Figure 10: ROC Analysis of 455 proteins from Pisces dataset in different states. The area under the curve (AUC) is shown with BCL::SAXS profiles generated for complete protein models (orange), models with approximated side chains (purple), approximated side chains and with loop approximation method (blue), approximated side chains without loop approximation method (red), and the derivative of the approximated side chains with the loop approximation method (green). The standard χ score was used to compare the profiles for all plots except for green, where the derivative χ score was used.

The area under the curve (AUC) for complete protein models is > 99%. When side chains are removed, the AUC remains > 99%. The AUC for proteins without side chains and loop regions is 76%. When loop regions are approximated, the AUC is 84%. The derivative score improves the AUC to 88%. See figure 5. 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.

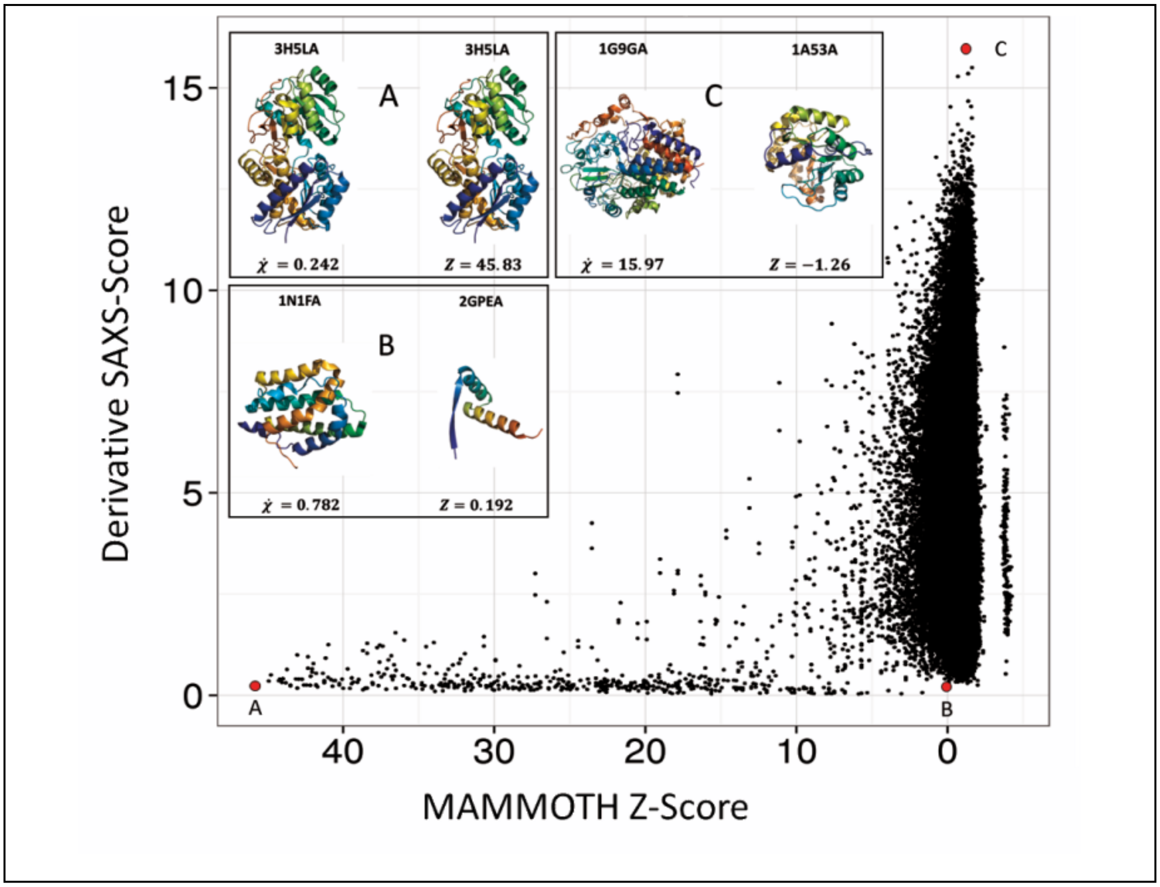


Figure 11 Structural MAMMOTH Z-score vs. SAXS profile similarity score of 455 proteins from Pisces dataset: All 455 proteins were scored by structural similarity to each other with self-pairing receiving the highest z-score (x-axis). SAXS profiles for all 455 proteins were generated and the χ score between all scores was computed (y-axis). Panels A, B, and C correlate with their respective red dot. Panel A depicts 3H5LA paired with itself. Panel B depicts 1N1FA paired with 2GPEA. Panel C depicts 1G9GA paired with 1A53A. The derivative χ score was used to compare the 455 SAXS profiles.

The 455 x 455 matrix was used to score the structural similarity of a pair of proteins. The diagonal represents self-paired proteins. The higher the Z-score, the more similar the two structures are. A Z-score below four indicates that two proteins are structurally different. In the SAXS analysis, a lower SAXS score indicates the scattering profiles of two proteins are very similar. In this analysis, a high Z-Score and a low SAXS score indicate that proteins identified by SAXS as similar are structurally similar. Figure 11 panel A depicts 3H5L chain A (molecular weight 44.92 kDA) paired with a copy of itself. As expected the SAXS similarity score is very low and the Z-score is high. Interestingly, panel B depicts 1N1F chain A (molecular weight 18.35 kDA) paired with 2GPE chain A (molecular weight 5.95 kDA). Although there is a difference of 12.4 kDA, the SAXS score indicates that the proteins are similar. Figure 11 shows that structurally similar proteins (high Mammoth Z-score) always 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 can appear similar in a SAXS experiment if their overall shape is similar.

SAXS Degeneracy in the scattering profile

During elastic scattering, energy is conserved between incident X-Rays that scatter by interactions with electrons in the target sample. The magnitude of the wave vector $\vec{k}$ for both the incident and scattered wave is given by 2π / λ. The change in wave-vector is only in direction and the difference between $\overrightarrow{k_i}$ and $\overrightarrow{k_f}$ is given by $\vec{q}$ – the momentum transfer vector. The X-ray scattering amplitude at $\vec{q}$ by a particle at position $\overrightarrow{r_j}$ is given by:

$$A_j(\vec{q}) = f(q)e^{(i\vec{q}\cdot\vec{r}_j)} \qquad 2.21$$

where f is the form factor for the atom j at a magnitude for q given by 4πsinθ / λ. The form factor decreases from a maximum at q = 0. At this q value, the form factor is equivalent to the atomic number Z of the atom. Hence, atoms with higher Z are stronger scatterers. The amplitude for an ensemble of particles is a summation of the amplitudes of all particles:

$$A(\vec{q}) = \sum_{j=1}^{n} f(q)\, e^{(i\vec{q}\cdot\vec{r}_j)} \qquad 2.22$$

The scattering intensity is given by the amplitude multiplied by its complex conjugate A($\vec{q}$)*:

$$I(\vec{q}) = A(\vec{q})A(\vec{q})^* \qquad 2.23$$

The observed scattering pattern is not the complex amplitude function. It is the modulus squared of the amplitude function. Most of the structural information obtained from X-ray scattering experiments reside in the phase of the wave-function. This phase information is stored in the imaginary part of the amplitude function and is lost when multiplied by the complex conjugate. This loss of phase information results in a loss of structural uniqueness. Furthermore the effect is compounded because during a SAXS experiment samples are free to rotate. The observed I(q) function is therefore also an average over possible orientations. The loss of orientation and phase information results in the degeneracy in the scatting profile (multiple structures yielding similar SAXS profiles) as observed in figure 11.

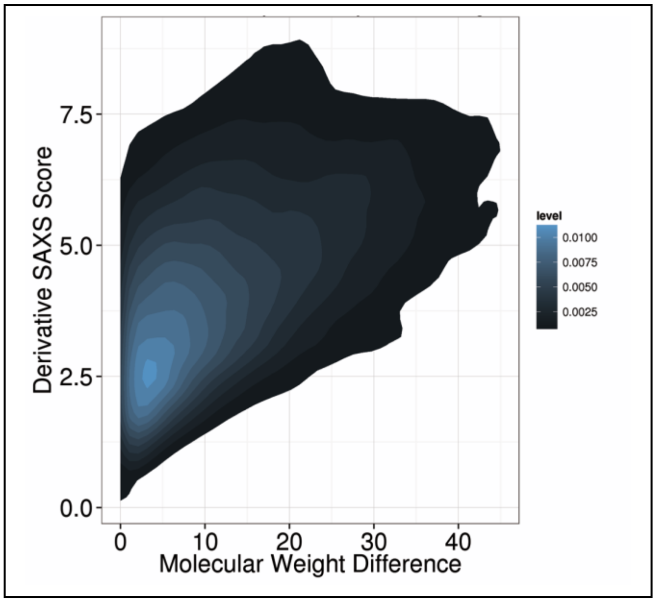
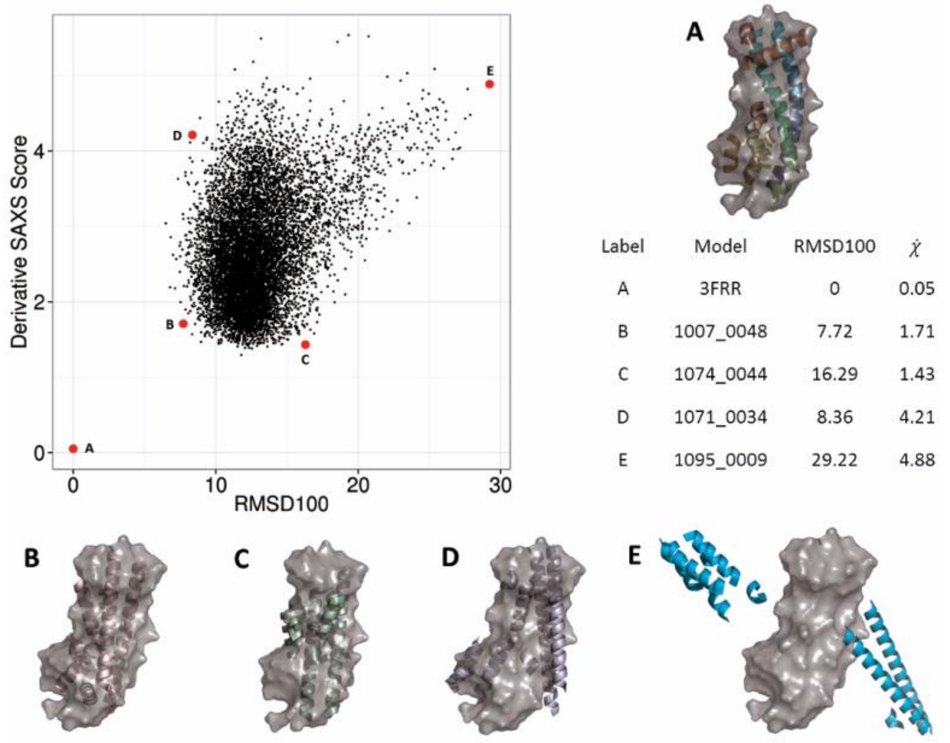


Figure 12: The SAXS similarity scores χ in relation to molecular weight difference: Molecular weights for all 455 proteins from the PISCES data set were calculated. The absolute value of the difference in weight between two proteins was computed for all pairs. The density plot depicts the difference in molecular weight on the x-axis and the derivative SAXS similarity score χ on the y-axis.

n

To show the relation between the molecular weight of the compared proteins and the similarity of the SAXS profiles, we calculated molecular weights for all 455 proteins in the PISCES data set used in the MAMMTOH analysis. We then combined the molecular weight difference with the derivative SAXS score to generate a density plot (Figure 12). As expected, we observe that for proteins of similar molecular weight a range of SAXS similarity scores χ are possible from very similar to dissimilar determined solely by the similarity in overall shape. As the difference in molecular weight increases, the minimum SAXS similarity scores χ increases also, i.e. structures with large molecular weight differences do not have similar SAXS profiles, even if the overall shape is similar.



Label	Model	RMSD100	$\dot{\chi}$
A	3FRR	0	0.05
B	1007_0048	7.72	1.71
C	1074_0044	16.29	1.43
D	1071_0034	8.36	4.21
E	1095_0009	29.22	4.88

Figure 13: BCL::SAXS was used to score 10,000 protein confirmations of 3FRRA generated by BCL::Fold. In each case the surface of the native confirmations is shown in gray. Each black dot represents one model. The red dots labeled with A,B,C,D,E show examples of different conformations sampled by BLC::Fold and their respective scores. The derivative χ score was used to compare the 10,000 BCL models with the native structure.

Scoring BCL::Models with SAXS

BCL::Fold was run to generate 10,000 protein models of 3FRR. These models were only comprised of secondary structure elements. Using the side chain and loop region approximations, BCL::SAXS was used to construct SAXS profiles for all 10,000 models generated by BCL::Fold. (Figure 13) From this figure, we observe that the correct topology has a very low SAXS score. We note that model C has a lower SAXS score (1.43) than model B (1.71) although model B has a much lower RMSD100 score (7.72) than model C (16.29). This behavior is expected because SAXS cannot distinguish topologies that fit inside the overall SAXS envelope. Agreement with by SAXS score is a necessary condition for correct protein identification, but not sufficient to uniquely identify the correct model. However, because of this, the SAXS score can be used as a filter to remove models that score above a threshold.

GPU Algorithm Yields Orders of Magnitude Speed Improvements

The GPU accelerated Debye calculation was benchmarked on several protein systems from the PDB with sizes ranging from 1,800 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 (GTX680). See Table 5. The speed was determined by measuring the time in seconds from the start of the Debye formula to the SAXS profile return from the Debye formula. The Maximum Speed up is the maximum of the ratio of the CPU time in seconds divided by the GPU time in seconds.

Discussion

We have demonstrated how to compute SAXS profiles from atomic coordinates. In our approach for complete protein models we did not make approximations to the Debye formula, rather we used GPU acceleration to handle the double summation of all atoms and used the Hartree-Fock scattering factors directly. For proteins of sizes ranging from 1832 atoms to 91,846 atoms we find, as expected, that without GPU acceleration, the $O(N^2)$ computational cost of the Debye formula results in a significant slow-down when compared to the $O(q^2D^2N)$ algorithm implemented in CRYSOL (Table 5). The magnitude of the momentum transfer vector is given by q and D is the max dimension of the macromolecule. With GPU acceleration computation times are comparable. The GPU card that gave the best performance was GTX680.

Table 5 Timing results of GPU vs. CPU benchmarks. All timings are reported in seconds. Crysol reported timings to the nearest second. The first two measurements were not accurate and have been omitted.

PDB	Atoms	BCL::SAS	Crysol	Quadro	GTX470	GTX480	GTX580	GTX6	C1060	Max
		CPU $1300		**600**	$325	$325	$550	**80**	$1300	**Speedup**
				$225				$1000		
1O26A	1832	3.6	-	0.1	0.07	0.07	0.07	0.07	0.09	51x
1WA5C	7543	65	-	1	0.31	0.28	0.27	0.20	0.37	325x
1NR1	23217	624	2	9.3	2	1.9	1.8	1.2	2.7	520x
1ZUM	43243	2300	5	30	4.9	4.1	3.9	2.4	6.5	958x
1VSZ	91846	15365	10	132	19.8	16.9	15.8	9.0	26.3	1707x

In order to compare experimental scattering profiles with approximated profiles we computed the first derivative of the profiles and then computed the similarity score ($\dot{\chi}$), between the derivatives of the SAXS profiles. This enabled us to reduce the amount of false positives obtained during our analysis and improve the accuracy in structure identification using SAXS profiles from 84% to 88%. BCL::SAXS was >99% accurate in picking the native protein from a set of other proteins when using complete proteins from the PDB and using the standard χ comparison score. With the side chains approximated, BCL::SAXS remained >99% accurate in picking the native protein from a set of other proteins. With the loop regions removed, the accuracy dropped from >99% to 76%. This result shows that loop regions play an important role in defining overall protein shape. Using our loop approximation algorithm and the derivative of the χ score, the accuracy increased to 88%. 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 rigid bodies.

The MAMMOTH analysis shows that proteins with very similar z-scores (structurally similar proteins) also have a low SAXS $\dot{\chi}$ 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.

Conclusion

We explored the idea of approximating the SAXS score for protein models without side chain and loop coordinates by placing dummy atoms along a path between secondary structure elements. The SAXS profile can be used to distinguish different proteins from each other, but cannot be used exclusively to distinguish different permutations of the same topology. However, the SAXS profile can be used as a filter to exclude protein models that are very different from the native from further analysis as a filter.

Acknowledgements

We would like to thank Mariusz Butkiewicz, and Jeff Mendenhall for their insight and assistance throughout the development of BCL::SAXS. We acknowledge Oak Ridge National Laboratory supercomputing resources on TITAN. DKP is supported through a NLM 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).

CHAPTER III

CASP10-BCL::FOLD EFFICIENTLY SAMPLE TOPOLOGIES OF LARGE PROTEINS

Overview

This chapter is a reproduction of CASP10-BCL::Fold efficiently samples topologies of large proteins [54]. In this paper I was responsible for writing and referencing all sections except the topology score description, beta sheet alignment, and conclusion section. I created figures: 11, 12, 13, 16, 17, 18, 21A, 21C, 22, and table 8. Jens Meiler, Sten Heinze, and Axel Fischer provided insight during the writing process. Tim Kohlman and Sten Heinze generated data for the tables.

Significance

In relation to my thesis this manuscript details the limitations of BCL::Fold within the folding pipeline and process (Table 9). SAS data can be used to improve the attrition depicted in Table 8. *De Novo* protein structure prediction remains a challenging prospect. The average CASP10 free modeling (n=20) GDT_TS score for the top scoring methods was 33%. For 3 models in this category, BCL::Fold was able to sample models with 34% accuracy by GDT_TS score. The sampling of BCL::Fold is as accurate as other methods final model selection.

Innovation

This work shows that BCL::Fold samples native-like topologies for some proteins, but the scoring function does not identify these topologies as favorable and are not selected for further analysis. This paper forms the basis for my claim that Small Angle X-Ray Scattering could be used to improve model selection in the BCL::fold pipeline: 1) Filter SSE arrangements produced by BCL::Fold by SAXS score agreement, 2) Incorporate SAXS restraint in the scoring function of BCL::Fold to penalize models during folding if they violate the restraint.

Experimental structures in the protein data bank (PDB) are biased toward small soluble proteins

The tertiary structure of a protein provides essential insights to its biological function in living organisms. Accordingly, experimental methods are applied to ascertain protein structure including X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy Currently, the PCB contains more than 89,258 proteins (December 2013) of which 79,585 (89%) were elucidated by X-ray crystallography, 8971 (10%) by NRM, and the remainder by other technologies.[59] Despite these efforts, the structures represented in the PDB are biased; 87,004 of the proteins in the PDB are soluble while only 2254 (2.5%) of the proteins represent membrane proteins.[59] Further, the size distribution of proteins in the PDB is biased toward small proteins omitting many large macromolecular assemblies greater than 500,000 Da (2.0%).[59, 60, 79] This bias is due to the limitations of experimental methods for structure determination. Membrane proteins are underrepresented in the PDB because they are too large for NMR and their embedding in the two-dimensional membrane complicates formation of three-dimensional crystals required in X-ray crystallography.[44] For membrane proteins up to ~1000 folds remain to be determined. [55, 80] Large macromolecular assemblies are also underrepresented in the PDB because its protomers do not fold in isolation, they are difficult to crystallize, and they are too large for NMR spectroscopic methods. [81] Thus, for many biologically relevant proteins only limited experimental data can be collected with a combination of experimental techniques such as solid state NMR, cryo-electron microscopy, electron paramagnetic resonance, mass spectrometry, and small angle X-Ray scattering. On their own, these datasets are insufficient for atomic-detail structure determination. One major justification to develop de novo protein structure prediction algorithms is to complement such limited experimental datasets.

De novo protein structure prediction needs a reduced search space

The cornerstone of *de novo* protein structure prediction methods is based on the assumption that (most) folded proteins exist in their lowest energy conformations. [82] Protein folding becomes an energy minimization process that depends on interaction of amino acids with the environment and other amino acids in the sequence. Finding the global minimum of the energy function on the energy landscape is challenging for several reasons including that the energy landscape contains many local minima. Currently, no universal method of identifying the global minimum of the energy function exists. [83] In practice, the conformational space of a protein is also far too large to be comprehensively searched with a highly accurate and therefore slow to compute energy function. Therefore, the conformational space is reduced by working with simplified protein representations, at least in the initial folding simulation. In effect, this reduces the resolution of the energy function which allows a more rapid calculation but decreases its accuracy to the point where the global energy minimum cannot be unambiguously detected and several local energy minima need to be considered.

Competing *de novo* structure prediction software reduces the search space similarly. Rosetta addresses the sampling challenge by assembling protein models from three and nine residue peptide fragments. [52, 84, 85] These fragments are determined from peptides of similar sequence and secondary structure extracted from other proteins in the PDB. For proteins smaller than 80 residues Rosetta was able to predict atomic detail models in the absence of any experimental restraints for about 30% of the test cases. [86] For larger proteins up to around 150-180 residues Rosetta samples the correct topology about 50% of the time.

[86-88] Generally, Rosetta tends to perform better for α-helical proteins which are related to their reduced fold complexity. The complexity of a fold can be measured by contact order (CO) which is defined as the average sequence separation of residues in contact, that is, residues whose C_β atoms are < 8 Å apart. [89, 90] As the complexity of protein topology increases (high CO) the accuracy of the Rosetta prediction decreases. [49, 89, 90]

I-Tasser threads the target sequences through a library of PDB structures with a pair-wise sequence identity cut-off of 70% to search for plausible protein folds. Rather than using a fixed set of three and nine residue peptide fragments, I-Tasser uses fragments of variable size that are identified by threading. [91-93] The fragments are used to reassemble full-length models while the loop regions between fragments being constructed *de novo*. Critical to the success of I-Tasser is the identification of suitable templates to create the peptide fragments – a Pearson correlation coefficient of 0.89 for RMSD and 0.95 for TM-score. [93] Generally, I-Tasser samples the correct topology about a third of the time for proteins up to 155 residues long with RMSD < 6.5 Å. [93] I-Tasser shares the most critical limitation with Rosetta, the ready formation of long-range interactions between residues.

BCL::Fold was designed to overcome size and complexity limitations in *de novo* protein structure prediction methods

BCL::Fold is a de novo protein structure prediction algorithm based on the placement of disconnected secondary structure elements (SSEs) in three-dimensional space as previously published. [50, 51, 55] This algorithm was developed to test the hypothesis that for many proteins the core responsible for thermodynamic stability is largely formed by SSEs. In this case, likely protein topologies could be detected from SSE-only models. Thereby, the size and CO restrictions in protein structure prediction can be overcome by assembling disconnected, rather rigid SSEs, reducing the search space substantially and allowing the ready formation of nonlocal contacts. [50] A coarse grained knowledge based energy function identifies native-like SSE arrangements using a Monte Carlo simulated annealing sampling algorithm with metropolis criterion. [50, 51, 55] In contrast to I-Tasser or Rosetta, this algorithm is truly de novo as no fragments from the PDB are used. Loop regions between SSEs and side chains atoms are added to the model in subsequent steps using for example Rosetta. [94-96]

BCL::Fold uses a consensus of secondary structure prediction technologies to identify SSEs

Critical to the success of the BCL::Fold algorithm is the correct prediction of SSEs: α-helices, β-strands, coil regions, and trans-membrane spans from sequence. These predictions are obtained from a consensus prediction from PHD [97, 98], PsiPred [99, 100], and Jufo9D [53, 101-103] for soluble proteins. In addition to these methods we used Octopus [104, 105] and Jufo9D [101] for the trans-membrane span region of membrane proteins. The consensus prediction is used to build a pool of SSEs, which is input for protein folding.

A Monte Carlo Metropolis sampling algorithm positions SSEs in space

Protein models are assembled using a Monte Carlo sampling algorithm. Each iteration of the algorithm consists of a randomly selected modification to the current model. Modifications include the addition of an SSE from the SSE pool to the model; the removal of an SSE from the model; translational and rotational transformations of SSEs in the model; swapping of two SSEs; modifications of groups of SSEs (domains) consist of translating the domain; flipping; and shuffling the different SSEs.

After each modification, the model is evaluated by a knowledge based scoring function. [51]. This coarse grained scoring function is designed to evaluate the arrangement of SSEs in Euclidean space. It is a weighted sum of scoring terms that represent different aspects of SSEs of protein structures as observed in experimental structures like the preferred environment of amino acid types (buried or solvent exposed); the radius of gyration; an SSE packing and a strand pairing potential; a loop length potential; clash terms for amino acids and SSes; and a loop closure penalty. The loop closure penalty limits the Euclidean distance between two consecutive SSEs to the maximum length a stretched out amino acid chain can bridge and applies a steep penalty for longer loop distances.

The evaluation with the Metropolis criterion results in one of four possible outcomes: 1) improved and accepted, if the calculated energy score is lower than the energy of the previous model; (2) accepted by the Metropolis criterion with a function taking the energy difference and the simulated temperature into account; (3) rejected if the score is higher than the previous model and rejected by the Metropolis criterion; (4) skipped, if the modification is not applicable to the model, for example swapping SSEs if the model contains only a single SSE. The probability of a step being accepted with higher energy is based on the temperature used by the Metropolis criterion. BCL::Fold adjusts the temperature to achieve a ratio of accepted steps that reduces from 50 to 20% in the course of the simulation.

All scoring terms (except for the clash terms and the loop closure penalty) are statistically derived using Bayes' theorem from a divergent high resolution subset of the PDB generated by the PISCES server with a maximum sequence identity of 25% [76, 77] and then energies were approximated using the inverse Boltzmann relation.

The algorithm will continue generating modified models and evaluating them until a maximum number of 2000 steps was completed or no improvement in the score was found to 400 consecutive steps; this constitutes one folding stage. The folding process of one model has five assembly stages and one refinement stage, which employ a decreasing number of modifications for large scale perturbations (for example, swapping SSEs) and an increasing amount of small scale perturbations (for example, bending an SSE). The lowers energy models within the trajectory will be saved as resulting model for this run.

The CASP10 experiment: a critical tool for development of techniques for protein structure prediction

To evaluate the accuracy of BCL::Fold in *de novo* protein structure prediction, we participated in the Critical Assessment of protein Structure Prediction (CASP10) experiment, which is held every two years. [106, 107] The double-blind experiment tests protein structure prediction methods objectively because the experimentally determined structure is withheld from predictors, organizers and the assessors until the experiment is finished. After protein predictions have been made, the experimentally determined structures are revealed and the results are assessed. CASP10 contained the following categories: (1) Tertiary structure prediction, which can be classified as: (a) Template Based Modeling (TBM): starting from a homologous protein template in the PDB. (b) Free modeling (FM): no homologous template exists in the PDB; (2) Tertiary structure prediction with limited experimental information, for example, amino acids in contact [108]; (3) Residue-residue contact prediction [109] (4) Model refinement [110]; (5) Identification of disordered regions; (6) Function prediction; (7) Quality assessment. [111]

To maximally leverage CASP10 for testing BCL::Fold we assume all CASP10 targets to be FM targets

For some targets, templates can be found, that is, proteins with similar sequence and known structure that can guide the prediction. Based on if templates can be found and how similar the template structure is to the target structure, measured by the Global Distance

Test/Total Score (GDT_TS) [112], prediction for CASP10 gargets is categorized as easy or hard TBM (easy if the maximal GDT_TS ≥ 50, hard if the maximal GDT_TS < 50), FM or a combination of both (TBM/FM). The GDT_TS could obviously only be employed after the target structures were available; in the prediction process other measures like sequence similarity to proteins in the PDB were used to classify targets. To maximize the assessment of the BCL::Fold *de novo* protein structure prediction algorithm in CASP10 we treated all targets as FM targets, that is no homologous template from the PDB was used at any point as input into the BCL::Fold prediction algorithm.

Table 6: Clustering Statistics of CASP10 Targets folded by BCL::Fold

Target	Folded models	After filtering	Top cluster	Top scoring	Top homology
T0644	9980	4485	2	0	0
T0649	10,000	5135	3	5	0
T0655	9980	4335	1	3	2
T0663	12,000	6495	3	2	3
T0666	12,000	5979	3	3	0
T0676	12,000	6341	3	0	1
T0678	12,000	6271	5	1	2
T0682	12,000	5554	0	3	4
T0684	12,000	5884	16	0	1
T0686	12,000	6230	2	1	0
T0691	12,000	6083	4	1	2
T0700	12,000	6605	1	2	3
T0704	12,000	5932	1	3	1
T0720	12,000	6345	2	2	1
T0722	12,000	8747	1	2	1
T0724	11,999	5886	3	1	0
T0743	12,000	6374	2	2	4
T0745	12,000	6108	2	2	0

Materials and Methods

Secondary structure and transmembrane span prediction

In the first step, the secondary structure is predicted for soluble proteins using Jufo9D [101-103], PsiPred [99, 100] and ProfPHD [97, 98]. For membrane proteins Jufo9D [101] and Octopus [104, 105] are used to detect secondary structure and transmembrane spans. From the predicted secondary structures, a pool is created for use by BCL::Fold as described before [50]. The pool is manually examined to ensure a complete as possible set of SSEs.

Fold recognition and domain identification

Fold recognition methods combined in bioinfo.pl were used to see if the target sequence contains multiple domains [113], and if proteins of those folds have been experimentally determined. If the fold recognition result indicated that the target has multiple domains, the SSE pool is split up into sub pool according to the domain boundaries.

BCL::Fold folding simulation

BCL::fold is run next to produce 12,000 models for each domain of one target. Depending on the target, the soluble or membrane protocol is employed. For each model, a completeness estimate is calculated as a fraction of the sum of the sequence lengths of all SSEs in the models to the total sequence length of the target. Models that are 2% less complete than the average model produced are removed.

Clustering to identify topologies that reside in wide energy funnels

After filtering the 12,000 models per target by completeness score, models were selected by three criteria for further refinement. The first method for selection was clustering by average RMSD linkage between models where the clusters ideally only contain models with the same fold. Cluster sizes varied with the largest clusters having a few hundred models and the smallest clusters contains a few or even a single model. Cluster radii leafs were between 0 and 18 Å with the most at 10 Å. The RMSD cutoff was manually adjusted based on protein size and model similarity. Up to five models from each cluster were selected for further refinement. The second method for selection was ranking by the BCL scoring function. All filtered models were sorted by BCL sum score and the lowest scoring models were selected. The third method was only used if we successfully identified a template model of the target protein and models were pooled into a separate set. In this case, the RMSD between the template and BCL generated models were computed. The models with the highest similarity (lowest RMSD) were selected for further refinement. Furthermore, in some cases the selected models were visually inspected in PyMOL to evaluate sequence length and Euclidean distances for later loop reconstruction. In this step, some models were removed from further processing if loops went through the center of the protein core.

Combining domains into complete models

If the target consisted of multiple domains, models of all possible combinations of domains are created either by arranging the domains in space close to each other or, if possible, by aligning the domain models to a template. The domains do not have to be connected by creating a loop at this point, because all models consist of only SSEs and loops will be built in the next step.

Loop construction using cyclic coordinate descent

Adding loops is a two-step process of inserting the missing amino acids in a model and creating coordinates for them by CCD. Once SSEs have been placed, loop regions between SSEs must be built. Creating loops is a two-step process of inserting the missing amino acids in a model and creating coordinates for them. This is accomplished by adding loop residues using (1) knowledge based potentials, (2) likely phi and psi backbone angle, and (3) cyclic coordinate descent (CCD). The first step is to dynamically add missing residues in the loop region. Residues are added with initial phi and psi angles derived from a probability distribution of experimentally observed angles. They are then perturbed and scored using a knowledge based potential for native like angles. This potential has scoring terms that penalize clashes between atoms using van der Waals radii, compare the sequence length with the Euclidean distance, measure the gap between adjacent SSEs, incorporate angles derived from Ramachandran plots, and score the likelihood that the distance between the SSEs can be closed by a loop. Once the initial residue coordinates of the loop region have been placed, CCD [114] is used to minimize the distance between a freely moving and fixed set of coordinates to close a loop. In this second step, an additional penalty term is added to the scoring function that scores how close the residue at the loop end is to the pseudo residue at the N terminus of the target SSE. Between 200 and 8400 loop models were built depending on model size and complexity to achieve a sufficiently low BCL sum score that is, in a similar score range than the non-loop start model. Models with loops difficult to close were either modified to allow an easier lop closure by shortening the SSEs adjacent to the loop or they were removed from further modeling. The best scoring loop models according to the BCL sum score were further processed.

Addition of side chains and model relaxation

One of two methods was used: either side chains were added with a relax step in which the relative position of the amino aces were restrained, or if the first method fails because of misaligned β-strands, by adding and repacking side chains. Between 10 and 200 side-chain models were built to obtain an optimal overall Rosetta score.

Model selection for submission

From the lowest scoring side chain models for each loop models, the ones deemed most native-like by visual inspection were selected for CASP10 submission. If a template model and a similar BCL model were found before, it was selected as the fifth submitted model.

Topology score to evaluate protein models

To evaluate if BCL::Fold can sample the folding space required for our target proteins, we introduce a new measure that focuses on SSE contacts instead of comparing atom positions like RMSD100 [115] or GDT [112]. This new measure computes the similarity of a model to a native protein by calculating the fraction of SSE contacts of the native that are present in a given model and the total number of SSE contacts of the native (true positive rate, sensitivity). An SSE contact is assumed if the distance of the central axis of two SSEs is less than a certain threshold. An SSE can be represented by its central axis for the purpose of the distance calculation, because all SSEs in a BCL model are idealized. The threshold below which two SSEs are assumed in contact depends on the type of SSE contact (helix-helix: 16 Å; helix-sheet: 16 Å; strand-strand: 5.5 Å; sheet-sheet 14 Å) and was derived from native protein structure from the PDB. These thresholds were chosen to be large to be as inclusive as possible. The strength of the interaction is represented by line thickness of the connecting lines.

Results

Eighteen targets included in this analysis

During CASP10 a total of 53 targets were released for human predictors. Eighteen of these had a least one domain in the FM category. To focus our efforts we excluded proteins that were very small (<50 residues) or very large (> 400 residues). Further, for some targets calculations did not finish in time for submission. For 21 targets models were submitted, five of them in the FM category. For two targets files were corrupted on our server, for one target no experimental structure has been released. This leaves 18 targets, three in the FM category, for analyses. Accordingly, treatment of the TBM targets as FM targets substantially increased the number of proteins that could be included in the study beyond the small number of FM targets. One consequence of this procedure is that BCL::fold will not rank among top methods from the TBM section, as we do not expect BCL::fold to predict protein structure more accurately than comparative modeling.

An automated pipeline with minimal human intervention was setup

Here we give an overview of the overall protocol. A detailed description of the individual steps is given in the methods section. The folding pipeline starts with the downloaded target sequence from CASP10 Prediction center. In the first step, secondary structure and transmembrane spanning regions are predicted and stored in a "pool" using the consensus SSE prediction results. The SSE pool is manually examined to ensure that weakly predicted SSEs are available. Domain boundaries were identified with bioinfo.pl – a consensus fold recognition Meta server. [113] At this stage of folding, templates were identified from TBM targets and comparative models were constructed using the Modeler[53, 94-96] link of the bioinfo.pl server. The homology model was saved for later analysis or prioritization of the *de* *novo* folded models. It was not used to bias the folding simulation. If the fold recognition result from bioinfo.pl indicated that the target consisted of multiple domains, the SSE pool was split into subpools according to the domain boundaries. Next, each domain was folded 12,000 times with BCL::fold. Resulting models were filtered for completeness before entering the clustering protocol. The completeness estimate is the total number of residues in SSEs divided by the total number of residues in the protein model.

Table 7: Statistics on 18 CASP 10 Targets Predicted with BCL::Fold

Target	PDB ID	Length	NCO	Category	Oligomeric state	Domains	α-helices	TM α-helices	β-strands
T0644	4FR9	166	22.1	TBM-easy	Monomer	1	2	0	8
T0649	4F54	210	58.9	TBM-hard	Monomer	1	4	0	9
T0655	2LUZ	182	44.2	TBM-easy	Monomer	3	4	0	8
T0663	4EXR	205	28.4	FM	Monomer	2	2	0	8
T0666	3UX4	195	64.9	FM	Trimer	1	6	6	0
T0676	4E6F	204	45.2	TBM-hard	Dimer	1	4	0	7
T0678	4EPZ	161	30.5	TM-hard	Monomer	1	7	0	0
T0682	4JQ6	235	63.5	TMB-easy	Trimer	1	7	7	0
T0684	4GL6	270	36.9	FM	Dimer	2	8	0	8
T0686	4HQO	259	55.7	TBM-easy	Dimer	3	4	0	5
T0691	4GZV	163	25.7	TBM-easy	Monomer	3	0	0	8
T0700	4HFX	86	18.0	TMB-easy	Tetramer	2	3	0	0
T0704	4HG2	254	55.4	TMB-easy	Dimer	3	9	0	8
T0720	4LC1	202	47.5	TMB-easy	Monomer	1	7	0	6
T0722	4FLA	152	44.1	Cancelled	Tetramer	Cancelled	4	0	0
T0724	4FMR	265	42.6	TMB-easy	Tetramer	2	4,5	0	16
T0743	4HYZ	149	36.9	TMB-easy	Monomer	1	4	0	5
T0745	4FMW	185	49.4	Cancelled	Dimer	Cancelled	6	0	6

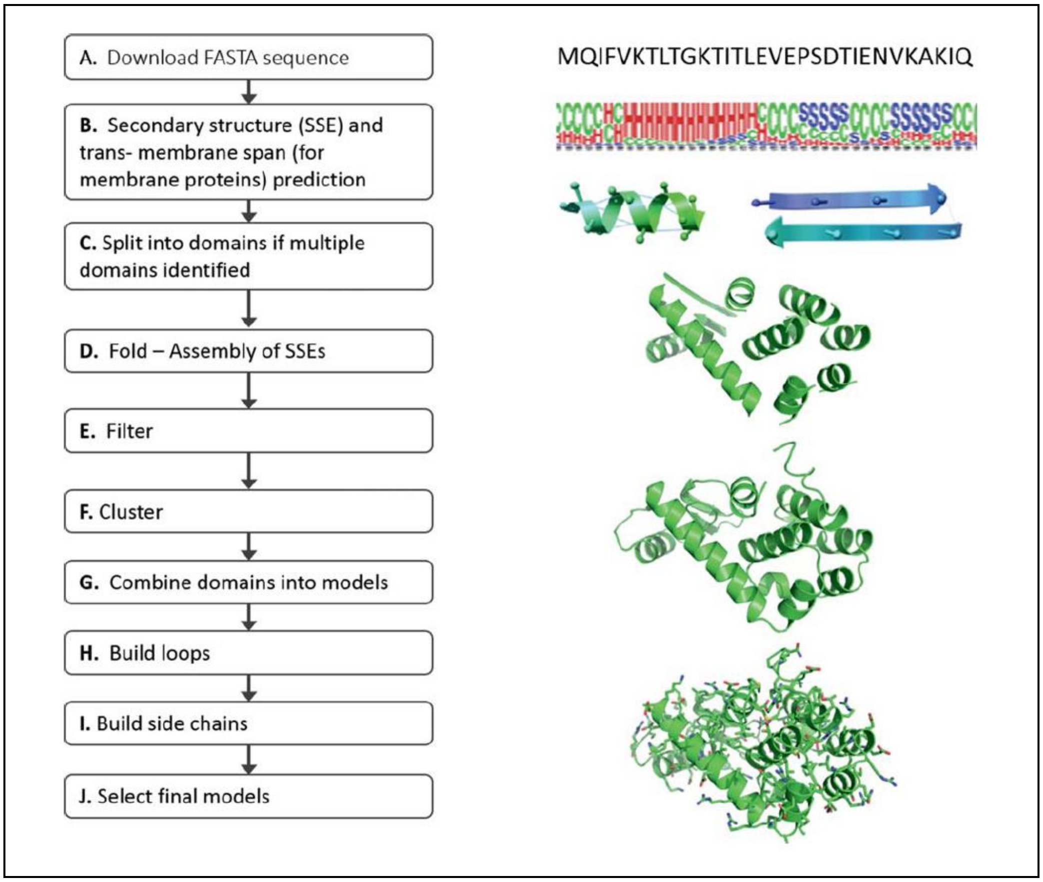
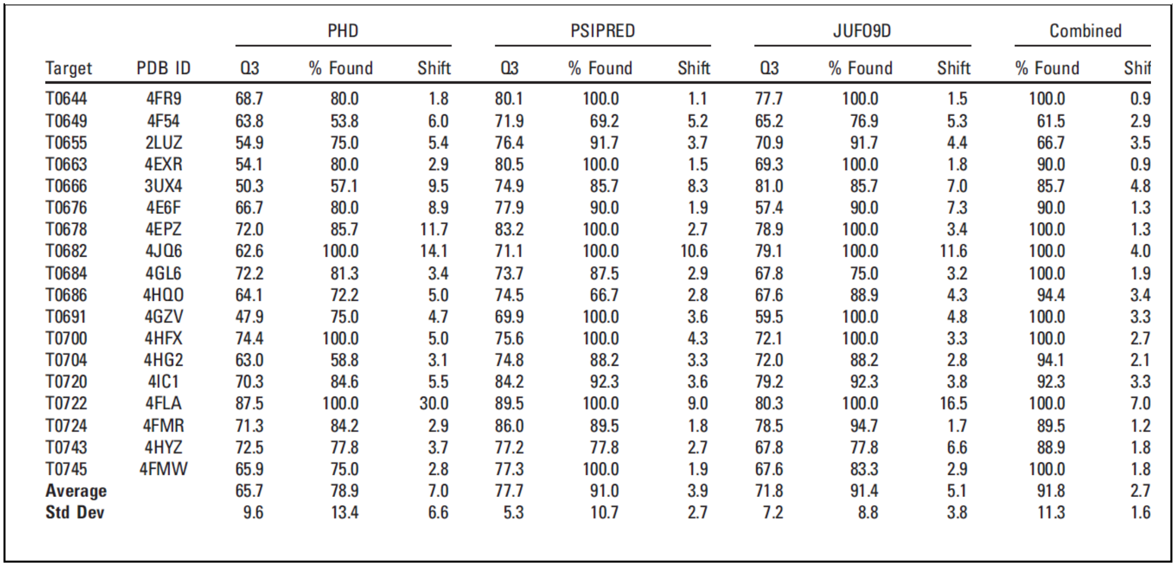

A. Download FASTA sequence
B. Secondary structure (SSE) and trans- membrane span (for membrane proteins) prediction
C. Split into domains if multiple domains identified
D. Fold – Assembly of SSEs
E. Filter
F. Cluster
G. Combine domains into models
H. Build loops
I. Build side chains
J. Select final models
MQIFVKTLTGKTITLEVEPSDTIENVKAKIQ

Figure 14 CASP10 Pipeline. Obtain target sequence from CASP10 prediction center (A); Perform SSE prediction (B); Split multimeric proteins into individual domains (C); Assemble SSEs in Folding algorithm, analyze fold models, compare generated models with native secondary structure, evaluate loop closure potential and bet sheet register shift (D); Filter erroneous models from further analysis €; Cluster predicted folds and analyze cluster centers (F); Combine domains if previously split (G); Reconstruct loop regions an analyze models (H); Build side chains with Rosetta(I); Select final models and analyze final model selection (J).

Table 8: Secondary Structure Pool Statistics for CASP10 Targets



		PHD			PSIPRED			JUFO9D			Combined	
Target	PDB ID	Q3	% Found	Shift	Q3	% Found	Shift	Q3	% Found	Shift	% Found	Shif
T0644	4FR9	68.7	80.0	1.8	80.1	100.0	1.1	77.7	100.0	1.5	100.0	0.9
T0649	4F54	63.8	53.8	6.0	71.9	69.2	5.2	65.2	76.9	5.3	61.5	2.9
T0655	2LUZ	54.9	75.0	5.4	76.4	91.7	3.7	70.9	91.7	4.4	66.7	3.5
T0663	4EXR	54.1	80.0	2.9	80.5	100.0	1.5	69.3	100.0	1.8	90.0	0.9
T0666	3UX4	50.3	57.1	9.5	74.9	85.7	8.3	81.0	85.7	7.0	85.7	4.8
T0676	4E6F	66.7	80.0	8.9	77.9	90.0	1.9	57.4	90.0	7.3	90.0	1.3
T0678	4EPZ	72.0	85.7	11.7	83.2	100.0	2.7	78.9	100.0	3.4	100.0	1.3
T0682	4JQ6	62.6	100.0	14.1	71.1	100.0	10.6	79.1	100.0	11.6	100.0	4.0
T0684	4GL6	72.2	81.3	3.4	73.7	87.5	2.9	67.8	75.0	3.2	100.0	1.9
T0686	4HQO	64.1	72.2	5.0	74.5	66.7	2.8	67.6	88.9	4.3	94.4	3.4
T0691	4GZV	47.9	75.0	4.7	69.9	100.0	3.6	59.5	100.0	4.8	100.0	3.3
T0700	4HFX	74.4	100.0	5.0	75.6	100.0	4.3	72.1	100.0	3.3	100.0	2.7
T0704	4HG2	63.0	58.8	3.1	74.8	88.2	3.3	72.0	88.2	2.8	94.1	2.1
T0720	4IC1	70.3	84.6	5.5	84.2	92.3	3.6	79.2	92.3	3.8	92.3	3.3
T0722	4FLA	87.5	100.0	30.0	89.5	100.0	9.0	80.3	100.0	16.5	100.0	7.0
T0724	4FMR	71.3	84.2	2.9	86.0	89.5	1.8	78.5	94.7	1.7	89.5	1.2
T0743	4HYZ	72.5	77.8	3.7	77.2	77.8	2.7	67.8	77.8	6.6	88.9	1.8
T0745	4FMW	65.9	75.0	2.8	77.3	100.0	1.9	67.6	83.3	2.9	100.0	1.8
Average		65.7	78.9	7.0	77.7	91.0	3.9	71.8	91.4	5.1	91.8	2.7
Std Dev		9.6	13.4	6.6	5.3	10.7	2.7	7.2	8.8	3.8	11.3	1.6

The filtering cutoff is the average of all the completeness estimates reduced by 0.01. After filtering, cluster centers of the 10 to 20 largest clusters were selected for further processing. In addition, we included the five best scoring models measured by the BCL sum score. If templates were identified, best-scoring models that were similar to the template by Mammoth z-score [78] were retained in a separate pool of models. If the garget was split into multiple domains, these were recombined at this stage. The backbone of the resulting models was completed using a Cyclic Coordinate Decent (CCD) [114] loop closure algorithm within the BCL. Afterwards, side chain coordinates were constructed and the model was relaxed using Rosetta. From the resulting set of up to 200 models, five were chosen for submission by Rosetta energy. If a template has been identified, the fifth model submitted was chosen from the second pool as the one most similar to the template, to assess BCL::Fold's sampling capability independent from scoring.

Accuracy of secondary structure and transmembrane span prediction

Table 8 depicts Q3 accuracies (a measure of the accuracy for prediction per residue secondary structure), the percentage of native secondary structures correctly predicted and the average shifts for the SSE pool of the 18 CASP10 protein targets. The shift values are the sum of the deviation in the first and last residues of the predicted SSEs when compared with native SSEs. The overall average percentage of native secondary structures correctly predicted (% found) using PHD, [97, 98] PSIPRED, [99, 100] and JUFO9D [101-103] was 91.8%. In the original benchmark of BCL, the overall average % found was 96.6%. [50] We achieve the highest overall accuracy by combining multiple secondary structure prediction methods to create the SSE pool, rather than relying on a single secondary structure prediction method. For example, the % found values for PHD, PSIPRED, and JUFO9D are 78.9, 91.0, and 91.4%, respectively. In the original BCL benchmark, these values for PSIPRED and JUFO are 96.1 and 90.3% respectively. This indicates that the secondary structure prediction is more challenging for the CASP10 targets than the original BCL benchmark. IN addition, during a folding run, BCL::fold can merge, grow, or shrink SSEs based on the predicted probabilities.

Quality of CASP10 FM models submitted by other research groups

There were 20 FM targets in CASP10. For all participating methods the average GDT_TS score ranged from 7.0 to 36.0% with a mean GDT_TS score of 21.7% and a standard deviation of 7.2%. The maximum GDT_TS score ranged from 16.5 to 44.0% with a mean GDT_TS score for 32.8% and a standard deviation of 8.2%. For the three targets attempted with BCL::Fold (T0663, T0666, and T0684) the average GDT_TS score submitted by CASP10 participants was 24.5% with a standard deviation of 10.2%. The mean of the maximum GDT_TS scores for these targets was 34% with a standard deviation of 9.5% (Figure 15)

Quality of BCL::fold models and sampling of the topology space

We assess the quality of BCL::Fold models in two ways. The GDT_TS score allows for comparison with other results; the topology score focuses its evaluation criteria specifically on SSE contacts which tests BCL::Fold's method of assembly.

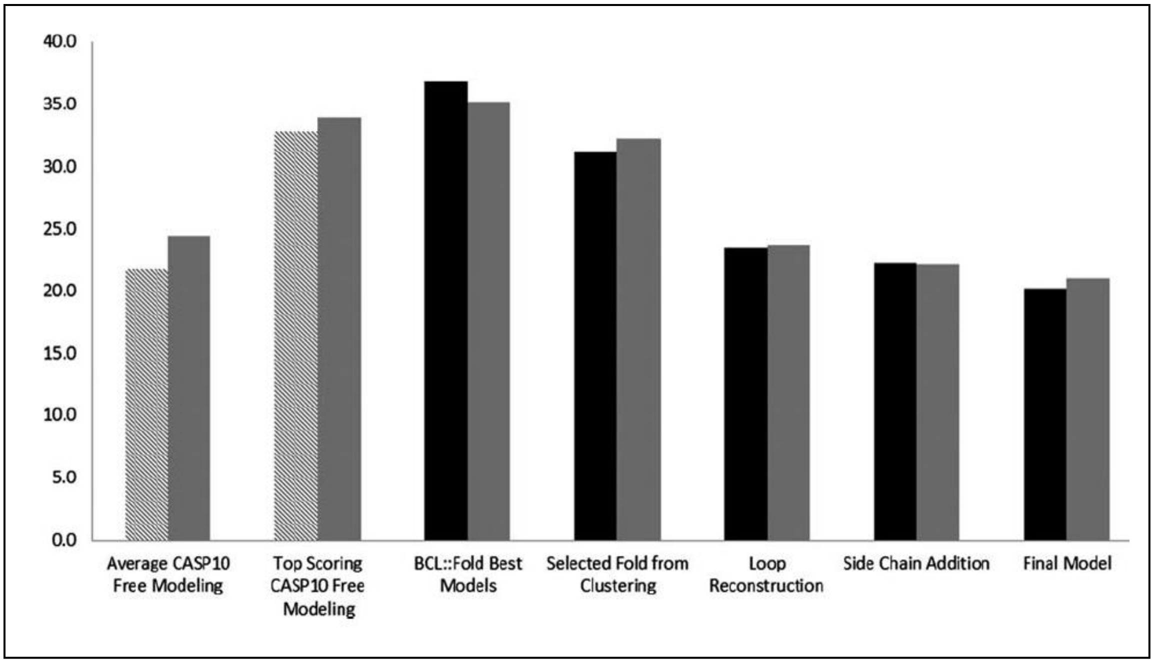


Figure 15: GDT_TS score analysis. Twenty FM targets from CASP10 (left two pars, pattern). Three targets folded also by BCL::fold from FM category in CASP10 (left two bars, gray). All 18 targets folded by BCL::Fold (black). Three FM targets folded by BCL::Fold (right five bars, gray). The y axis represents GDT_TS score.

GDT_TS scores for the best model generated by BCL::fold ranted in from 23.3 to 64.5% with a mean GDT_TS score of 36.8% and a standard deviation of 10.4%. Using the mean GDT_TS score of 33% as a comparative measure between other methods, BCL::Fold was able to sample models above this threshold in 12 out of 18 cases. Comparisons of the BCL models with the experimentally determined structure by measuring RMSD100 [116] and GDT_TS show efficient sampling of the correct topology (Table 9, Figure 15).

BCL::Fold's sampling performance was evaluated previously with soluble and membrane proteins. BCL::Fold was able to sample the correct topology in 61 of 66 soluble benchmark proteins [50] and in 32 of 38 membrane benchmark proteins. [55] The correct topology was defined as the ability to fold models with an RMSD100 of <8 Å to the native.

While RMSD100 is suitable to assess Rosetta models, it is not as helpful for BCL::Fold models that are focusing on sampling long-range contacts between SSEs. Figure 16 shows how well BCL::Fold samples the different protein topologies, measured the topology score. Its applicability is limited foremost by the number of SSE contacts. For targets with very few contacts (T0722 has a single contact) many models achieve a high score, and the discriminative value of the topology score is reduced. While the topology score does currently not consider specific types of interactions between SSEs, it does include the secondary structure type; thus, an incorrectly predicted secondary structure type leads to all contacts of this incorrect SSE to be evaluated as false. The thresholds to assume a contact between two SSEs are derived from idealized, native protein models and therefore fairly large; this can lead to detection of SSE contacts even for SSEs that are only indirectly in contact but still a very short Euclidean distance apart, like the first and third strand of a sheet. Additionally, the value of the topology visualization is narrowed by the projection of three dimensional protein structures into two dimensions, which reaches its limits for complex topologies. While the topology score has some caveats, overall it captures the protein topology quite well.

For the topology score, which measures the true positive contact ratio, we set the threshold to 0.8. At this level, two topologies share an overwhelming number of SSE contacts. Furthermore, we observe similarities when visually inspecting the topology plots of protein models (Fig 17).

BCL::Fold samples models above the threshold of 0.8 for 11 out of 18 targets (Fig 18). All targets with a native SSE contact count up to 20 have a topology score above the threshold. With increasing native SSE contact count and complexity, the topology score decreases expectedly.

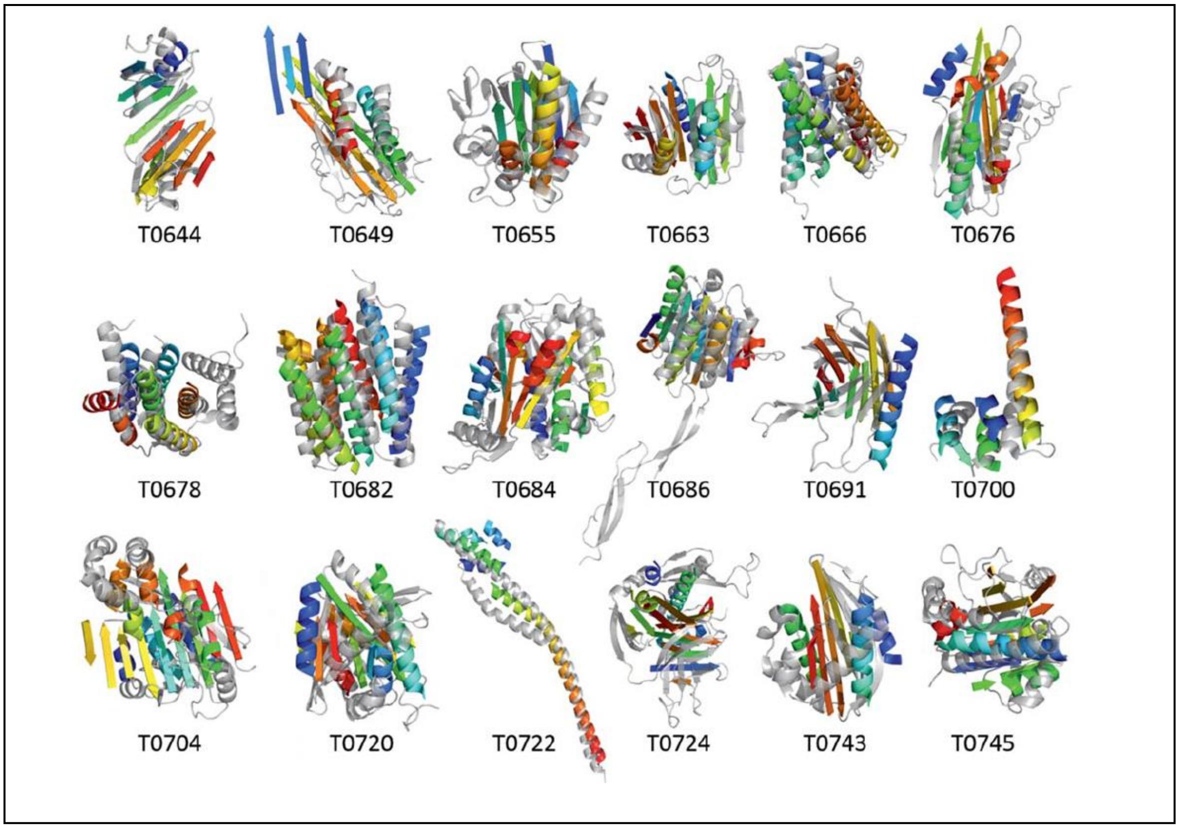


Figure 16: Highest GDT_TS model sampled with BCL::Fold (rainbow) overlaid with experimental protein structure (gray)

Selection of models for loop and side chain construction

Difficult, however, proved the selection of models for the subsequent refinement steps. During CASP10 we attempted selecting the best models by BCL sum score, the centers of the largest clusters, and the best scoring models in each cluster. However, no method enriched for high GDT_TS and consequently the models most similar to the native were consistently lost. For model T0700, we sampled a topology with an overall GDT_TS score of 64.5. We selected a model with a GDT_TS score of 57.6 for further refinement. After loop and side chain reconstruction, our model drifted further from the true native structure with a GDT_TS score of 38.6. Our final submitted model for this target had a GDT_TS score of 31.3. Most of the targets folded with BCL::Fold had this attrition pattern. Interestingly, model T0682 improved substantially after loop reconstruction from a GDT_TS score of 28.8 to 37.1. Our final submitted to CASP10 for this target had an RMSD100 score of 5.4 and GDT_TS Score of 33.0 (Fig 12 and 16).

Addition of loop and side chain coordinates

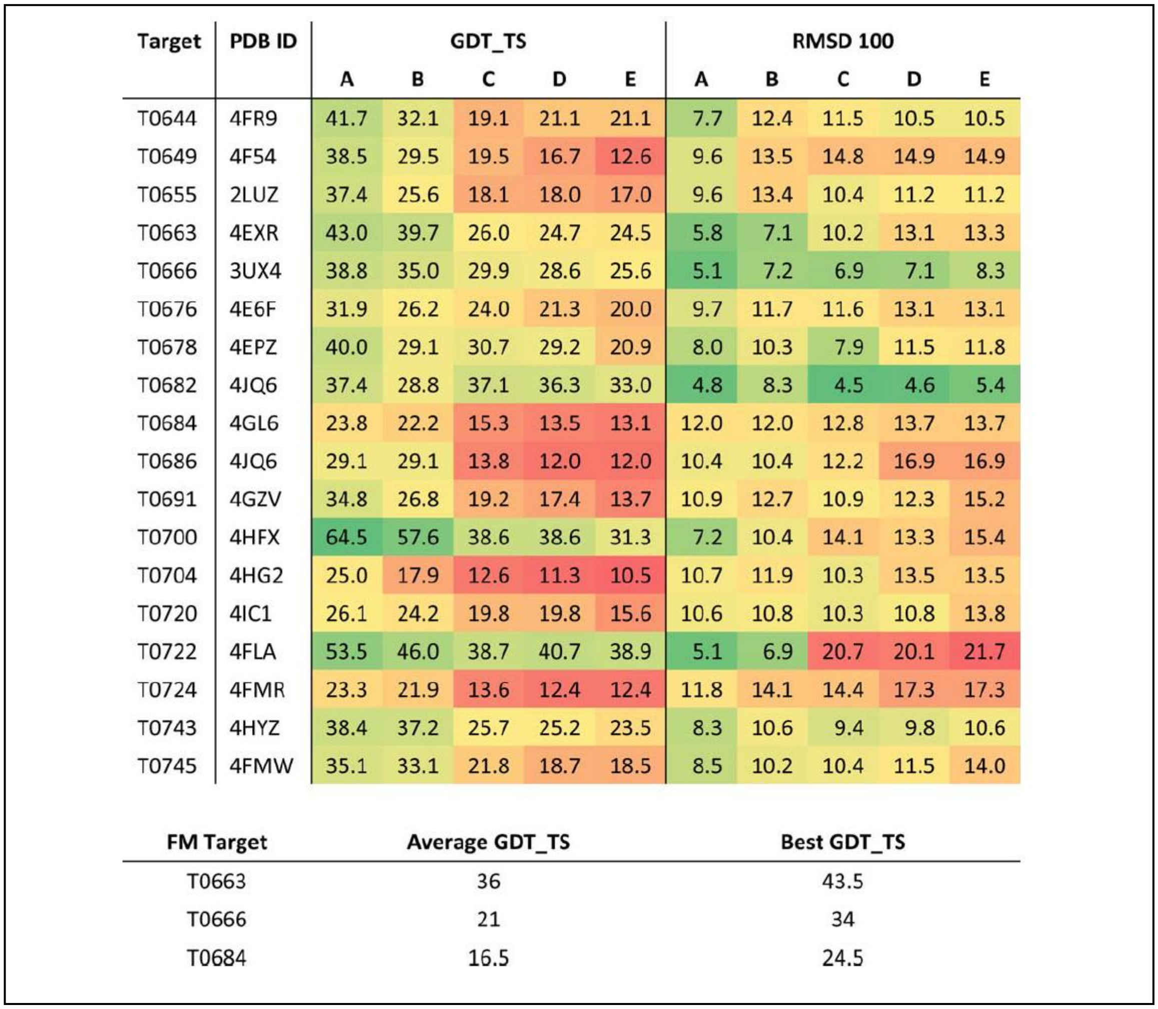
While adding loops to the cluster centers decreased the average GDT_TS scores from 31.2 to 23.5, the GDT_TS average dropped again from 23.5 to 22.4 when the side chains were added with Rosetta version 3.3. To rebuild side chains, the models were relaxed. To limit movement of the backbone constraints for every C_α-C_α bond distance below a cutoff of 8 Å were applied using a harmonic function with a standard deviation of 0.5. During side-chain reconstruction with Rosetta, 12 of the 18 CASP10 targets had a radius of gyration score >1100 for approximately 30% of all models indicating unfolding despite the constraint used (T0644, T-649, T0655, T0663, T0666, T0684, T0691, T0704, T0720, T0722, T0743, and T0745). This unfolding-like event was triggered because the BCL models scored poorly in the Rosetta energy function (Fig 20). Models that were unfolded were not considered further. As a method of last resort, Rosetta was used to add side chains without relaxing the backbone but only repacking the side chains.

Discussion

BCL::fold fails to sample the correct topology in seven cases

In 7 out of 18 cases, the best scoring BCL::fold model had a topology score of <0.8, which means the correct topology was not found. Investigation the reasons for these failures, we found that the target with the lowest topology scores had SSEs missing in the secondary structure prediction and subsequently in the SSE pool. T0655 had a topology score of 0.44 and had two helices missing; T0649 had a score of 0.68 and had one helix missing.

Table 9: Comparison of the GDT_TS Score and RMSD100 Score with the native. The Best Model Produced During folding with BCL::Fold (A); The Selected Models from Clustering (B); The Models After Loop Reconstruction (C); The Models After Side chain Addition (D); The Final submitted Model (E)



Target	PDB ID	GDT_TS					RMSD 100				
		A	B	C	D	E	A	B	C	D	E
T0644	4FR9	41.7	32.1	19.1	21.1	21.1	7.7	12.4	11.5	10.5	10.5
T0649	4F54	38.5	29.5	19.5	16.7	12.6	9.6	13.5	14.8	14.9	14.9
T0655	2LUZ	37.4	25.6	18.1	18.0	17.0	9.6	13.4	10.4	11.2	11.2
T0663	4EXR	43.0	39.7	26.0	24.7	24.5	5.8	7.1	10.2	13.1	13.3
T0666	3UX4	38.8	35.0	29.9	28.6	25.6	5.1	7.2	6.9	7.1	8.3
T0676	4E6F	31.9	26.2	24.0	21.3	20.0	9.7	11.7	11.6	13.1	13.1
T0678	4EPZ	40.0	29.1	30.7	29.2	20.9	8.0	10.3	7.9	11.5	11.8
T0682	4JQ6	37.4	28.8	37.1	36.3	33.0	4.8	8.3	4.5	4.6	5.4
T0684	4GL6	23.8	22.2	15.3	13.5	13.1	12.0	12.0	12.8	13.7	13.7
T0686	4JQ6	29.1	29.1	13.8	12.0	12.0	10.4	10.4	12.2	16.9	16.9
T0691	4GZV	34.8	26.8	19.2	17.4	13.7	10.9	12.7	10.9	12.3	15.2
T0700	4HFX	64.5	57.6	38.6	38.6	31.3	7.2	10.4	14.1	13.3	15.4
T0704	4HG2	25.0	17.9	12.6	11.3	10.5	10.7	11.9	10.3	13.5	13.5
T0720	4IC1	26.1	24.2	19.8	19.8	15.6	10.6	10.8	10.3	10.8	13.8
T0722	4FLA	53.5	46.0	38.7	40.7	38.9	5.1	6.9	20.7	20.1	21.7
T0724	4FMR	23.3	21.9	13.6	12.4	12.4	11.8	14.1	14.4	17.3	17.3
T0743	4HYZ	38.4	37.2	25.7	25.2	23.5	8.3	10.6	9.4	9.8	10.6
T0745	4FMW	35.1	33.1	21.8	18.7	18.5	8.5	10.2	10.4	11.5	14.0

FM Target	Average GDT_TS	Best GDT_TS
T0663	36	43.5
T0666	21	34
T0684	16.5	24.5

Models for T0724 have an incorrect strand topology because BCL::Fold models were created as protomers while the native exists as dimer in which strands from both monomers for a sheet.

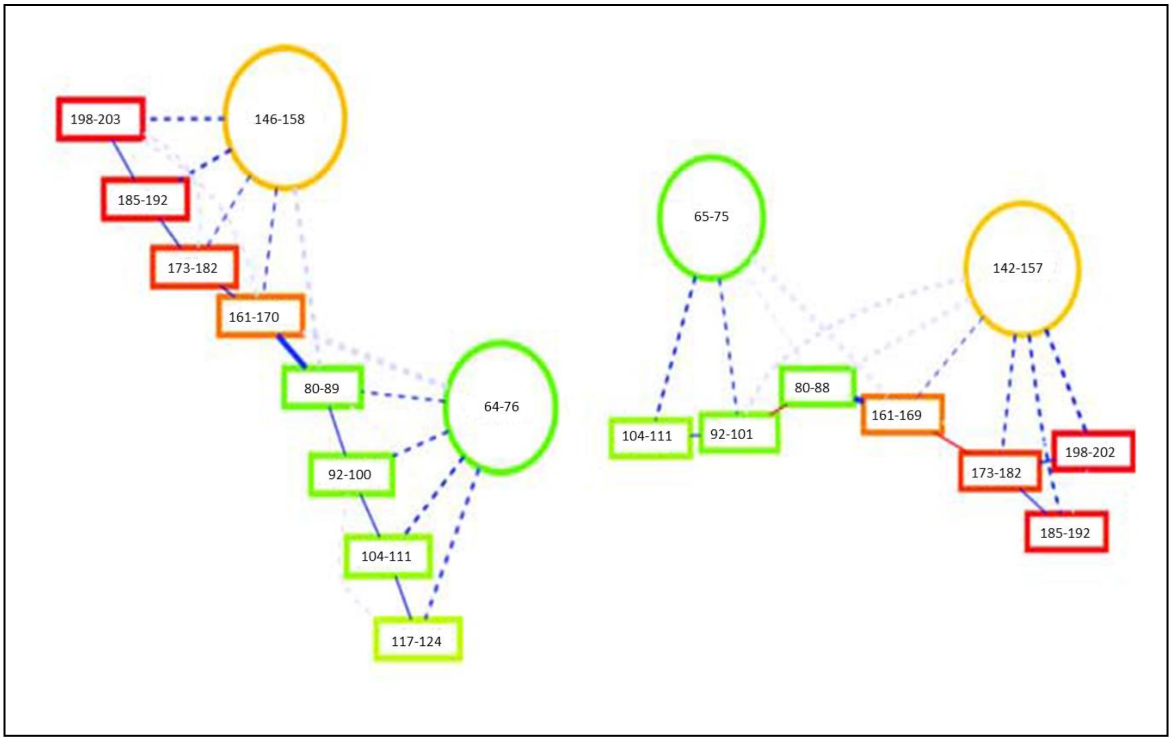


Figure 17: Visualization of the topology for the native and the best scoring model according to the topology score for T0663 with topology score of 0.81

The remainder of four incorrect targets failed to sample the correct topology because of a combination of reasons, most notable for two reasons. Long SSEs were split into two smaller ones, either by DSSP when assigning secondary structure to the natives, or by the secondary structure prediction methods that we employed. The correct topology was simply not sampled and recognized as a best scoring model, often with the order of strand SSEs in sheets being incorrect.

BCL::Fold models have loops that are impossible to close

BCL::Fold assembles tertiary structure from disconnected SSEs. Because of this, we must ensure that the distance between the end of one SSE and the beginning of the next SSE can be bridged by a loop. Two components of the BCL::fold scoring function control this requirement: First, there is a penalty if the Euclidean distance between two SSEs is longer than the maximal Euclidean distance that can be bridged by the number of amino acids in the loop. Models that violate this rule are heavily penalized during Monte Carlo sampling and likely rejected. The second component is deigned to place SSEs so that loops between them match a loop score potential that reflects native loop conformations from the PDB (PISCES dataset, see Methods). This loop score potential evaluates the Euclidean distance probability in dependence of number of residues. [51] As this score is a function of only Euclidean distance and sequence distance, it neglects the spatial arrangement of SSEs. Analysis of CASP10 models revealed that BCL::Fold constructs models where loops cannot be closed without passing through SSEs. Figure 18 depicts a model produced by BCL::fold for target T0663. The Euclidean distance between residues ASN55 of helix 1 and TYR65 of helix 2 is 25.5 Å. To bridge this distance with 9 amino acids, each amino acid has to be 2.8 Å on average, which is less than the average C_{α}-C_{α} distance of 3.3 Å. However, with the placement of strand SSEs between the loop ends, all paths to close the loop between helices 1 and 2 pass through the strand SSEs. Overall, 76% of BCL::Fold models produced during CASP10 folding simulations contains non-closable loops because of this behavior.

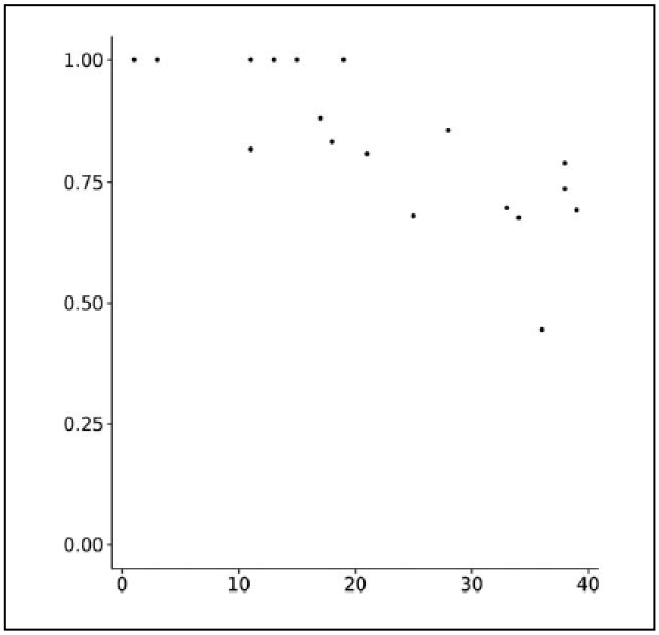


Figure 18: Topology Score Threshold, True positive rate (precision, y axis) compared with the complexity of a protein (number of SSE contacts in the native, x axis). The true positive rate of BCL::fold models decreases with increasing complexity.

The BCL::Fold loop potential is often violated for consecutive SSEs

Loops found in native proteins bridge preferable Euclidean distances d_e depending on the loop's sequence length d_s. The current loop potential of BCL::fold mirrors this preference. It is a sequence independent score, which contributes to the overall energy function. The PISCES data set used to create this potential includes all possible loops, that is, loops between consecutive and nonconsecutive SSEs. Because BCL::fold does not assemble SSEs in sequence order, the potential must evaluate incomplete protein models with unplaced SSEs. Therefore, nonconsecutive SSEs were included in the loop scoring potential.

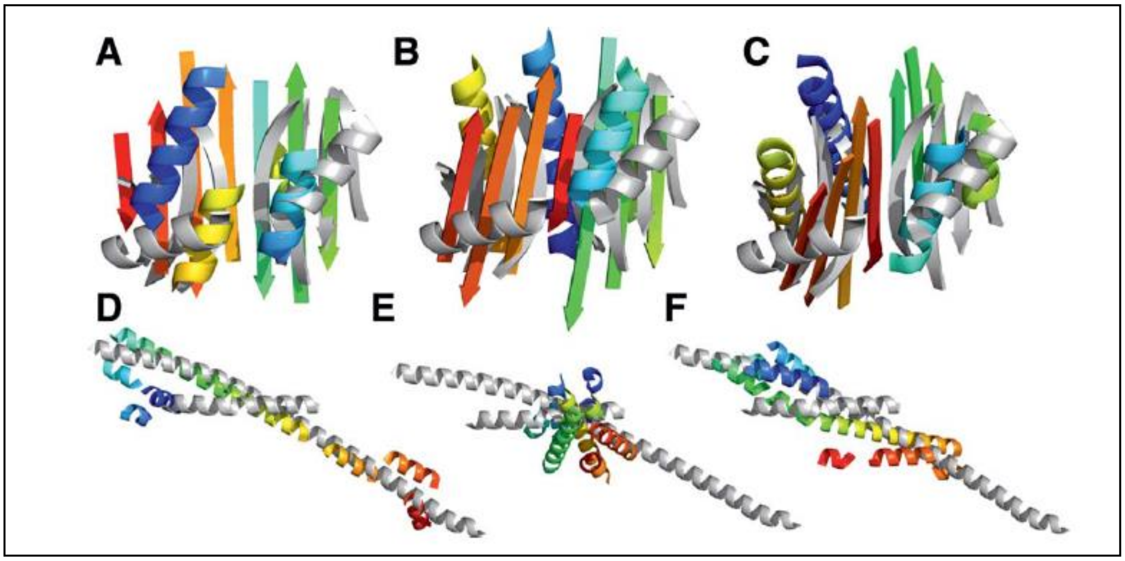


Figure 19: Comparison of example BCL models with the native target structure for T0663 (top) and T0722 (bottom). The experimental structures without loops are shown in gray (based on PDBIDs 4EXR and 4FLA, respectively). The predicted models (rainbow) show the highest scoring model produced by BCL (A, D, with a GDT_TS of 43.0 and 53.5, respectively); The best scoring model by BCL energy function (B,E with a GDT_TS of 28.9 and 26.9); The best scoring model in largest cluster (C, F, with a GDT_TS of 22.1 and 32.6)

To test the loop potential accuracy, we compare the CASP10 models produced by BCL::Fold to structures from the PISCES pdb set. Because the Euclidean distance that a loop spans depends on the sequence length of the loop, we normalize the Euclidean distance by the logarithm of the sequence length, $d_e/\log d_s$; this results in homogeneous distributions independent of loop length. The all-loop distributions (that is, consecutive and nonconsecutive loops_ for $d_e/\log d_s$ for CASP10 models, CASP10 natives, and PISCES are alike (Fig 22(A)). The means of the distributions are 6.2, 6.6, and 6.5 Å, respectively, and confirm their similarity. Thus, we conclude that this weighted potential distinguishes native-like sequence and distance length of loops from non-native configurations in terms of sequence length and corresponding Euclidean distance.

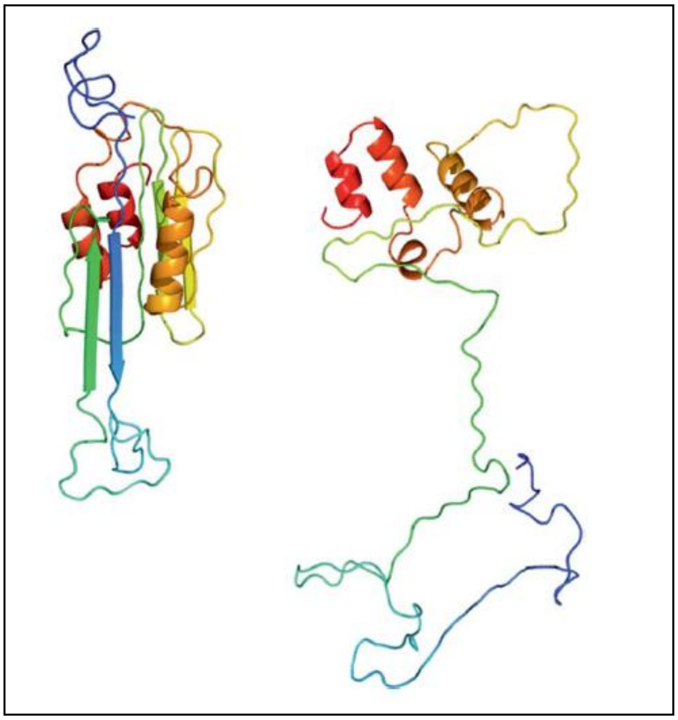


Figure 20: BCL model for target T0655 before (A) and after side chain addition and relaxation with Rosetta (B)

However, when evaluating the CASP10 models with the consecutive-only loop distribution (that is, only loops between consecutive SSEs are included), we find a substantial bias between CASP10 models and both CASP10 natives and PISCES structures (Fig. 12(B)). Their means are 8.1, 5.8, and 5.7 Å, respectively. The sequence length d_s of a loop is not changing as it is defined by the secondary structure (prediction) of the particular protein and only used for normalization. Therefore, the difference between the distributions can only be caused by differences in the Euclidean distances d_e. Creating models with loops of longer Euclidean distances d_e than found in native structure for a given sequence length causes BCL::Fold to produce non-native like loop arrangements. Thus, the loop potential is not a sufficient metric to generate native-like models from disconnected SSEs. Furthermore, the current loop potential does not consider the spatial positioning of other SSEs and does not account for potential clashes between these SSEs and a loop (Fig. 21).

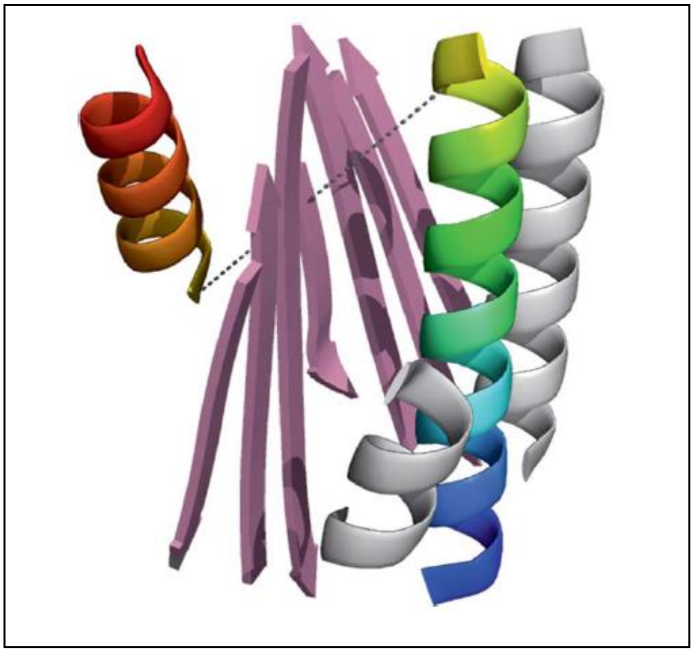


Figure 21 A model for CASP10 target T0663 folded by BCL. The Euclidean distance between residues ASN55 in helix 1 (rainbow colored on the right) and TYR65 in helix 2 (rainbow colored on the left) is 25.5 Å. Without the central sheet (pink) the loop could be closed; it is impossible to close the loop if the connecting amino acids have to be positioned around the sheet.

A small loop angle favors more native-like loops

To address the shortcoming we devised a loop measure that reflects this difference between consecutive and non-consecutive SSEs more drastically. For native proteins, we observe that loops between consecutive SSEs are positioned locally on a protein structure, that is, consecutive loops tend to begin and end on the same side of the structure and do not connect through the center. Geometrically this can be measured as the angle between the end of one helix, the center of the protein, and the start of the next helix (Fig 23(A)). In native protein structures, consecutive loops overwhelmingly favor small angles, as shown for the CASP10 native and PISCES pdb sets, of which 75% are smaller than 40° (Fig 23(B)) green and blue, respectively). Models with loops that would clash with other parts of the protein frequently have large angles of close to 180° (Fig. 13(B) red). We can use this information to discriminate native like arrangements from models with large angles.

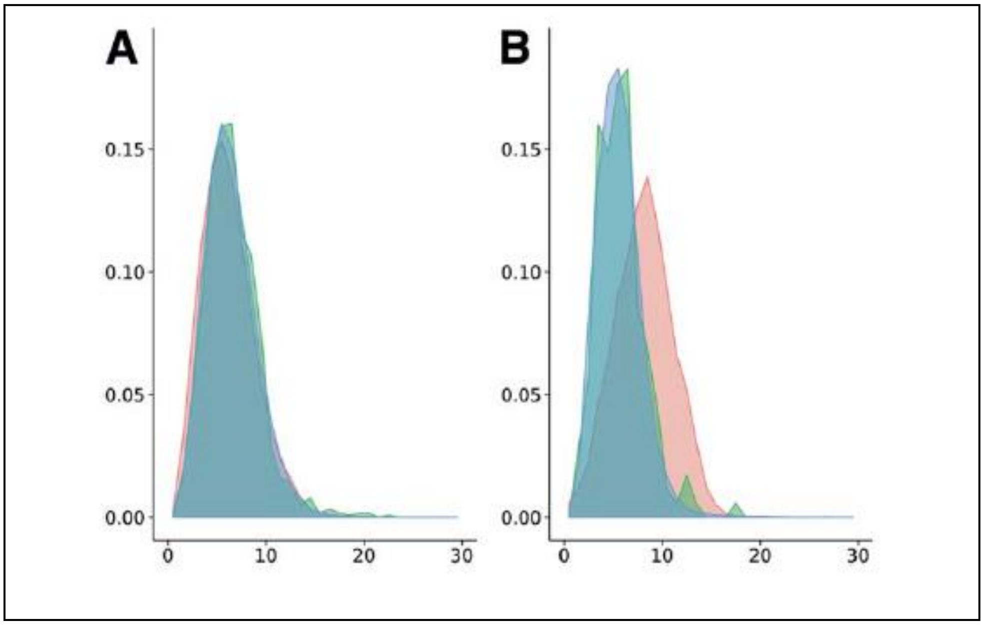


Figure 22 The density distribution of the BCL loop score displaying Euclidean distance over the logarithm of the sequence separation for loop regions between all SSEs (A) and consecutive SSEs only (B). While the distributions of BCL models (red), CASP10 natives (green) and PISCES dataset (blue) match each other for lops between all SSEs (A), the distribution of BCL models shows a shift when only loops between consecutive SSEs are considered (B)

When including nonconsecutive loops, the distribution of loop anges is exhibiting two frequently occurring angles, small ones for loops connecting consecutive SSEs, and large ones for connecting nonconsecutive SSEs (Figures 20(c)). To evaluate the loop angles of a protein model, we must differentiate between loops that connect consecutive and nonconsecutive SSEs.

To test whether filtering by the new loop angle measure would select for lower RMSD models compared to the existing loop score, we folded models for eight CASP10 targets (1000 models for T0655, T0663, T0676, T0678, T0684, T0700, T0745; 700 models for T0722). The RMSD cutoff was set to 10th percentile. Both, the existing loop score and the loop angle score were then used to select the best 50% of the models below the RMSD cutoff and in three cases decreased the number of models below the RMSD cutoff by more than the expected 50% (T0684, T0700, and T0722). The loop angle score filtered on average 61% of the models below the RMSD cutoff and only in one case, T0722, it selected less than 50% of the models below the RMSD cutoff. Thus, the loop angle score is selecting more native-like models and can improve the BCL scoring function moving forward. (Table 10)

Table 10: The percentage of models below the RMSD cutoff kept when filtering models for each target with the existing loop score and the loop angle score, showing that the loop angle score keeps in all cases more low RMSD models.

Target	% Models kept by existing loop score	% Models kept by loop angle score
T0655	70	70
T0663	67	76
T0676	52	57
T0678	52	63
T0684	44	57
T0700	37	57
T0722	16	43
T0745	59	62
Average	50	61

BCL::Fold misaligns β-strand registers

Carbonyl and amide groups in parallel and antiparallel strands of native proteins are aligned to allow the formation of stabilizing hydrogen bonds. A hydrogen bond is formed between the carbonyl-oxygen (hydrogen-bond acceptor) of one amino acid with the amide hydrogen of another amino acid (donor). In a sheet with the antiparallel strands i and j, the following pairs of atoms form hydrogen-bonds, here denoted as (acceptor, donor): (C_i, C_j), (C_j, C_i), (C_{i+2}, C_{j-2}), (C_{j-2}, C_{i+2}), (C_{i+4}, C_{j-4}), (C_{j-4}, C_{i+4}), … (Figure 24(A)); the pattern for parallel strands i and j is (C_i, C_{j+1}), (C_{j+1}, C_{i+2}), (C_{i+2}, C_{j+3}), … (Figure 24(C)).

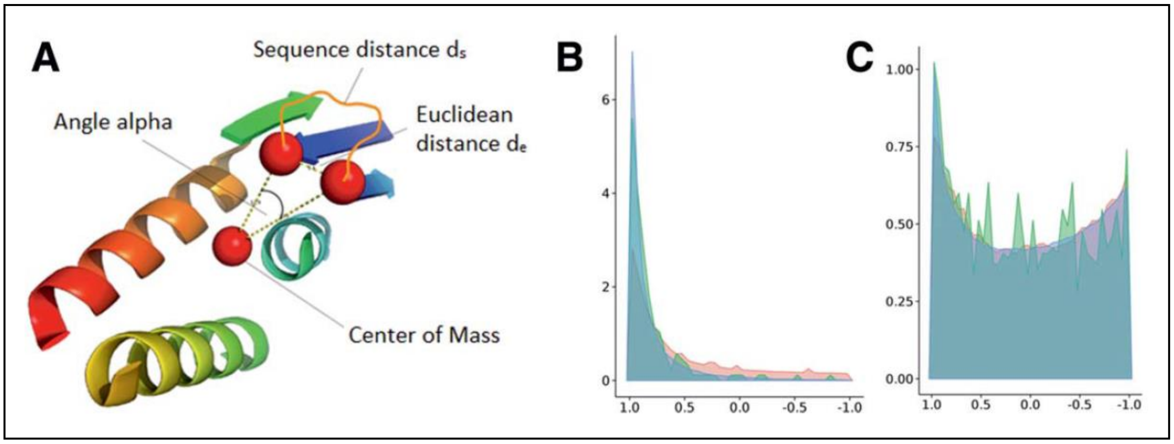


Figure 23: Visualization of loop angle metric, which measures the angle α between the end of one SSE (dark blue), the center of gravity, and the beginning of the next SSE (light blue; A). The density distribution of the cos(α) metric for lop regions between consecutive SSEs only is concentrated to acute angles for PISCES and CASP10 natives (B, blue and red, respectively). BCL models exhibit a higher number of large angles for consecutive loops (B, red). The density distribution of the cos(α) metric for lop regions between all possible SSEs shows two frequently found angles, small ones and large ones, for all sets, BCL models (red), CASP10 natives (green) and PISCES(blue; C).

BCL::Fold does not control for this alignment in order to simplify the folding energy landscape. It only controls for distance and relative orientation of β-strands within β-sheets. We hypothesized that misalignment of hydrogen bonds within β-sheets might cause clashes that are responsible for the large fraction of models that unfolds during Rosetta refinement.

To evaluate the strand register alignment of BCL models and compare them to natives, we measured the angle between carbonyl-carbon, the carbonyl-oxygen and the amide-hydrogen, and the distance from the carbonyl-oxygen to the amide-hydrogen. While in native proteins a hydrogen bond rarely has a Euclidean distance longer than 2.1 Å, we measured putative hydrogen bond atom pairs that were in paired β-strand SSEs and within a relaxed cutoff of 4.5 Å. The hydrogen-bonds in aligned strands of elucidated proteins have characteristic angles close to 180° and distances of 1.9 to 2 Å. Analysis of CASP10 BCL::Fold models, CASP10 experimental structure and the PISCES is summarized in Figure 24. In BCL models, we find substantial deviations to smaller angles and larger distances up to 4 Å for more than half of the models for both antiparallel and parallel sheets. The deviation in hydrogen bond angle and distance is correlated in BCL models. Additionally, BCL models exhibit a slightly shorter hydrogen bond distance of 1.8 to 1.9 Å even for hydrogen bonds with a native-like angle. This points to an incorrect placement of SSEs.

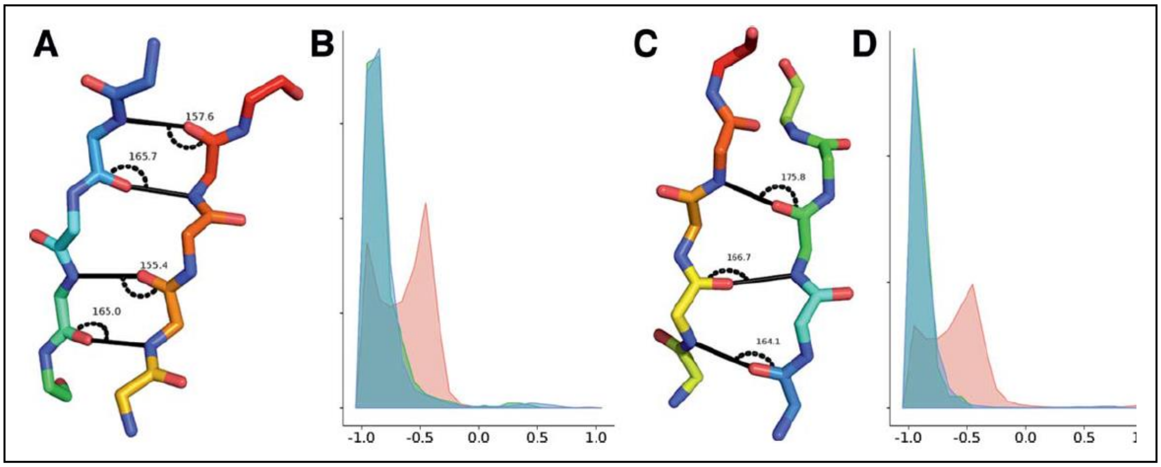


Figure 24: Hydrogen-bond pattern and angles between the carbonyl-carbon, carbonyl-oxygen, and amide-hydrogen in antiparallel (A) and parallel strands (C). Comparison of the hydrogen-bond angle for BCL models (red), CASP10 natives (green), and PISCES (blue) for antiparallel (B) and parallel strands (D). While the angles for CASP10 native and PISCES sets match, BCL models deviate. The x-axis shows the cosine of the hydrogen-bond angle, the y axis the normalized density.

Misaligned β-Strands cause clashes in Rosetta

The misaligned β-strands result in a high positive contribution from the repulsive score term (fa-rep) and no attractive contribution form the hydrogen bond score term (hbond_lr_bb), which leads to an unfavorable Rosetta score overall. The ra_rep term is the repulsive component of the van der Waals force, for example originating from carbonyl-oxygen of two strands being positioned too close to each other. The hbond-lr_bb term evaluates backbone-backbone hydrogen bonds distant in the primary sequence as they appear in sheets. Due to the misalignment of strands, the hbond_lr_bb term is zero and does not contribute to the overall Rosetta score. (Figure 25)

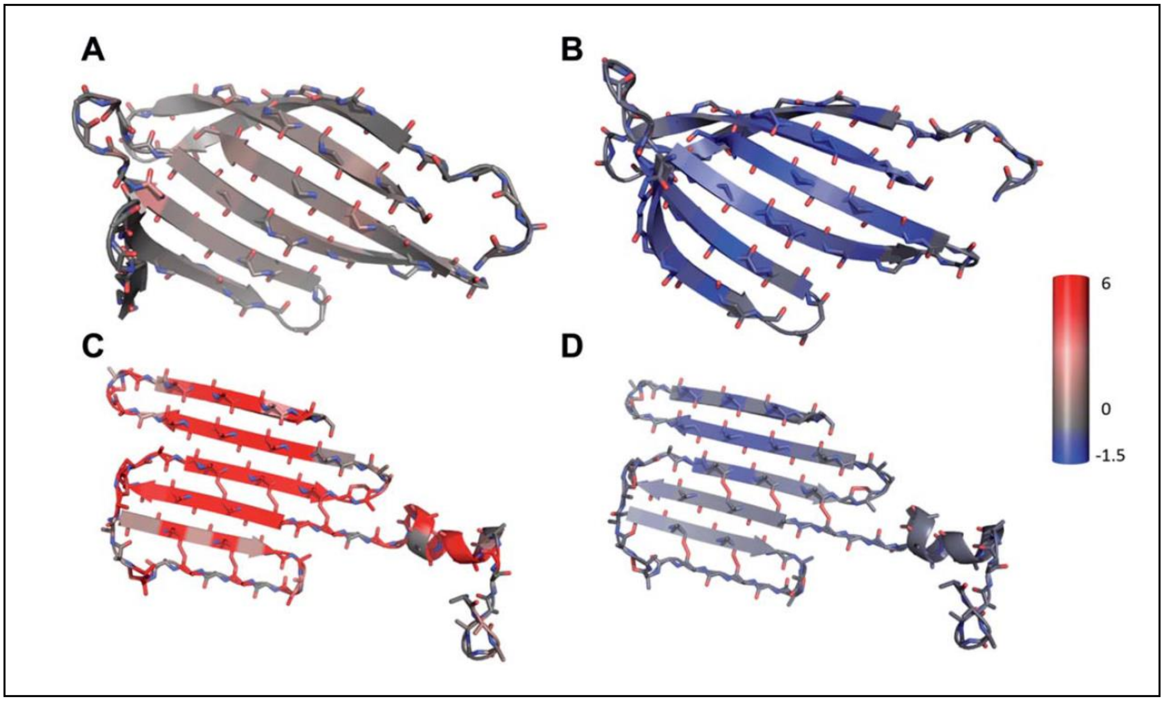


Figure 25: The analysis of Rosetta energy scoring terms for the native and a BCL model of target T0655 (shown is only the sheet part of native and model). The native shows no penalty from the repulsive score (A, fa_rep Rosetta score term) and a beneficial contribution from the hydrogen bonding score term (B, hbond_lr_bb Rosetta score term). Contrary, the BCL model exhibits a very high repulsive score (C, fa_rep) and little benefit from the hydrogen bonding term (D, hbond_lr_bb). The color scale stretches from blue representing -1.5 Rosetta energy units (REU) through gray (0 REU) to red (6 REU); the scale was chosen to red depict a value further from zero than blue to account for the bigger range of the repulsive score.

This causes Rosetta to unfold BCL models, despite constraints (Figure 20), in the last step of our CASP10 pipeline, which adds side chains and structurally refines the protein by cycling through repack and minimization steps.

B-Strand placement in BCL::fold models needs to be refined to align hydrogen bond donors and acceptors

The assembly of disconnected SSEs allows BCL::Fold to sample different sheet topologies and register positions without being restricted by the residues connecting the two strands SSEs. For this reason β-strand placement is controlled only be a mutate function that places one strand next to another in the preferred angle and distance. [51] However, the placement of β-strands only be the distance and torsion angle within the β-sheet is insufficient to produce BCL::fold models that can be refined with other programs. We plan to add a refinement stage into BCL::Fold that translates β-strands along their z-axis and evaluates a scoring term that controls the angle α introduced above. This will result in an improved scoring function that selects for more native-like models. We expect that improved alignment of β-strands will reduce the unfolding events observed during Rosetta refinement.

Conclusion

Despite inaccuracies in secondary structure prediction, BCL::Fold was able to sample the correct fold for most of 18 cases studies herein. The best methods in CASP10 submitted models with an average GDT_TS of around 33% in the RM category. BCL::Fold achieves this threshold in initial models after folding for 12 of 18 targets. Similarly, BCL::Fold is able to produce models with a topology score of at least 0.8 for 11 of 18 targets. However, the post folding filtering and refinement strategies removed correctly folded models from consideration in almost all cases, mostly for structural artifacts present in the BCL::Fold models. This result shows that BCL::Fold has the potential to compete with the best *de novo* structure prediction algorithms if a) unrealistic geometries in loops and β-strands can be removed and thereby the attrition of accurate topologies during model refinement can be stooped and b) an approach can be found that recognizes the most accurate models within the BCL::fold ensemble. However, with this analysis and planned work to address the recognized weaknesses, future versions of BCL::Fold produce more native-like models without incorporating templates or experimental data.

CHAPTER IV

RECONSTRUCTION OF EXPERIMENTAL SANS PROFILE FROM PROTEIN MODELS

Overview

A Small Angle Neutron scattering (SANS) measurement represents a molecule's rotationally average intensity (I) as a function of scattering angle (q). As with SAXS, large pairwise atomic distances are represented by small scattering angles and small pair wise atomic distances are represented by large scattering angles. (See Figure 26) In the case of SANS, the overall scattering curve represents the radially averaged contribution of all Neutrons including hydrogen atoms in all orientations. The same parameters from SAXS can be extracted directly from SANS. These parameters include the molecular mass (MM), radius of gyration (Rg), hydrated particle volume (Vp) and maximum particle diameter (Dmax). Furthermore, the SANS scattering curve contains information related to the overall shape of the molecule and is routinely used to validate structural models [1].

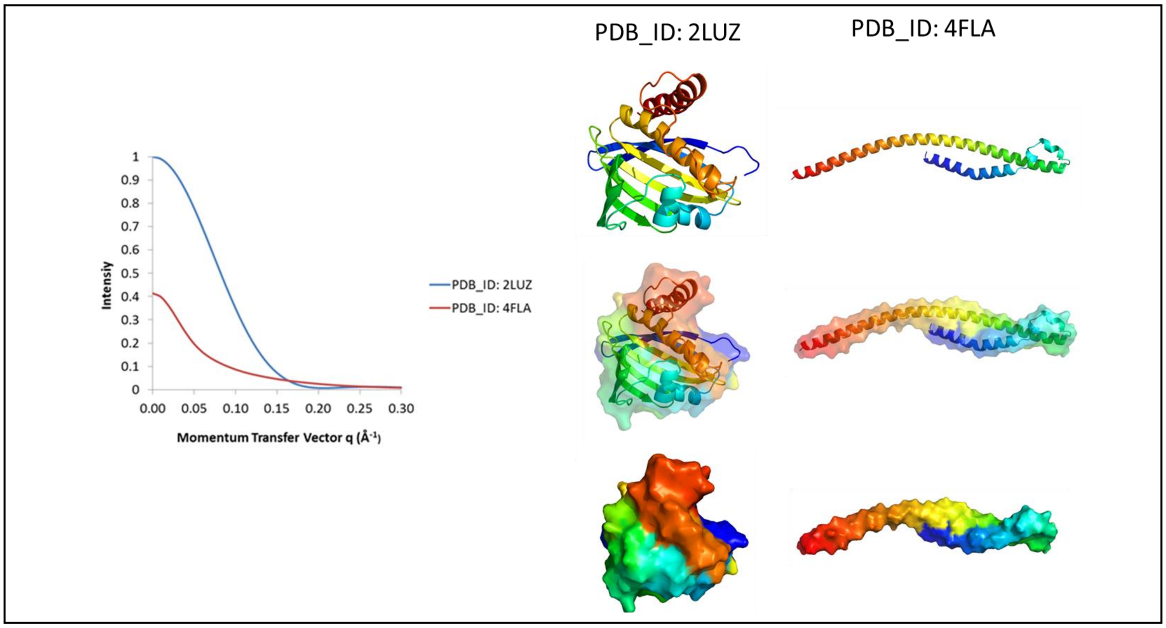


Figure 26: Envelopes of SAXS / SANS profiles. Two proteins of distinct shape: globular (2LUZ and rod (4FLA) are shown. On the left are the different scattering profiles, on the right are the proteins with the envelope surrounding the protein.

Significance

The Guinier analysis is a method to compute protein size, particle interactions (aggregation), oligomeric state, and overall data quality in reciprocal space. The radius of gyration and forward scattering I(0) are 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, then inter-particle interactions, polydispersity, or improper background subtraction has occurred in the sample. The I(0) value normalized to solute concentration is proportional to the MM. The MM can be used to distinguish different oligomeric states.

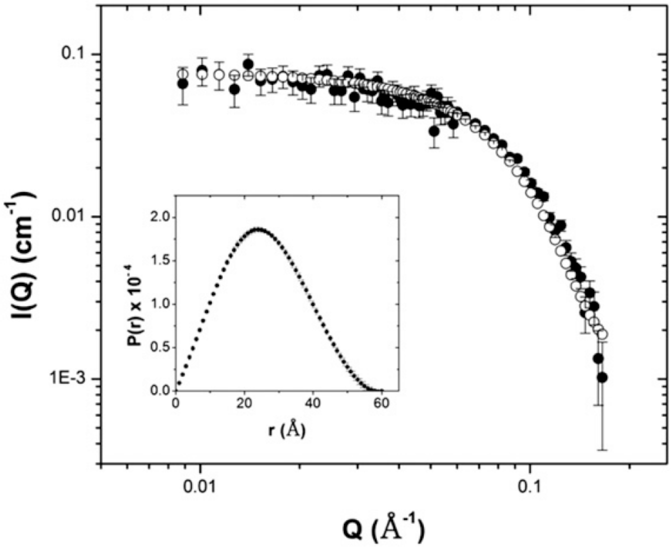


Figure 27: SANS analysis of AtCESA3CatD monomers. SANS profile for AtCESA1CatD (black circles) overlaid with simulated scattering curve for GhCESA1CatD (white circles) Inset shows P(r) plot derived from experimental scattering data of ATCESA1CatD. www.plantcell.org. Copyright American Society of Plant Biologists [1]

After Guinier analysis, the distance distribution function, p(r), is computed by a Fourier transformation of the SANS pattern. The shape of the p(r) function is used to characterize the overall shape of the system. A Direct Fourier transformation of the SAXS profile to obtain the distance distribution function is not possible and indirect Fourier methods must be used. A common method used is implemented in GNOM [117]. A spherical particle has a bell shaped p(r) curve and is shown in figure 27.

Innovation

This implementation of SANS uses the Debye equation with q-independent neutron scattering lengths used in the place of q-dependent from factors. These quantities are a measure of the scattering power a particular electron or neutron contain. The excluded volume is handled the same way it was for SAXS, with a parameter to account for varying neutron densities. The solution and hydration layer is controlled by a thickness parameter and a percentage of deuterium in solution. In the presence of deuterium, hydrogen exposed to the surface will be replaced with deuterium, which changes the neutron scattering profile.

Reconstruction of SANS Profiles from atomic models

Neutrons and protons form the nucleus of an atom and both have a mass of 1.67 x 10^{-27} kg. Protons have a positive net charge of +1 while neutrons have a net charge of 0. They have a spin state of +1/2. Electrons have a much smaller mass of 9.109 x 10^{-31} kg. Because the neutrons are uncharged, they are not affected by ionic interactions. They are only scattered by nuclear forces. This allows them to penetrate deeply into the target until they interact with other nuclei. Hydrogen scatters neutrons strongly. Although magnetic scattering occurs, I will focus on nuclear scattering.

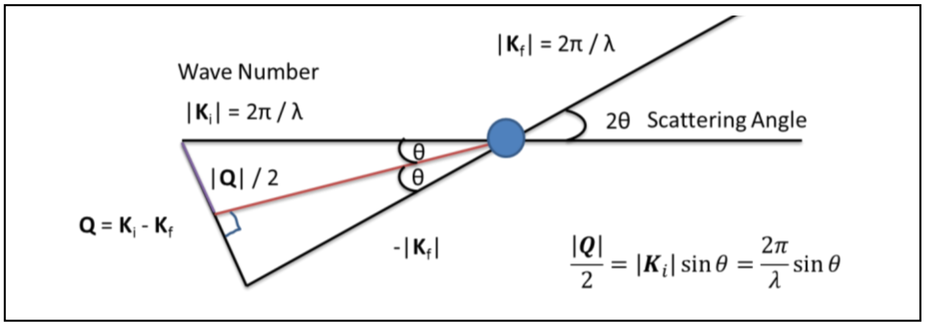
Neutron scattering is governed by the four dimensional scattering law. This law characterizes scattering as a function of the momentum transfer in three dimensions (**Q**) and energy (E). The scattering is recorded on a detector as a spot of intensity (I). For this implementation I restrict analysis to elastic scattering (no energy loss or gain): ΔE=0

$$I_{elastic}(\boldsymbol{Q}) = I(\boldsymbol{Q}, E = 0) \qquad 4.1$$

The scattering law dimensionality is reduced from four to three. The modulus of the wavevector and hence the wavelength (λ) is unchanged.

$$|K_i| = |K_f| = \frac{2\pi}{\lambda} \qquad 4.2$$

Because the energy change is zero during elastic scattering, the kinetic energy is conserved. The magnitude of the momentum transfer is linked with the scattering angle and the wavelength of the incident beam.



$|\mathbf{K}_f| = 2\pi / \lambda$
Wave Number
$|\mathbf{K}_i| = 2\pi / \lambda$
θ
θ
2θ Scattering Angle
$|\mathbf{Q}| / 2$
$\mathbf{Q} = \mathbf{K}_i - \mathbf{K}_f$
$-|\mathbf{K}_f|$
$\frac{|\boldsymbol{Q}|}{2} = |\boldsymbol{K}_i| \sin\theta = \frac{2\pi}{\lambda} \sin\theta$

Figure 28: Physical basis of momentum transfer vector. Because energy is conserved in elastic scattering the initial wavenumber K_i = the final wavenumber K_f. Q is the difference vector between K_i and K_f. Using basic trigonometry, Q is derived from the sin definition.

$$Q = |Q| = \frac{4\pi sin\theta}{\lambda} \qquad 4.3$$

The differential cross section

Consider a steady stream of thermal neutrons (flux) all with the same energy incident on a target. The cross section is obtained from measurements made on neutrons after they interact with the scattering system. It represents the effective area presented by a nuclease to an incident neutron and is quantified as the fraction of incident particles that scatter. The differential cross section is the number of scattered neutrons per second into a solid angle divided by the flux.

The differential cross section is linked to the elastic scattering law and scattering intensity. All scattering interactions are assumed to be elastic. This means the incident beam contains particles of the same energy as the scattered beam. With energy fixed, the scattering of a neutron is characterized by a change in momentum P. Changes in momentum vary depending on the atom type and are characterized by neutron scattering lengths. Neutron scattering lengths are experimentally measured for different atoms. They do not vary with the atomic number in a predicable way. The magnitude of the scattering length (b) determines the strength of scattering. The values of scattering length depend on: 1) particular isotope of the element, 2) The spin state of the nucleus-neutron system, 3) Every nucleus with non-zero spin has two values of the scattering length. If the spin of the nucleus is zero, the system can only have spin ½ and there is only one value of the scattering length. Furthermore, b is positive for repulsive potential. Neutron scattering lengths are independent of the modulus of the momentum transfer vector. The formula to compute a SANS profile from a rigid body is:

$$I(q) = \sum_{i=1}^{m}\sum_{j=1}^{m} b_i(q) b_j(q) \frac{\sin(qr_{ij})}{qr_{ij}} \qquad 4.4$$

where b_i and b_j are the neutron scattering lengths and the sinc function is the orientational averaging. The neutron scattering lengths are computed as follows:

$$b_i(q) = b_i - c_1 f_{s,i}(q) + \ c_2 S_i(2b_h + b_o) \qquad 4.5$$

The major differences between the Debye implementation in SANS from SAXS is: 1) the neutron scattering lengths of atoms and atomic groups are used to evaluate form factors. 2) In solutions with D_2O fractions 0 < Y <1 all hydrogen atoms in hydrophilic (NH, NH_2, NH_3, OH, SH) groups are replaced with probability Y. The main chain NH groups are replaced with probability 0.9Y. The scattering density for the excluded volume calculation was modified. The Scattering results are showed in the appendices for protein 1ENH.

CHAPTER V

SAS AS AN EXPERIMENTAL RESTRAINT FOR PROTEIN STRUCTURE PREDICTION

Overview

The combination of SAS experimental data with computational protein structure prediction algorithms provides an opportunity to predict structures closer to the native topology [118-120]. SAS profiles have been used to identify native-like protein models from a large set of alternative protein models [1, 10, 121]. Furthermore, SAS profiles have been used to filter models in protein structure prediction algorithms. [6, 8, 17, 18, 122] Because the SAS experimental technique represents proteins with spherically averaged election / neutron densities, multiple structures can be reconstructed from the same SAXS profile. 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 [123]. 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 [124]. Grishaev et al. used NMR and SAXS restraints to refine the solution structure of the 82-kDA enzyme malate synthase G [125].

Significance

BCL::SAS is a module inside of our *de novo* protein structure prediction algorithm BCL::Fold. The integration of BCL::SAS with BCL::Fold provides a means to utilize SAXS / SANS scattering profiles as an additional term in the potential energy function[51]. My algorithm computes complete SAS scattering profiles for complete protein models and an approximate scattering profile for models missing side chains and loop regions that are produced after the initial folding stages. I mentored Oanh Vu, a summer rotation student to compare the calculated scattering profile with the ‘experimental’ profile to identify likely protein structures. A lower SAXS score suggests a higher probability that the BCL::fold model has a similar topology to the native structure and vice versa. To prepare positive controls for testing, side chains and loop regions of crystallographic structures of 13 monomeric protein samples were omitted and then simulated and recovered by BCL::fold. These samples were obtained through the Northeast Structural Genomics (NESG) consortium[126].

Innovation:

During a protein folding run with BCL::fold we penalized models that deviated from the overall SAXS score by incorporating it as a term in the linear weighted scoring function. We compared our fold benchmark set with and without SAXS restraints and found that experimental SAXS improves sampling by a small margin.

We used both ab initio and rigid body modeling techniques to investigate the variability of the SAXS χ agreement score. Ab initio methods search for three dimensional shapes represented by beads that fit the experimental SAXS profile and are used for SAXS envelope construction[8, 127]. This method provides an ensemble of configurations that correspond to a given scattering pattern. In Rigid body modeling , a SAXS profile is computed from the atomic coordinates and compared with experimental data using the χ agreement score [122].

Compute SAXS scores

The SAXS χ score compares the similarity between a reconstructed profile (from a model produced through BCL::Fold) with its corresponding experimental SAXS profile. Since we want to include the overall shape of the SAXS profiles into the comparison, the χ score was computed based on the derivatives of the experimental and computed profiles.[41] To create a differentiable function representing the experimental data, and to minimalize variability of intensities at high q values, a fitted line of the experimental SAXS profile was generated using locally weighted scatterplot smoothing (LOESS) in R (Appendix VI). This fit line was used as a standard to evaluate each of the putative models generated by BCL::Fold.

Missing residues in the X-Ray Crystal Structure affect agreement between SAXS profiles

The method of X-Ray Crystallography for structure determination can only be used upon successful formation of crystals around the desired protein target. This crystallization process is unpredictable and in many cases unsuccessful. One of the reasons for failure is caused by long flexible regions of amino acids preventing the formation of crystals. To overcome this problem, crystallographers will cleave (when possible) the flexible region of the protein to enable crystallization and subsequent structure determination of the core. These floppy regions are oft times found at the N or C terminal of an amino acid sequence.

To investigate the effect these missing residues have on experimental SAXS profiles, we obtained the experimental SAXS profile of 3HZ7. This is a protein domain of unknown function. The domain is a monomer with 87 residues and a molecular weight of 9523 Da. The PDB contains coordinate information of 74 residues. The final 13 residues were missing. Using the modeler program in Chimera, we modeled 500 structures of this protein with the missing residues added. By modeling the missing residues on the C terminus of 3HZ7, we were able to improve the chi agreement from 5.24 to 1.33. (Figure 29) Modeling the missing residues caused the chi agreement score to vary between 1 and 6. This indicates that long floppy regions at the termini of proteins have an effect on the SAXS chi agreement score.

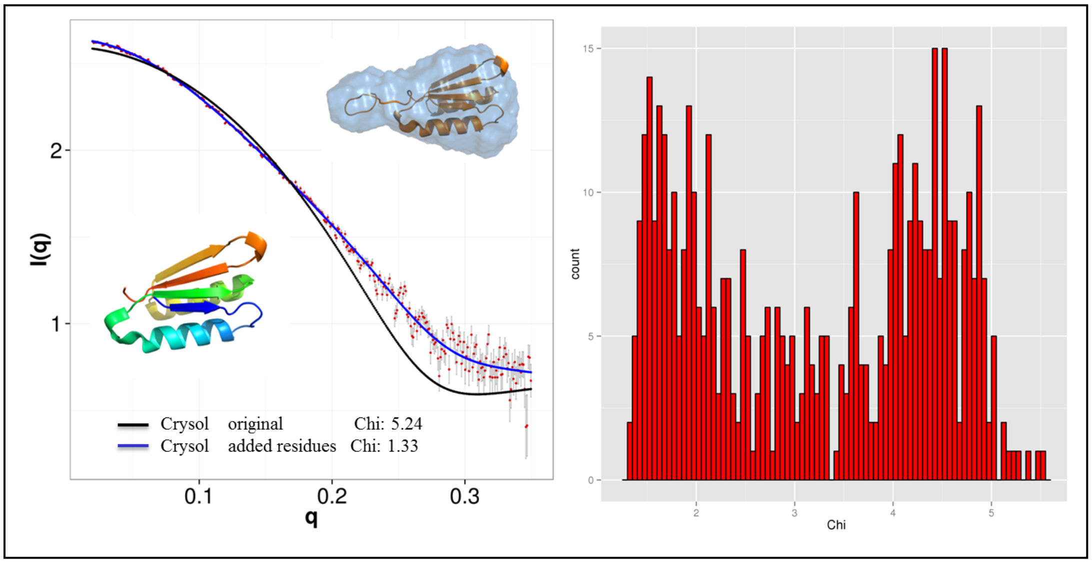


Figure 29: Modeling missing residues improves SAXS profile agreement. Shown in the left is the experimental SAXS profile (red dots) and error bars (gray lines) of 3HZ7. The green line is the computed SAXS profile with missing residues; the blue line is the best fit profile after modeling. The computed SAXS envelope from DAMMIN was computed and overlaid over the structure with the best chi agreement to experimental data (upper left corner). The right panel depicts the chi agreement distribution of the 500 models created in modeler.

SAXS experimental profiles and pdb files of crystallographic structures with missing residues from our benchmark set were used to explore loop modeling with Chimera. They were proteins 1_3HZ7, 6_3LYY, 9_3ICL, 10_3IGN, 12_3LJX, and 13_3HXL. These proteins ranged in size from 9.5 KDa to 48.5 KDa. (Table 11)

Table 11: Summary of 13 monomeric proteins ranging in size from 9.5 KDa to 48.5 KDa

Protein	Name	Molecular Weight Da	Observed Residues	Missing residues
1_3HZ7	Domain of Unknown Function	9523	74	13
6_3LYY	MucBP domain of adhesion PEPE_0118	14300	102	5
9_3ICL	EAL/GGDEF domain protein	18738	162	9
10_3IGN	Diguanylate cyclase	20256	165	12
12_3LJX	MmoQ (Response regulator)	32032	252	36
13_3HXL	Putative uncharacterized protein	48519	416	30
18_2KW9	MKL/myocardinlink protein 1	8276	75	0
20_2KVZ	Putative peptidoglycan bound protein	9712	85	0
21_2LOB	E3 ubiquitin-protein ligase Praja 1	10297	91	0
22_2KZ5	Transcription factor NF-E2	10623	91	0
24_2L0D	Cell surface protein	12385	114	0
26_2KW7	N-terminal PG_0361 from P. gingivalis	17485	157	0
28_3LD7	Lin0431 protein	12747	87	13

FASTA files of full protein sequences were attained from the protein data bank (PDB). Missing residues were added by to the crystal structures using Chimera. (Table 12) There was one case where the addition of missing residues caused an increase in χ agreement score. This was protein 13_3HZL. In this case the N-terminal residues protrude away from the protein instead of filling in the open space in the envelope. (Figure 30) Modeler is not a stochastic process, and may not have sufficiently sampled the conformational space.

Table 12: Effect of modeling missing residues with modeler

Protein	Residues Observed	Total Residues	Residues Missing	Missing residues χ	Modeled residues χ
1_3HZ7	74	87	13	4.2	1.3
6_3LYY	107	131	24	6.1	5.4
9_3ICL	162	171	9	3.7	1.6
10_3IGN	165	177	12	4.2	2.5
12_3LJX	252	288	36	2.9	2.3
13_3HZL	416	446	30	1.4	2.5

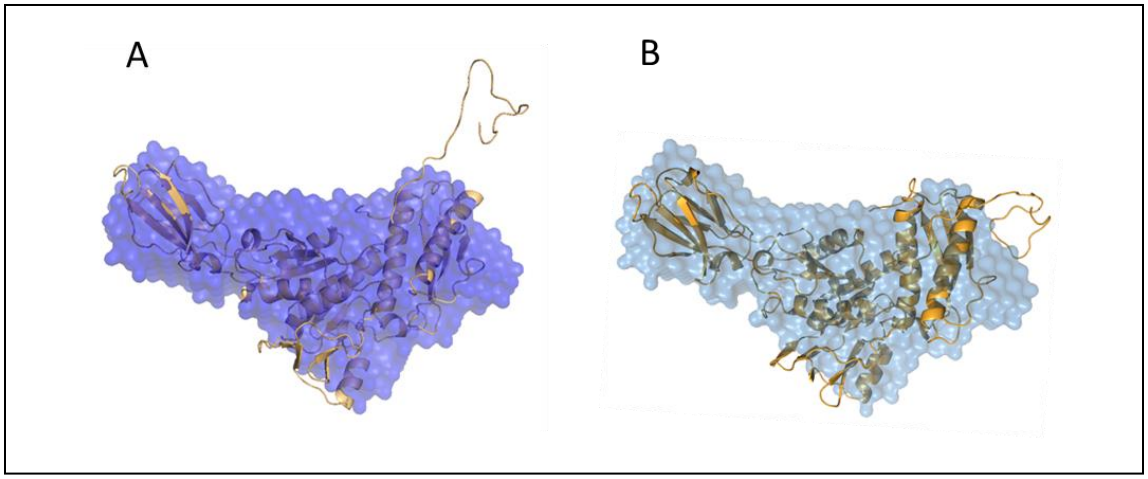


Figure 30: Structure and SAXS Envelope of 13_3HXL. Five ab initio shape reconstructions were generated by DAMMIF and averaged with DAMAVER. The best scoring model by chi agreement is superimposed in envelope using SUPCOMB. A) Modeled residues with Modeller B) Modeled residues with Rosetta

Rosetta models missing residues

To resolve the protruding loop (Figure 30). The loop modeling application in Rosetta version 2014.35.57232 (Appendix VI) was used to relax the top scoring model from modeler. For each of the 500 structures relaxed, we used CRYSOL to compared experimental profile with the simulated profile from the model. (Table 13) In the case of 13_3HZL, Rosetta modeling did not improve the χ agreement. In the remaining cases, modeling the missing residues with Rosetta improved the χ agreement between experimental data.

Table 13: Effect of relaxing Chimera models with Rosetta

Protein	Residues Observed	Total Residues	Residues Missing	Missing residues χ	Modeled residues χ
1_3HZ7	74	87	13	4.2	1.3
6_3LYY	107	131	24	6.1	5.1
9_3ICL	162	171	9	3.7	2.9
10_3IGN	165	177	12	4.2	2.0
12_3LJX	252	288	36	2.9	2.3
13_3HZL	416	446	30	1.4	1.4

We cut out the modeler step entirely and used Rosetta to model the missing residues for the entire benchmark of monomeric proteins and added one more protein 28_3LD7 to the set. (Table 14)

Table 14: Effect of modeling missing residues with Rosetta only

Protein	Residues Observed	Total Residues	Residues Missing	Missing residues χ	Modeled residues χ
1_3HZ7	74	87	13	4.2	1.3
6_3LYY	107	131	24	6.1	5.0
9_3ICL	162	171	9	3.7	1.6
10_3IGN	165	177	12	4.2	2.4
12_3LJX	252	288	36	2.9	2.2
13_3HZL	416	446	30	1.4	1.7
28_3LD7	87	100	13	7.4	2.4

The χ range of correct secondary structure topology

To explore the behavior of the χ score, a set of 1000 folding models were generated for each of the 13 protein samples using BCL::Fold without using the SAXS score as a restraint. For each of the 50 models, SAXS profiles were computed and compared with experimental data for three levels: 1) without approximation, 2) with side chain approximation and 3) with both side chain and loop approximation. Table 15 depicts the effect of comparing models without loops or side chains with experimental SAXS data. Figure 28 shows the folding results of model 1_3HZ7. The remaining models are shown in Appendix VIII.

Table 15: Statistics of 1000 BCL::Folding models for each protein. The minimum χ score obtained (min), maximum χ score obtained (max) mean chi score (ū), standard deviation (σ) and χ score of the native protein without side chains or loop regions (N) are reported for each level of approximation.

	No Approximation					Side Chain Approximation					Loop Region and Side Chain Approximation				
Model	min	max	ū	σ	N	min	max	ū	σ	N	min	max	ū	σ	N
1_3HZ7	1.4	5.1	3.3	0.7	4.75	1.5	5.1	3.1	0.7	4.16	1.5	3.6	2.4	0.4	2.24
6_3LYY	1.6	3.4	2.5	0.2	2.44	1.7	3.5	2.5	0.3	2.59	1.8	3.7	2.7	0.3	2.40
9_3ICL	3.4	7.5	5.2	0.9	3.36	2.5	6.8	4.5	1.0	2.46	2.0	5.1	3.5	0.6	3.02
10_3IGN	2.3	6.5	4.1	0.9	2.50	1.7	5.7	3.7	0.9	1.69	1.5	5.3	2.9	0.7	2.58
12_2L0D	3.0	9.1	5.6	1.1	3.30	3.0	8.8	5.5	1.2	2.98	2.1	10	5.4	1.1	2.10
13_2KW7	2.9	6.6	4.3	0.9	3.37	2.7	7.2	4.4	1.0	3.35	1.8	6.0	4.0	0.9	1.82
18_2KW9	2.8	3.7	3.1	1.1	3.20	2.6	3.5	3.0	0.1	3.16	2.5	3.4	3.1	0.1	3.17
20_2KVZ	1.5	3.6	2.5	0.5	2.80	1.5	3.4	2.4	0.4	2.90	1.3	3.4	2.1	0.3	2.81
21_2LOB	1.2	3.8	2.3	0.6	2.37	1.2	3.5	2.2	0.6	2.66	1.4	3.6	2.1	0.3	3.56
22_2KZ5	2.7	4.7	4.0	0.4	2.79	2.7	4.8	3.9	0.4	2.91	2.6	4.7	3.8	0.4	3.04
24_2L0D	2.0	5.2	3.0	0.6	3.52	2.0	5.3	3.0	0.7	3.49	2.1	5.3	3.3	0.6	3.15
26_2KW7	6.0	11.0	8.4	0.9	6.06	4.6	11.0	8.0	1.1	4.62	4.0	9.1	6.9	0.9	4.02
28_3LD7	1.7	6.4	3.9	1.1	3.5	1.7	5.6	3.6	1.1	3.31	1.5	4.2	2.7	0.7	2.83

Without approximations the native topology of model 1 had a χ score of 4.75. At each subsequent approximation, the χ score decreased to 4.16 (Side chain approximation) and 2.24 (Loop region and side chain approximation). This ideal behavior was not observed in all cases. Model 6, 9, 10 all deviated from this pattern. For model 6, the side chain approximations caused SAXS χ agreement score to increase. In this case the model has long disordered regions that are not captured in SSEs. These disordered regions are present in the experimental SAXS data, but only approximated by BCL::SAXS. In models 9 and 10, the loop region approximations caused the SAXS χ agreement score to increase.

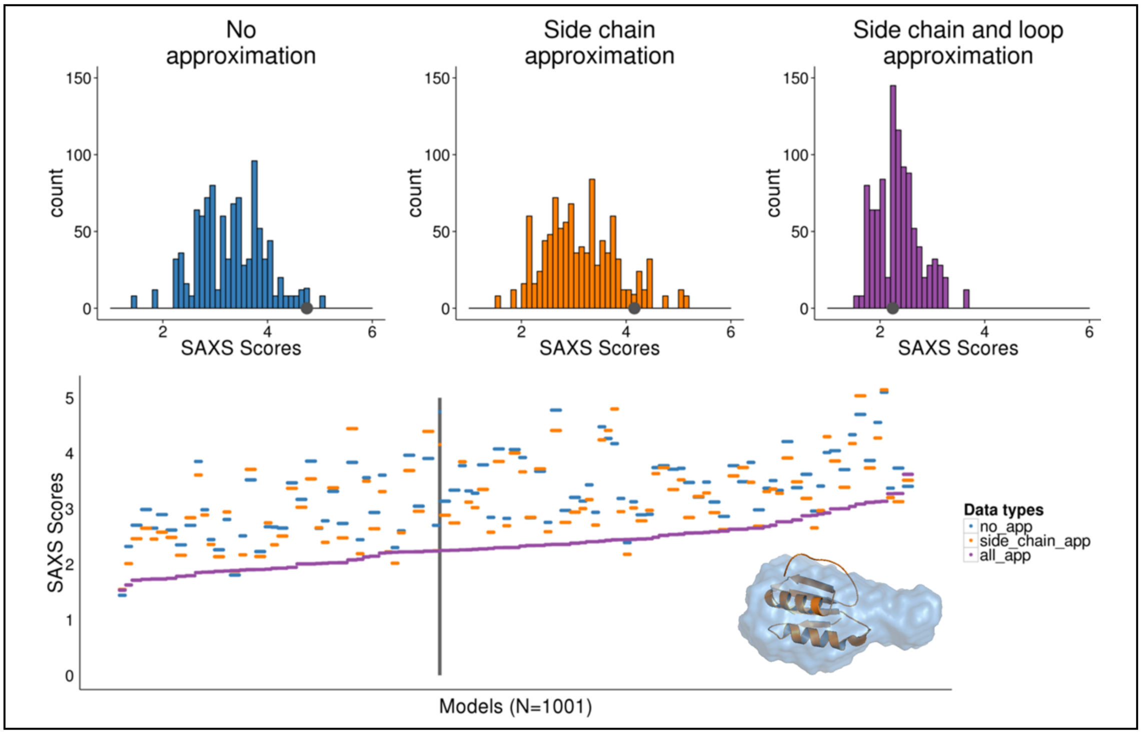


Figure 31: Distribution of SAXS agreement χ score for 1_3HZ7 with three states of approximation, blue, orange, purple (Top panels). The gray data point represents the score of the native structure. The bottom plot depicts the SAXS score for each model in three approximation states sorted by purple

To explore how BCL::Fold assembles proteins, we captured the ending pose of a model during the Monte Carlo sampling process as well as the confirmations of the secondary structure elements (SSES) in the native state. By doing this, one can quickly observe the difference between the native SSE arrangement and the SSE arrangement produced by BCL::Fold. The SAXS score is degenerate, meaning that structures that are very different can have similar χ agreement score. Because of this the SAXS χ agreement is a necessary, but not sufficient condition for protein identification by SAXS.

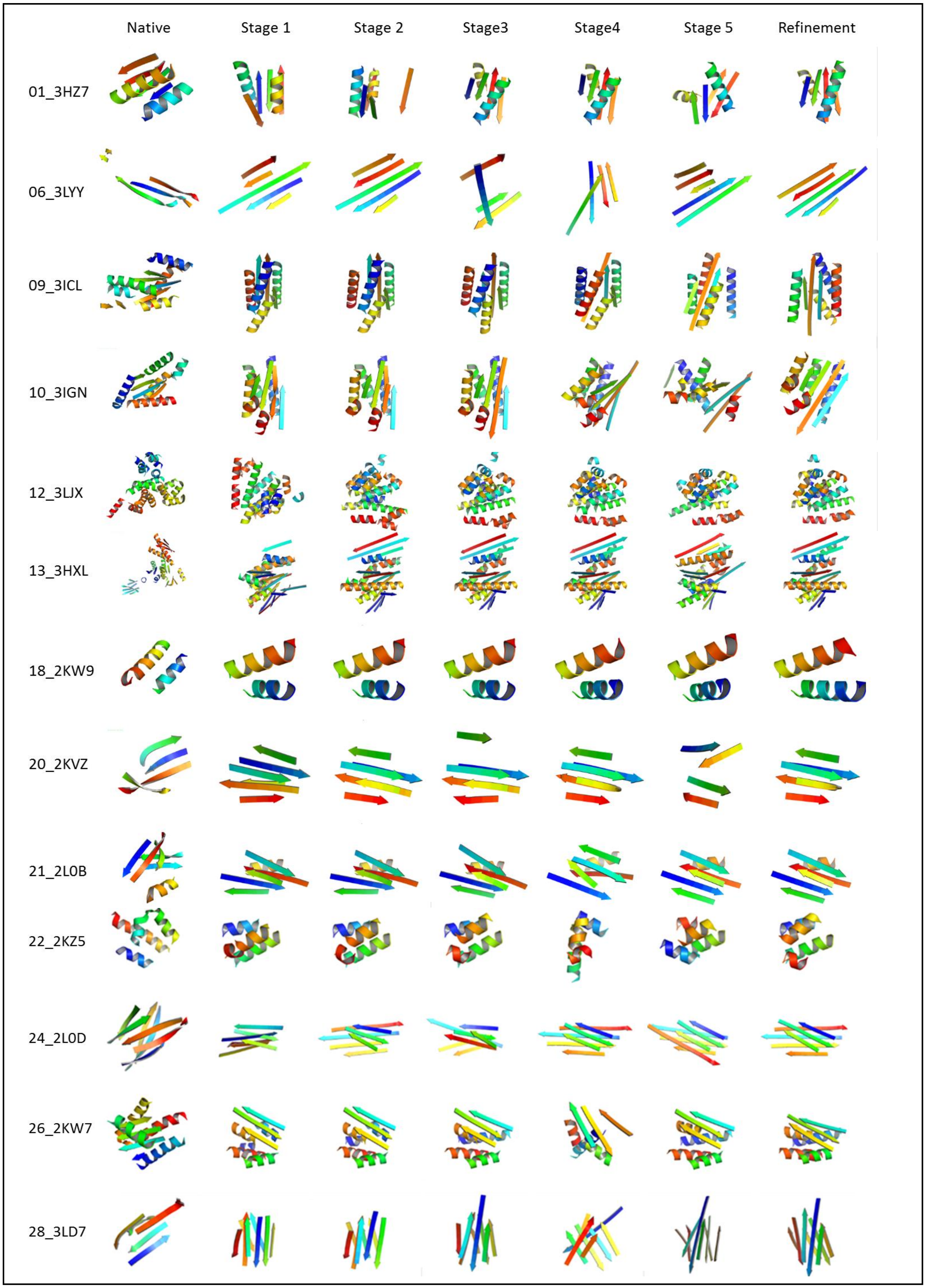


Figure 32: SSE orientation of different protein models at the end of each folding stage of BCL::Fold. Different protein topologies have similar χ agreement scores.

BCL::Fold with SAXS Restraint Score

For each of the 13 proteins I generated 2000 models through BCL::Fold without using the SAXS restraint during the simulation. The first 1000 models were created with the native SSE pool, while the second 1000 models were created using a predicted SSE pool. After generating the models, I computed the SAXS agreement score and enrichment score for each model. (Table 16)

Table 16: Folding Results for 13 protein models without using the SAXS restraint during simulation. Either predicted or native SSE pools were used during folding. The minimum rmsd100 and mean rmsd100 and standard deviation for the top 10% of the models by RMSD100 score are reported as well as SAXS score enrichment.

Model	Pool	Min RMSD100	Mean RMSD100	SD RMSD100	Enrichment
1_3HZ7	Native	3.59	7.54	1.58	0.4
	Predicted	2.40	5.10	0.90	0
6_3LYY	Native	10.46	20.14	3.33	1.9
	Predicted	20.79	27.28	1.84	1.0
9_3ICL	Native	8.59	11.12	0.65	1.4
	Predicted	8.45	11.85	0.78	0.9
10_3IGN	Native	8.53	10.94	0.79	0.9
	Predicted	8.16	10.95	0.76	1.4
12_3LJX	Native	9.82	11.60	0.49	1.2
	Predicted	11.69	13.03	0.46	1.2
13_3HXL	Native	15.76	17.02	0.42	1.3
	Predicted	14.50	16.33	0.53	0.9
18_2KW9	Native	10.03	15.65	1.71	2.8
	Predicted	5.75	17.04	2.61	3.4
20_2KVZ	Native	12.49	19.43	2.25	0.4
	Predicted	12.45	22.48	2.54	0.7
21_2LOB	Native	9.94	12.42	0.98	1.6
	Predicted	10.29	13.38	0.91	1.3
22_2KZ5	Native	7.99	10.52	0.83	1.2
	Predicted	7.29	10.72	0.86	3.0
24_2L0D	Native	8.50	11.62	1.01	1.0
	Predicted	8.18	12.14	1.03	0.4
26_2KW7	Native	7.73	10.21	0.87	1.5
	Predicted	6.62	10.02	0.82	1.5
28_3LD7	Native	6.74	11.11	1.25	0.8
	Predicted	9.11	12.23	1.02	1.1

The enrichment score is a metric to determine how effective a scoring metric (in our case the SAXS score) is able to select the most accurate models from a set of protein models. A given set of 1000 models from the above benchmark (S) was sorted by the RMSD100 score. The top 10% (100 models) with the lowest RMSD100 scores were classified as positive (P) and the rest of the models (900) were classified as negative (N). The models of S were scored by SAXS score and the top 10% (100 models) with the lowest SAXS score were classified as (T), while the remaining models were classified as (Z). The intersection of T and P (models correctly selected by the scoring function) were classified as true positive (TP). The number of models that were in set P but not in T represents models that are not identified by SAXS as correct, yet are similar to the native structure by RMSD100. This set is classified as FN (False Negative). The enrichment is calculated as

$$E = \frac{TP}{P} \cdot \frac{P+N}{P} \qquad 5.1$$

The positive models are the 10% of the models with the lowest RMSD100 values. P+N/P is a constant value of 10. The maximum enrichment score is 10.0. No enrichment would be a value of 1.0 and a value between 0.0 and 1.0 indicates that the SAXS score selects against accurate models.

Folding with the SAXS restraint

1000 models were folded with BCL::Fold using experimental SAXS data as restraint in the scoring function. These proteins were folded using the native secondary structure pool. During folding there were five assembly stages and one refinement stage. The first assembly stage had the weight of the SAXS score set to zero to allow the initial placement of secondary structure elements. All subsequent stages of folding with BCL::Fold had the weight of the SAXS score set to 100. With this weight, the total contribution of the SAXS restraint to the overall BCL energy function was approximately 10%. A perfect SAXS agreement score would be zero and would not impact the BCL energy score. Any score above zero would impact the overall BCL energy score by adding a positive value to the score.

Using the SAXS score during folding caused a slight shift toward native like proteins from an average RMSD100 value of 7.53 Å (without SAXS restraint) to 5.74 Å (with SAXS restraint) for protein model 01_3HZ7 (Fig 33A) and 20.1 Å (without SAXS restraint) to 19.45 Å (with SAXS restraint) for protein model 06_3LYY (Fig 33B)

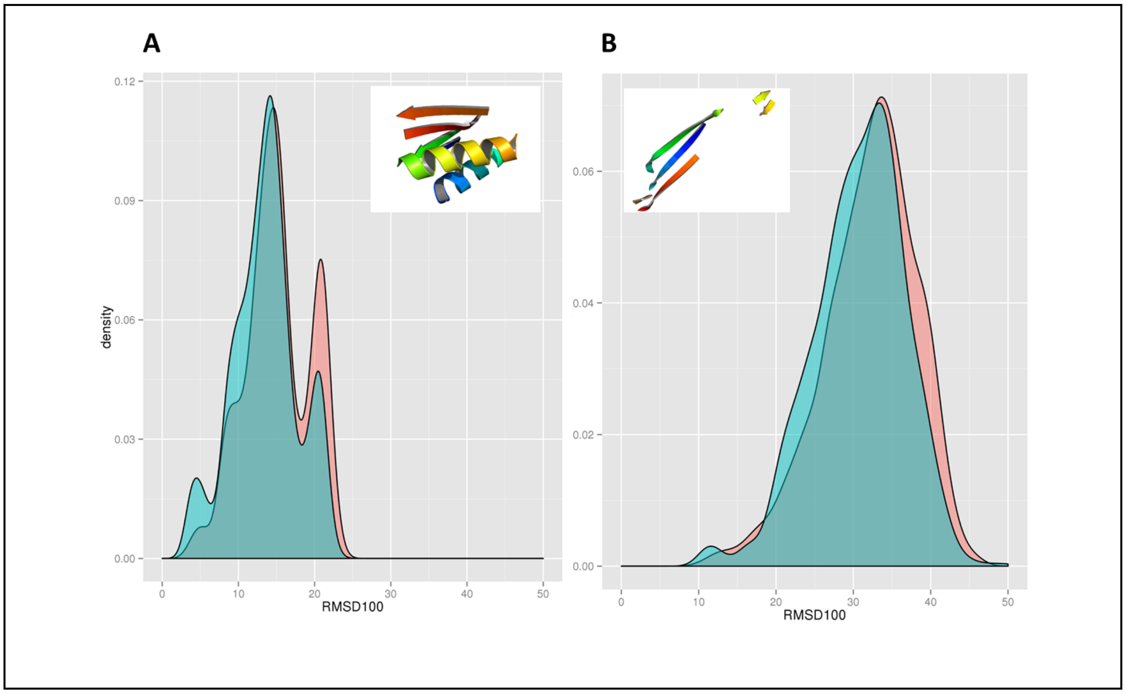


Figure 33: Folding of 1000 models with and without SAXS restraint with BCL::Fold with native SSE Pool. Panel A depicts protein 01_3HZ7. Panel B depicts 06_3LYY. Pink represents folding without the SAXS restraint, Blue represents folding with the SAXS restraint.

Table 17: Folding Results for 13 protein models using the SAXS restraint during simulation. Either predicted or native SSE pools were used during folding. The minimum rmsd100 and mean rmsd100 and standard deviation for the top 10% of the models by RMSD100 score. N is the number of distinct models in the set.

Model	Pool	Min RMSD100	Mean RMSD100	SD RMSD100	Enrichment	N
1_3HZ7	Native	3.18	5.74	1.71	0	1000
	Predicted	2.50	4.71	0.75	0	50
6_3LYY	Native	10.62	19.46	3.39	1.7	1000
	Predicted	20.00	26.57	2.14	0.9	50
9_3ICL	Native	8.04	11.26	0.75	1.4	50
	Predicted	9.73	11.91	0.60	1.1	50
10_3IGN	Native	8.57	11.09	0.72	0.6	50
	Predicted	7.35	10.9	0.73	1.9	50
12_3LJX	Native	9.36	11.59	0.51	1.2	50
	Predicted	11.64	13.04	0.56	1.2	50
13_3HXL	Native	14.51	16.93	0.61	1.4	50
	Predicted	14.81	16.35	0.55	0.1	50
18_2KW9	Native	9.26	14.83	1.24	1.3	50

	Predicted	12.88	16.72	1.25	3.3	50
20_2KVZ	Native	12.54	18.18	2.20	1.2	50
	Predicted	12.49	22.55	2.49	1.0	50
21_2LOB	Native	9.86	12.87	0.96	1.7	50
	Predicted	10.84	13.50	0.93	0.6	50
22_2KZ5	Native	7.50	10.32	0.95	1.1	50
	Predicted	7.19	9.54	0.88	3.6	50
24_2L0D	Native	7.24	11.87	1.12	1.5	50
	Predicted	10.09	12.37	0.90	2.1	50
26_2KW7	Native	6.21	9.58	1.06	1.9	50
	Predicted	7.17	9.62	0.88	2.0	50
28_3LD7	Native	6.94	11.64	1.52	0.6	50
	Predicted	8.68	11.87	0.99	0.8	50

Discussion

This benchmark was designed to test two scenarios for both the native and predicted SSE pools: 1) The enrichment of the SAXS score on models folded without using the SAXS score as a restraint, and 2) The improvement in RMSD100 for models that use the SAXS restraint during folding.

Table 16 shows the sampling without SAXS restraint with the native and predicted SSE pools. Using the native SSEs, the top model by RMSD100 was above 8 Å for 9 of the 13 models. This indicates that BCL::Fold is not sampling the correct topology for these models. This could be due to either, 1) insufficient sampling number, 2) bias in sampling algorithm. I recommend increasing the sampling from 1000 to 10000 and observe the minimum, and mean RMSD100 values observed. If the values do not improve, then we should analyze the sampling algorithm to understand why we do not sample the native topology using the native SSE pool.

We observed positive enrichments with the SAXS score for the native SSE Pool on 8 of 13 models, neural enrichment of 1.0 for one model, and negative enrichments on 4 models. For the predicted pool, we observed positive enrichments on 7 of 13 models, neutral enrichment of 1.0 for one model, and negative enrichments on 5 models.

In this analysis, I discovered why the enrichment values do not behave as expected when folding with SAXS Restraints. The folding setup had 50 jobs with 20 models in each job to produce a total of 1000 models. I discovered that each batch received the same folding seed creating 1 model duplicated 20 times. Work is currently underway to expand the protein simulations from 50 models to 1000 models in this benchmark. For model 01_3HZ7, we observe a shift in RMSD100 from 7.54 to 5.74.

CHAPTER VI

CELLULOSE SYNTHASE

Overview

Cellulose is composed of β-1,4 linked D-Glucose monomers ($C_6H_{10}O_5$) and is the major structural component of the cell wall in plants. *Arabidopsis Thaliana* is a flowering plant native to Eurasia. It is a popular organism in plant biology because of its small genome size (135 mega base pairs) and number of chromosomes (5). It is to plant biology what mice and fruit files are to animal biology. In *Arabidopsis Thaliana*, the individual cellulose synthase (CESA) protein has 10 isoforms. These isoforms are 64% to 98% similar by sequence identity comparison. CESA1, CESA3, and CESA6 are involved in primary cell wall synthesis. CESA4, CESA7, and CESA8 are involved in secondary cell wall synthesis. CESA2, CESA5, and CESA9 are involved in tissue-specific processes. CESA10 has a minor role in plant development. Each isoform of CESA contains approximately 1000 amino acids comprising 3 domains: 1) Zinc-finger n-terminal domain, 2) cytosolic catalytic domain, and 3) 8-helix transmembrane domain. (See Figure 34)

Significance

The department of energy (DOE) is interested in understanding how cellulose is produced, how it can be torn down, and how we can produce plants with weaker cell walls. This would facilitate the production of biofuels. Cellulose in plants is produced by the cellulose synthase complex (CSC) that is compromised by 6 lobes in hexagonal arrangements with a diameter between 24-40 nanometers (nm) forming a “Rosette”.

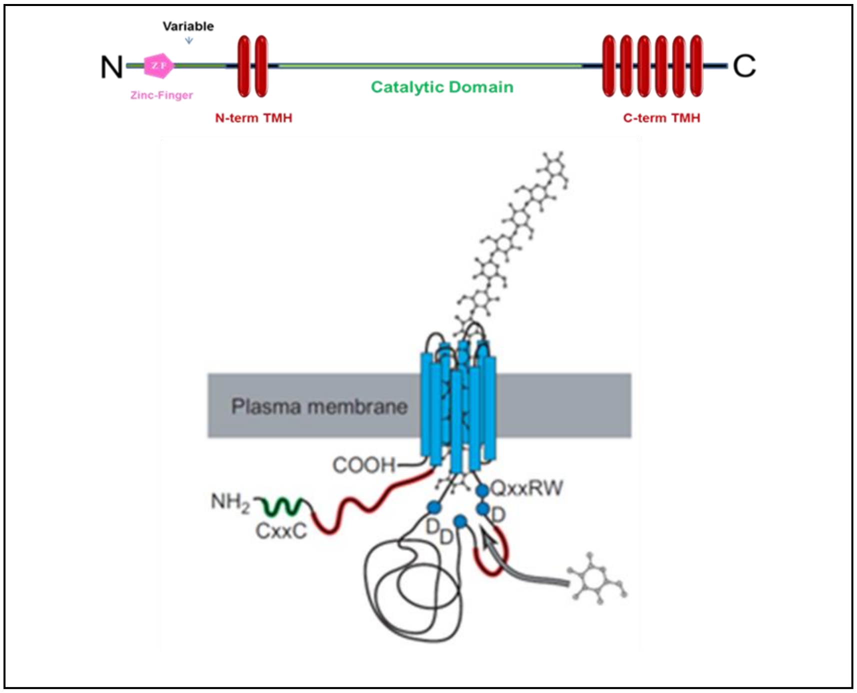


Figure 34: Schematic of atCESA3. Top depicts where domains occur in sequences space. There is a zinc-finger in the n-terminal intrinsically disordered domain, followed by 2 transmembrane helices. The catalytic domain is on the cytosolic side of the membrane. The C terminal domain contains the rest of the transmembrane helices. Bottom depicts a cartoon of how the domains may form the atCESA1 protein. The formed cellulose is shown emerging from the transmembrane domain

Innovation:

I was awarded a graduate fellowship to work at Oak Ridge National labs to study the structure of cellulose synthase. While there, I compiled BCL::Fold and Rosetta on the Titan Supercomputing cluster at ORNL. I used Rosetta Fold and Dock to create 100,000 putative dimer models of the zinc-finger domain of atCESA1. To cluster these models, I created a novel algorithm to identify the interface residues and cluster models that contain similar residues participating in the interface.

I was part of a publication on the catalytic domain of atCESA1 that combined experimental SAXS / SANS data with putative Rosetta models to show evidence of trimer formation. The combination of limited experimental data with computation provided a results superior to one that would be obtained by either method in isolation. This publication was featured on the cover of Plant Physiology in January of 2016.

N-terminal dimer domain

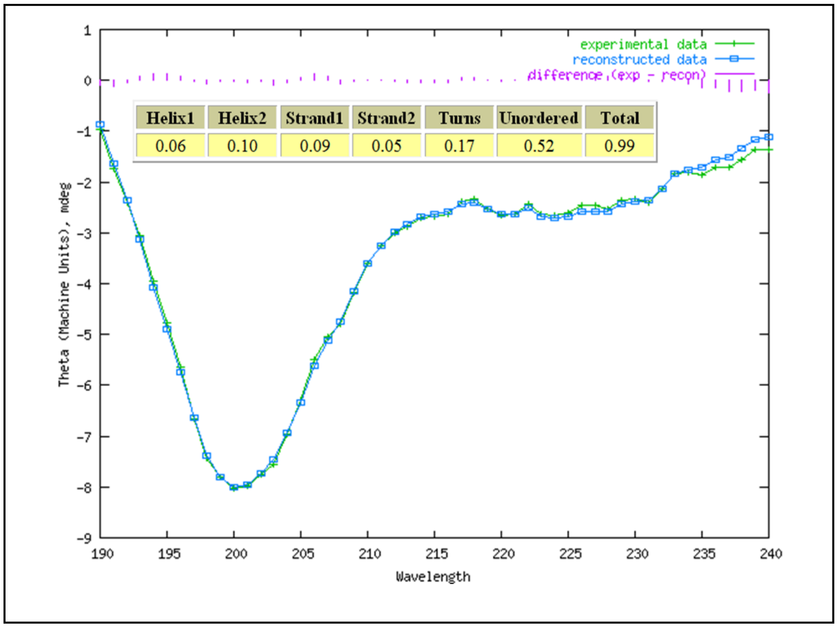
Hugh O'Neill and his team at Oak Ridge National Laboratory were able to express and purify samples of the n-terminal Zinc-finger n-terminal domain of atCESA3. The experimental data indicated that this domain formed dimers.

To explore predicted dimer interfaces of the Zinc-finger N-terminal domain (ZFNTd), I generated 100,000 putative dimer models of the ZFNTd of atCESA3 using the Rosetta Fold and Dock algorithm on Titan. This domain comprises 247 amino acid residues.

The Rosetta Fold and dock algorithm consists of multiple folding stages beginning with two extended amino acid chains. The backbone atoms of both chains are simultaneously moved and the energy is evaluated by considering hydrogen bonds and hydrophobic interactions. During each stage of folding Monte Carlo sampling is used to sample the conformational space and a final pose is selected for the next stage of folding. In the latter stages of folding side chains are added using rotamer libraries and energy evaluations include Van der Waals interactions and hydrogen bonds.

These models were filtered by agreement with circular dichroism (CD) experimental data, small angle neutron scattering (SANS) data, agreement with an homology model, and probable interface score. After filtering by these means, 4 models remained for further analysis.

Once the 100,000 models were generated on titan by Rosetta, I filtered the models output by agreement with any available experimental data. I obtained circular dichroism (CD) experimental data of atCESA3 from the experimental group at ORNL. The CD data showed the alpha helical content the atCESA3 dimers to be less than 20%. (Figure 35)

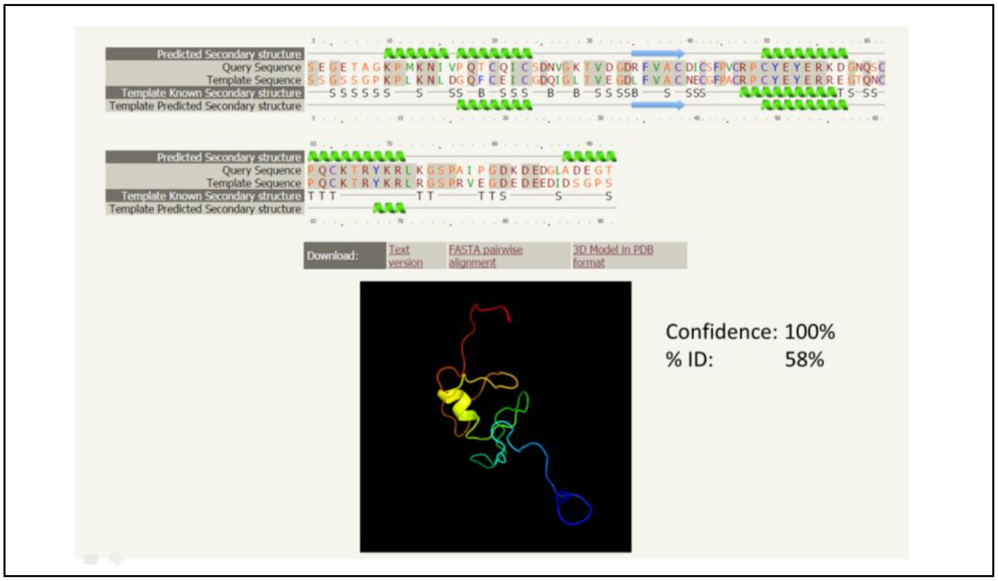


Helix1	Helix2	Strand1	Strand2	Turns	Unordered	Total
0.06	0.10	0.09	0.05	0.17	0.52	0.99

Figure 35: Circular dichroism (CD) data of atCESA3 using CDSSTR method. Analysis indicates that the alpha helical content of the sample is not greater than 16% with minimal difference between experimental and reconstructed data.

Using the Dictionary of Secondary Structure of Proteins, (DSSP) I computed the secondary structure content of all the 100,000 computational dimer models. I then wrote a script to compute the alpha helical percentage content based on the DSSP assignments and merged this into the pdb model. Based on CD data agreement, I filtered over 90% of the models generated. There were 9314 models remaining.

I then filtered the 9314 models by agreement with a homology model of the Zinc Finger domain. (Figure 36) The homology model was created for residues 1-91 of the nuclear magnetic resonance (NMR) structure 1WEO. The core of this structure from the NMR ensemble was between residues 18-70. Visual molecular dynamics (VMD), a molecular visualization program for analyzing large biomolecular systems was used to align the homology model and the Rosetta models. I computed the root mean squared distance (RMSD) between residues 18-70 of the homology model and all remaining 9314 models. Only structures with an RMSD < 8 Å were kept for further analysis. After this step, 200 models were remaining.



```
Predicted Secondary structure
               Query Sequence  S E G E T A G K P M K N I VP Q T C Q I C SDNVG K T V D GDR F V A C DI CSFPVCR P C Y E Y E R K D GNQSC
            Template Sequence  S S G S S G P K P L K N L DG Q F C E I C GDQI G L T V E GDL F V A C NECGFPACR P C Y E Y E R R E GTQNC
Template Known Secondary structure  S S S S S S   S   SS  B  S S S  B  B  S S SSB   S  SSS   T S  SS
Template Predicted Secondary structure

Predicted Secondary structure
               Query Sequence  P Q C K T R Y K R L K G S PA I P G D K D E DGLA D E G T
            Template Sequence  P Q C K T R Y K R L R G S PR V E G D E D E EDID S G P S
Template Known Secondary structure  T T T   T T   T T S   S   S
Template Predicted Secondary structure
```

Download: Text version | FASTA pairwise alignment | 3D Model in PDB format

Confidence: 100%
% ID: 58%

Figure 36: Threaded homology model of AtCESA3. Phyre identified the NMR structure 1WEO as a strong homology model for residues 1-90. I threaded the n-terminal sequence of atCESA3 onto all 20 poses from 1WEO. Then I generated 100 homology models for each of the 210 poses using Rosetta. The final threaded homology model was selected based on the Rosetta energy score and the n-terminal position.

SANS profiles for each of the models were computed using CRYSON and compared with the experimental SANS profiles obtained at ORNL. The similarity between the experimental SANS profile and the computed SANS profile from each model was measured as a χ score. Models with a chi χ ≥ 1.4 were filtered from further analysis. This leaves 13 models.

To cluster the remaining 13 models by interface similarity, I developed an interface clustering algorithm. This algorithm first creates a numerical descriptor vector based on a user defined interface distance for each model. For this project we selected 6 Å as the cutoff distance to define an interface. The interface of each model was represented as a vector of zeros and ones, with one indicating a particular residue is an interface residue. The vectors were compared using confusion matrix classification and a Matthews's correlation coefficient. The correlation coefficient had a range of [-1, 1] with -1 indicating a complete opposite correlation and 1 indicating a perfect match between two vectors. The correlation ranges were ranked as follows: 1) +0.7 or higher, very strong correlation, 2) +0.4 Strong correlation, 3) +0.3 Moderate correlation, 4) +0.2 Weak correlation. Structures with zero residues identified as interface residues were filtered from further analysis. One model was removed based on this criterion, leaving 12 structures. (Figure 37)

		Prediction	
		0	1
Actual	0	TN	FP
	1	FN	TP

Name	Descriptor
Model 1	000101000100001001100010010010010001010000
Model 2	00101010111101111101101010000111000110001111
Model 3	000101000100001001100010010010010001010001

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Score	Agreement
+0.70 or higher	Very Strong
+0.40	Strong
+0.30	Moderate
+0.20	Weak

Figure 37: Novel Clustering Algorithm. Interface residues are identified by a pre-specified cutoff distance. Residues participating in the interface are assigned a value of 1, while residues not part of the interface are assigned a value of 0. The interface is described by a numerical descriptor vector with each position representing an individual amino acid. The models are processed through a pair-wise comparison algorithm that compares each position of two models and classifies the agreement as a True positive, False positive, True negative, False negative, and sums the total for each category. Using these numbers, the Matthew's correlation coefficient is computed, and modes that have an overall score > user defined cutoff [-1, 1] are clustered together.

For each of these 12 structures I used the protein interfaces structures and assembly (PISA) to quantify the interfaces. This algorithm computes the interface area, the delta G in kcal/mol, which quantifies the solvation free energy gain upon formation of the interface. It also computes the number of hydrogen bonds and salt bridges at the interface. Each interface was assigned a probability value which is a measure of interface specificity, showing how surprising, in energy terms, the interface is. Models with a favorable interface probability >0.5

were selected. This resulted in 4 different models selected from 100,000 putative models generated on Titan.

Table 18: Summary statistics of filtered n-terminal atCESA3 dimer models, REU is the Rosetta energy unit representing the final energy function value during minimization. Helix percentage represents the α helical content of the models. Homology root mean squared distance (RMSD) is a measure of how similar the zinc finger domain is with the homology model found. SANS χ is a measure of how similar the experimental SANS profile is with the computed SANS profile of the model. The interface area measures how large the interface is between dimers. Delta G indicates the solvation free energy gain upon interface formation. The Delta G p-value indicates the probability of observing a lower than observed Delta G, when the interface atoms are pick randomly from the protein surface. HB is the number of hydrogen bonds in the interface. SB is the number of salt bridges in the interface.

Model ID	REU	HELIX %	Homology RMSD	SANS χ	Interface Area	Delta G kcal/mol	Delta G p-value	HB	SB
2939	-360	0.150	7.6	1.24	2862	-36.0	0.164	16	12
3767	-365	0.166	7.7	1.40	3251	-34.8	0.219	28	6
1919	-336	0.138	7.1	1.29	3430	-31.6	0.310	36	2
8817	-398	0.166	7.5	1.21	2341	-20.3	0.390	26	12
3514	-330	0.166	8.0	0.94	2627	-19.0	0.504	40	16
8625	-340	0.166	8.0	0.91	706	-5.0	0.541	12	2
4108	-387	0.166	6.4	0.96	291	-1.2	0.572	2	0
0533	-353	0.150	6.7	1.28	2622	-17.4	0.589	34	20
6828	-348	0.150	6.5	1.28	1551	-9.9	0.589	10	4
9348	-400	0.158	6.5	1.03	3735	-26.6	0.598	48	44
9027	-353	0.166	7.4	0.93	2902	-17.9	0.599	40	12
0428	-339	0.142	7.7	1.03	1362	-5.5	0.608	22	20

The four models and their SANS agreement remaining are depicted in figures 38 and 39. The plateau in the low q region of the log-log plot indicates uniform scattering without aggregation. In this low q region, model 1919 is the only model to directly interpolate the data points. Models 8817 and 3767 are below the points, while model 2939 is above the points. This region represents large atomic pairwise distances. It is critical to have tight agreement between model and experiment in this region.

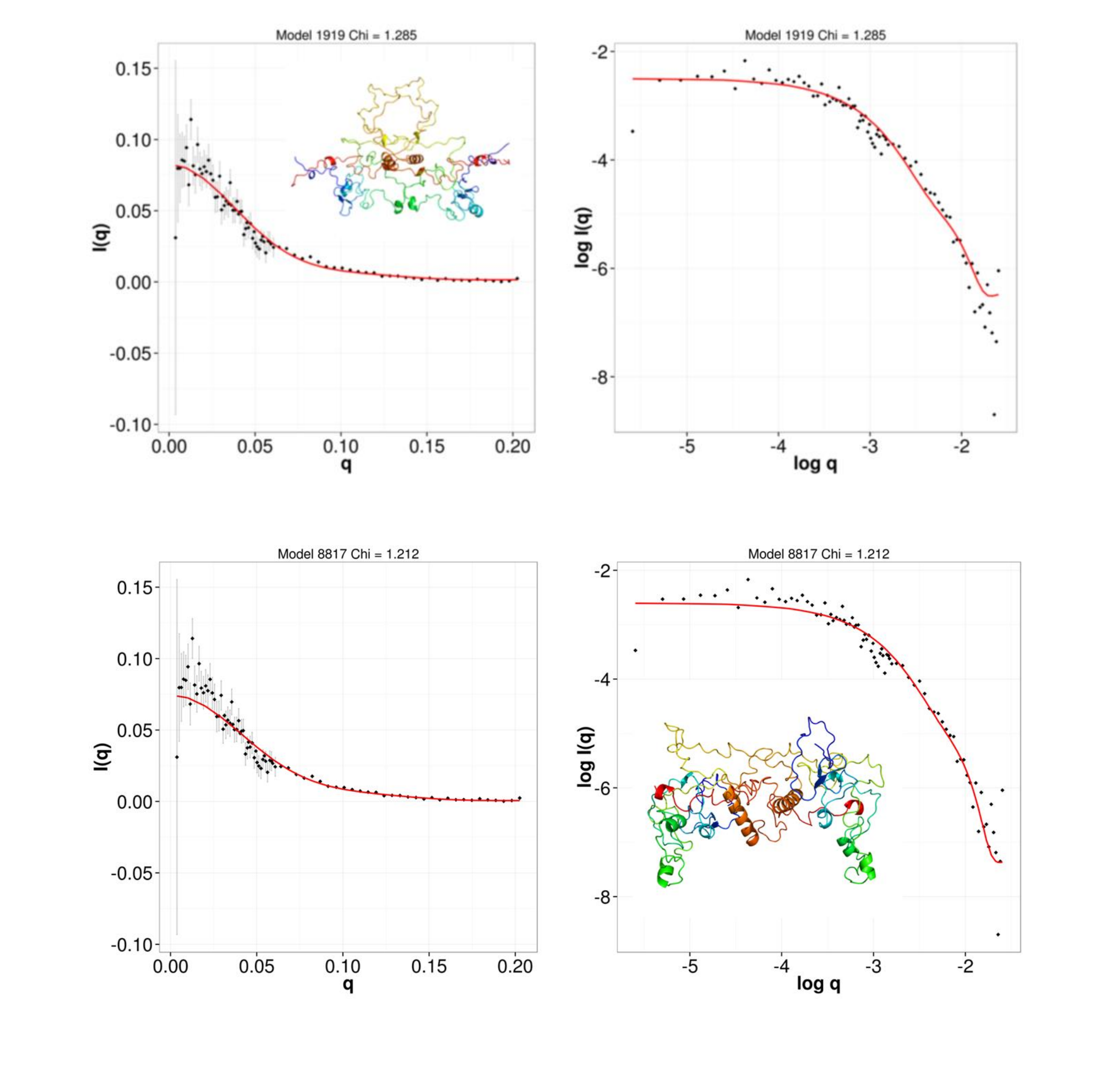


Figure 38: Agreement by SANS chi score between experimental data collected at ORNL and models on TITAN. The plateau in the experimental data (black dots) indicates uniform scattering without aggregation in the log log plot on right. The gray bars depict experimental error values.

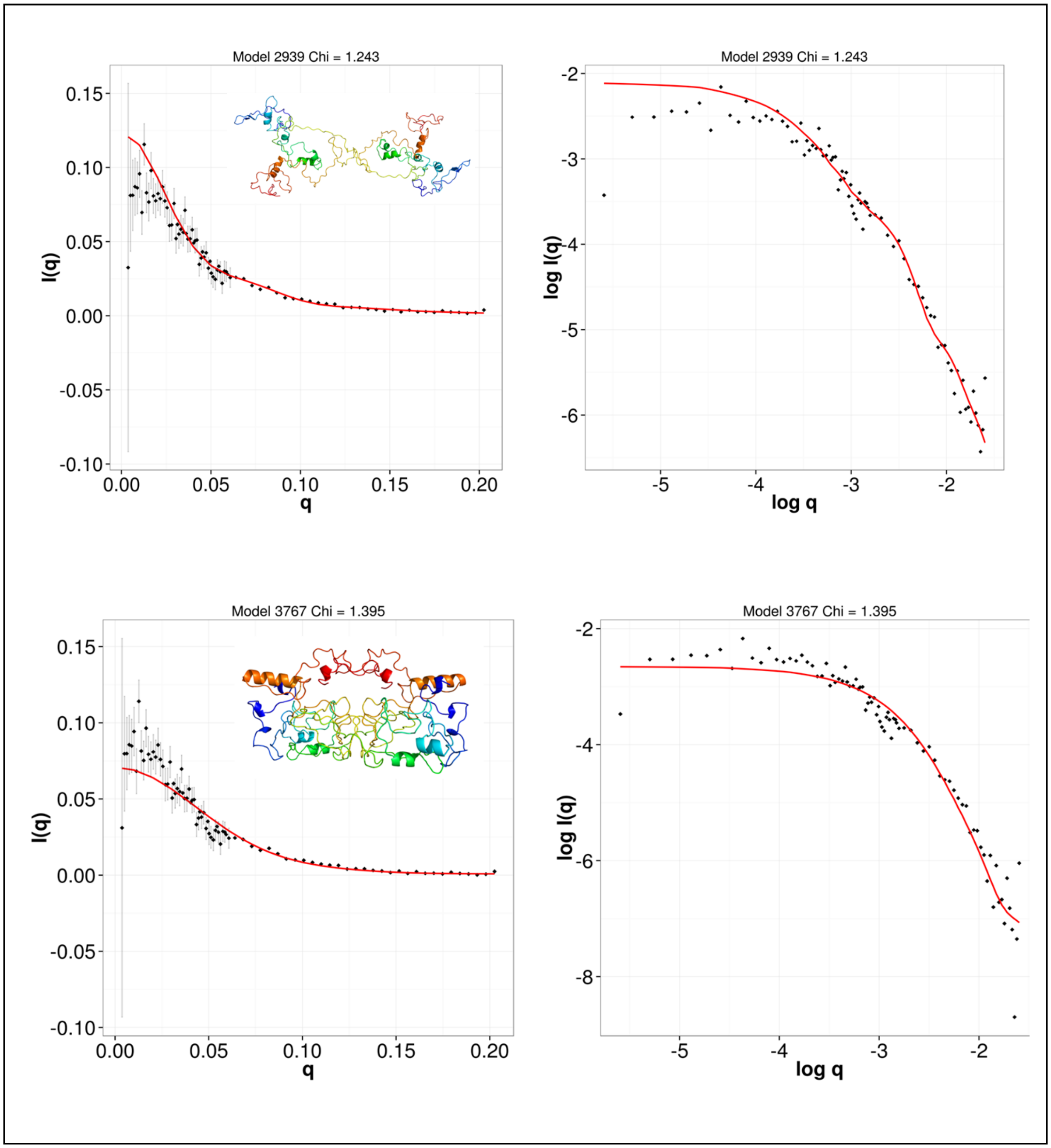


Figure 39: Agreement by SANS chi score between experimental data collected at ORNL and models produced on TITAN. The plateau in the experimental data (black dots) indicates uniform scattering without aggregation in the log log plot on right. The gray bars depict experimental error values.

N-terminal zinc finger domain discussion

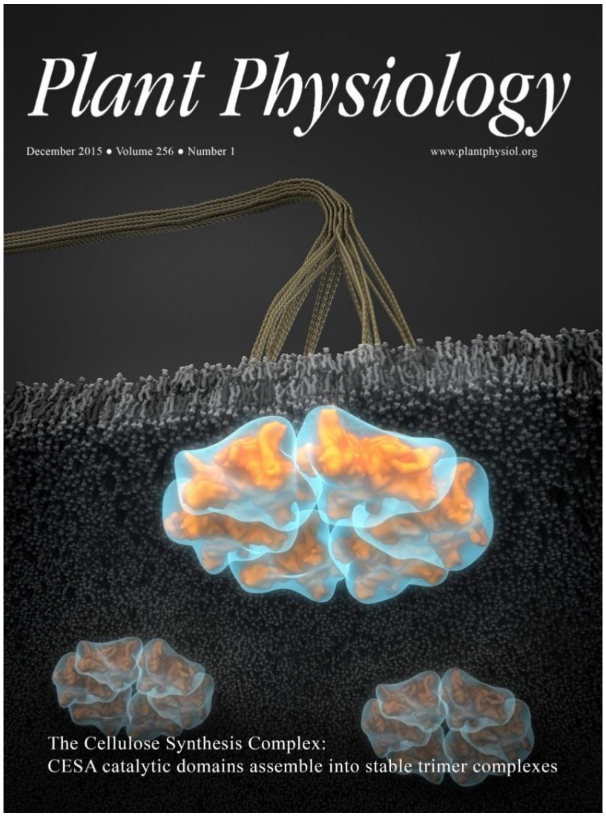
This was a challenging domain to model because it is intrinsically disordered. We had experimental evidence for dimer formation, but we do not know which residues form the dimer interface. We generated 100,000 putative dimer models with Rosetta fold and dock and filtered these models by agreement with SANS, CD, and PISA interface scores. I developed a method to cluster these models by interface similarity. The next step for this project is to analyze the 100,000 dimer models to quantify which residues are forming interfaces and how often they are involved in an interface. This information would be useful to create a list of predicted interface residues to guide the experimental team. Once the interface is experimentally verified, this information would provide a valuable filter on further modeling.

Cytosolic domain

This section is a reproduction of the computation modeling I did in the manuscript: A structural study of CESA1 catalytic domain of *Arabidposis* cellulose synthesis complex: evidence for CESA trimers[1]. The cellulose synthesis complex (CSC) is a large multi-subunit transmembrane protein complex responsible for synthesis of cellulose chains and their assembly into microfibrils in plants. This work reports a structural study of recombinant catalytic domain (residues 341 – 845) of Arabidopsis thaliana CESA1 (ATCESA1CatD) that was over-expressed and purified from Escherichia coli. Using a two-step procedure, it was possible to purify monomeric and trimeric forms of ATCESA1CatD, providing the first experimental evidence supporting the self-assembly of CESAs into stable trimeric complexes. The conformation of monomeric and homotrimeric ATCESA1CatD were studied using small-angle neutron scattering (SANS) and small-angle X-ray scattering (SAXS). A series of ATCESA1CatD trimer computational models were compared with the SAXS trimer profile to explore the possible arrangement of the monomers in the trimers. Four candidate trimers were identified with monomers oriented such that that newly synthesized cellulose chains project towards the cell membrane. In these models, the class specific region (CSR) is found at the periphery of the complex and the plant-conserved region (P-CR) forms the base of the trimer. This study strongly supports the hexamer of trimers model for rosette CSC that synthesizes an 18-chain cellulose microfibril as the fundamental product of cellulose synthesis in plants.[1]

Computational Modeling

A homology model of ATCESA1CatD was generated with the program MODELLER [96] using the GHCESA1CatD computational model [128] as a template. The root mean squared deviation (RMSD) value between the two models was 0.30 Å. The Symmetric Docking tool (SymDock) in the ROSETTA modeling software [129] was used to generate multiple configurations of symmetric trimers of the ATCESA1CatD model that were ranked based on a ROSETTA generated energy score. Simulated SAXS profiles of 1000 ATCESA1CatD trimer models with the lowest energy score were calculated using CRYSOL [26] and were fit to the experimental ATCESA1CatD SAXS profile. Based on this analysis, 30 trimer models with χ values smaller than 9.1 were chosen for further study. The χ value measures the discrepancy between experimentally determined and theoretically predicted small angle scattering data. The X-ray crystal structure of cellulose synthase from Rhodobacter sphaeroides (PDB code, 4HG6) [130] was structurally superposed on each monomer in the 30 trimer models using the DALILITE pairwise alignment tool [131]. The program PISA [132] was used to calculate the surface area in the interfaces in the trimers.



Plant Physiology

December 2015 • Volume 256 • Number 1

www.plantphysiol.org

The Cellulose Synthesis Complex:
CESA catalytic domains assemble into stable trimer complexes

Figure 40: Trimer models of atCESA1CatD on the cover of Plant Physiology. www.plantphysiol.org "Copyright American Society of Plant Biologists" [1]

Modeling of ATCESA1CatD trimers

To gain insight into possible arrangements of the ATCESA1CatD monomers, a series of ATCESA1CatD trimers was generated computationally and compared with the SAXS data (Figure 40). For this, a homology model of the ATCESA1CatD monomer was used as input for generating 1000 symmetric trimer configurations using the ROSETTA SymDock algorithm [129]. Theoretical SAXS curves were calculated using CRYSOL for each of the 1000 protein trimer configurations for comparison with the experimental SAXS data. The χ value was computed to quantify the fit of the theoretical SAXS curves to the experimental SAXS data. Comparison of the χ values with the ROSETTA energy score (E) in ROSETTA Energy Units (REU) of each trimeric model shows that, even for the trimer models with a low E value, there is large variation in the quality of the fit to the experimental SAXS data (Figure 41A). Furthermore, structurally very different models can have similar χ scores. This is exemplified in Figure 41B that compares the RMSD of each trimer model to the model with the lowest χ value (ATCESA1CatD-m1; χ = 4.39). Based on this observation, it is clear that one cannot rule out any particular arrangement of monomers based on the fit to the SAXS data alone.

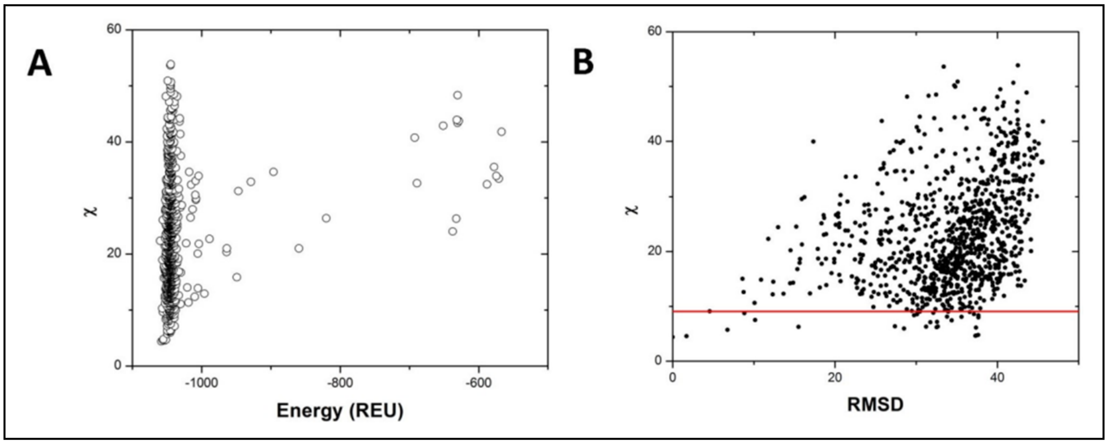


Figure 41: Analysis of ROSETTA generated ATCESA1CatD trimer models. (A) Plot of χ score obtained from fit of ATCESA1CatD trimer theoretical SAXs curves to experimental SAXS data versus the energy score of ROSETTA models. (B) A plot of χ versus RMSD of the ROSETTA models computed using ATCESA1CatD-m1 as a reference model. The models below the red line (χ < 9.10) were included in the analysis. www.plantphysiol.org "Copyright American Society of Plant Biologists" [1]

Since it is not possible to rule out a particular trimer model using solely the χ value, an additional constraint was needed to identify the most likely configurations among the best fitting, low E trimer models. Of the initial 1000 models generated by ROSETTA SymDock, 30 models were selected for detailed analysis based on the χ score (χ ≤ 9.1) obtained from the fits of the theoretical SAXS curves to the experimental SAXS data. In all cases, the theoretical scattering curves are similar in the low Q-region and fit the experimental data well, indicating that all trimer models capture the overall size of the scattering particle. However, theoretical curves deviate in the mid and high Q region (Q > ~0.08 Å-1) suggesting that differences in the arrangement of the monomers in the trimer are captured in that Q range. We can anticipate where the β-1,4-glucan chains will emerge from the monomers, given the structural similarity between the invariant DD, DCD, and QVLRW motifs that constitute the catalytic core in the

GHCESA1catD computational model and the crystal structure of *R. sphaeroides* bacterial cellulose synthase (BcsA) that also contains an emerging glucan chain [130]. Furthermore, class averaging of freeze-fracture TEM images of rosette CSCs indicate tightly associated TMH regions, which supports proximity of the glucan chains as they traverse the membrane. We thus evaluated the candidate trimer models according to where the glucan chains would be predicted to emerge from them. The fit to the experimental data for the best-fit trimer model (ATCESA1CatD-m1) is shown in Figure 42.

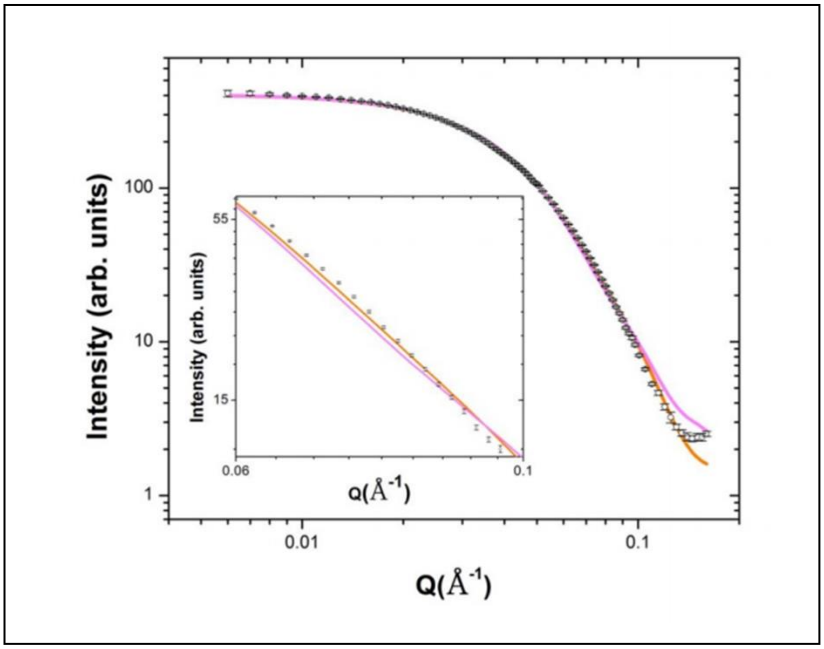


Figure 42: Comparison of theoretical scattering curves of the ROSETTA models with experimental ATCESA1CaD SAXS profile. The theoretical scattering profiles for the model, ATCESA1CatD-m1 and the model, ATCESA1CatD-m12 are shown in orange and magenta respectively. The ATCESA1CaD SAXS curve is shown as open circles. Inset shows the magnified fit for Q in the range of 0.06-0.1. www.plantphysiol.org. "Copyright American society of Plant Biologists" [1]

Superimposition of this model with the GASBOR ab initio model indicates that the overall size and shape of the computational and experimental structure are similar (Figure 43). In this trimer model, the CSR regions are at the interfaces of the monomers and the PCR regions project outward into the cytoplasm (Figure 44A, B). Each monomer-monomer interface has approximately 60 amino acids and a total of 964 Å2 of buried surface area, which is a reasonable value based on previously reported studies analyzing interfaces in protein complexes [132, 133]. A similar arrangement of the CSR regions was reported for the OSCESA8catD dimer [134].

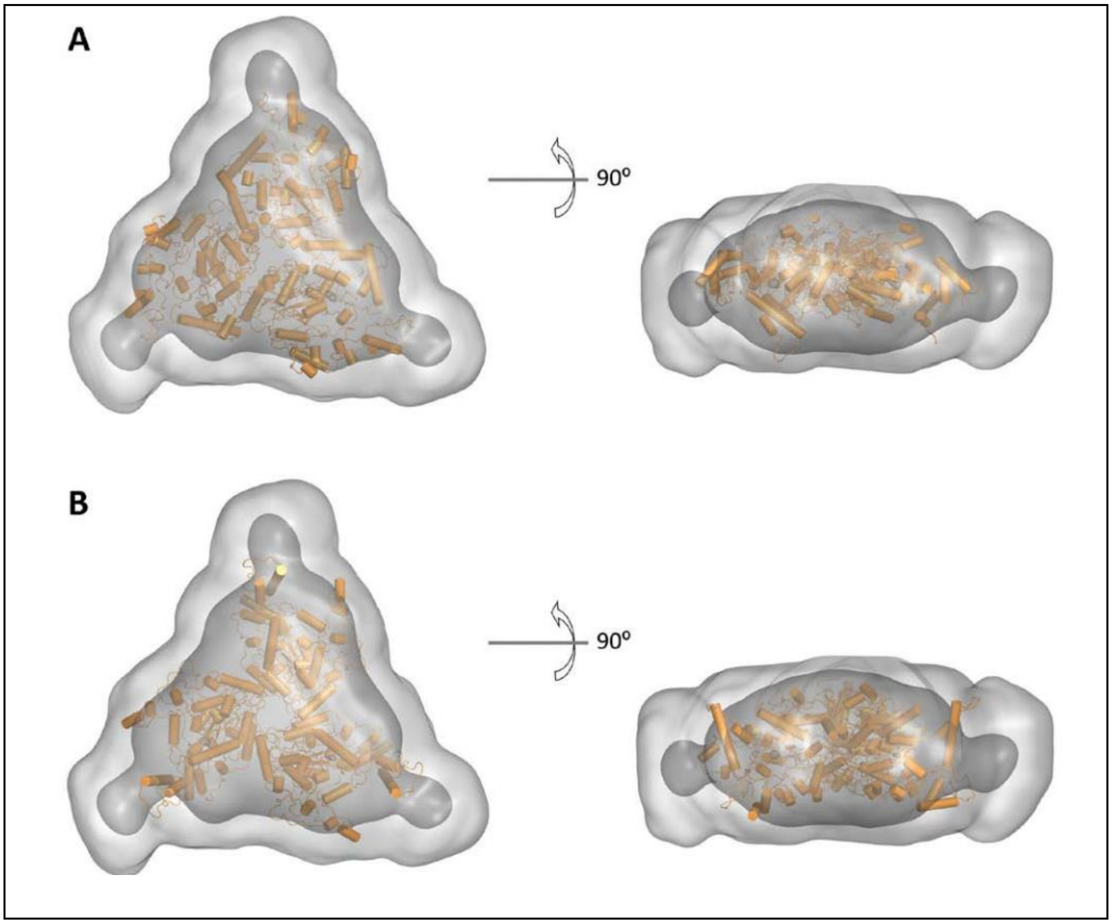


Figure 43: Ab initio models of ATCESA1CatD trimers with ROSETTA models. The light and dark gray surface models represent an averaged and a filtered ab initio models, respectively. (A) ROSETTA model, ATCESA1CatD-m1 superposed with the ab initio model, on right is rotated by 90°. (B) ROSETTA model, ATCESA1CatD-m12 superposed with the ab initio model, on right is rotated by 90°. Three subunits in the trimer are represented cartoon models in orange. www.plantphysiol.org. "Copyright American society of Plant Biologists" [1]

However, structural alignment with BcsA shows that the catalytic cores are projected radially inward in ATCESA1CatD-m1 such that the emergent glucan chains are at an acute angle to the membrane (Figure 44C). This orientation is not optimal for translocation of the cellulose chains across the plasma membrane. Analysis of most other trimer models resulted in a similar outcome, having the emergent cellulose chains from the catalytic domain projecting away from the membrane. However, notable exceptions are models ATCESA1CatD-m12, m-13, m-15 and m-28 for which χ scores resulting from the SAXS curve fitting are between 6.33 and 9.02. The fit of the theoretical SAXS curve for one of these models, ATCESA1CatD-m12 (χ = 6.33), is shown in Figure 42 and the trimer model is shown in Figure 44. Structural alignment with BcsA shows that the catalytic residues are oriented such that the emergent glucan chains would be near to each

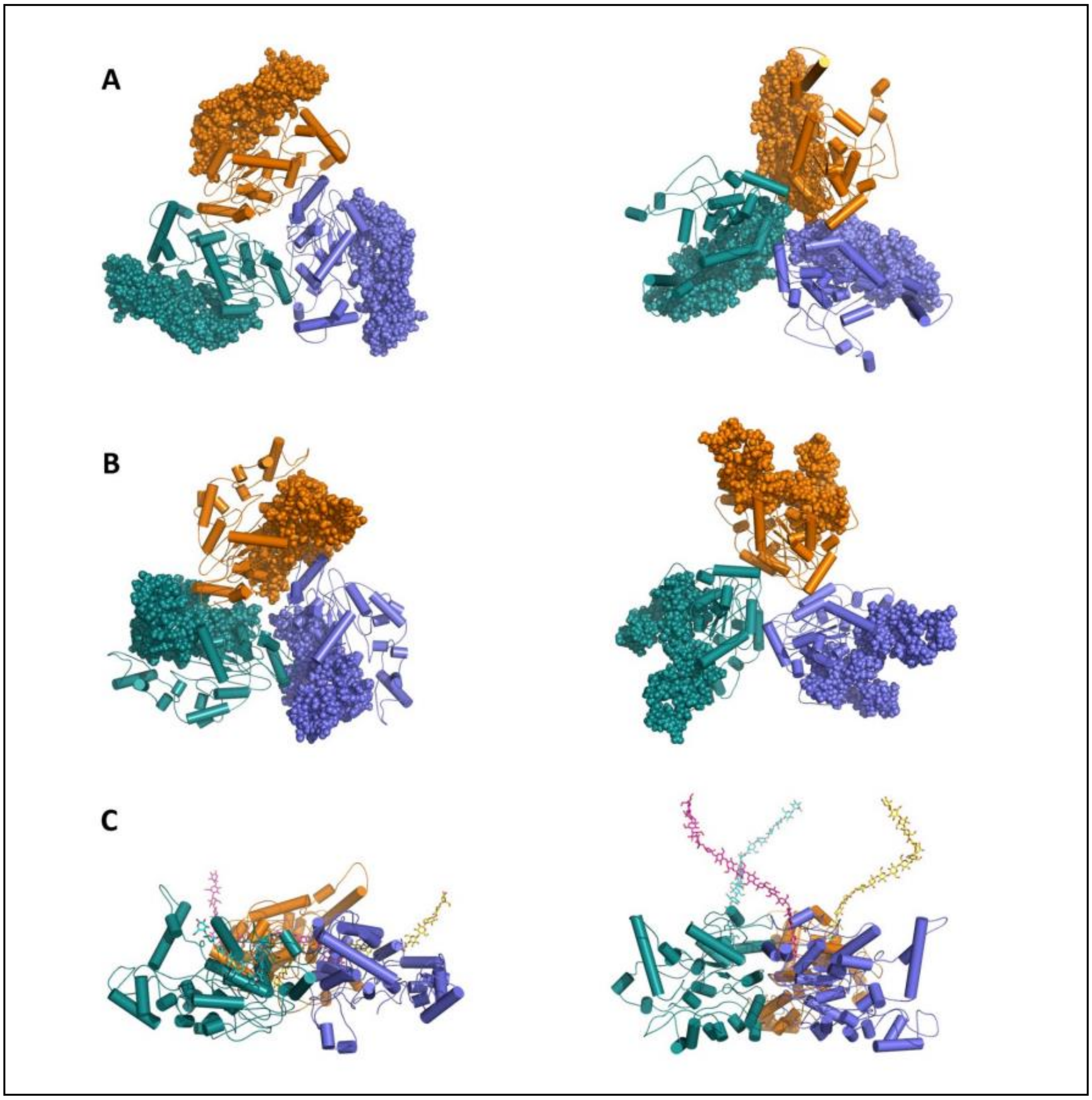


Figure 44: The ROSSETTA models of ATCESA1CatD trimers. Left and right panels represent ATCESA1CatD-m1 and ATCESA1CatD-m12 respectively. (A) P-CR regions are highlighted as spheres. (B) CSR regions are highlighted as spheres (C) The models are rotated 90° to provide side view of the emergent glucan chains based on structural superposition of the bacterial cellulose synthase (pdb code, 4hg6). www.plantphysiol.org "Copyright American Society of Platn Biologists" [1]

other and directed toward the membrane (Figure 44C). In this trimer model, the highly conserved P-CR regions form the base of the catalytic trimer pointing towards the cytosol and the CSR regions project radially outward and do not participate in any interfaces within the trimer (Figure 44A, B). Each monomer-monomer interface has approximately 32 amino acid residues and a total of 750 Å2 of buried surface area. This value is lower than that obtained for ATCESA1CatD-m1, but is a reasonable value for a stable protein-protein interface. The majority of the interfacial residues (26 residues) are from the highly conserved portions of the GT domain between the PCR and CSR regions. Superimposition of the ATCESA1CatD-m12 with the GASBOR ab initio model indicates that both structures overlay well, as was observed for the ATCESA1CatD-m1structure (Figure 43).

Catalytic domain discussion

To gain insight into arrangement of ATCESA1CatD monomers in the trimer, we constructed a series of computational ATCESA1CatD trimers from a homology model of the ATCESA1catD monomer. By examining the fit between 1000 symmetric trimers and the scattering data, it became clear that a very small subset of low energy models could be identified, but that it was not possible to rule out a particular arrangement of monomers based on the fit to the SAXS data alone. The close agreement at low Q and discrepancies in the mid-Q and high Q region revealed that these models largely differed in how the monomers were juxtaposed in the trimer geometry. We propose that it is possible to further select from among these low energy models by considering a priori structural information. In a companion paper to this one, Nixon et al. demonstrate that the CESA TMH region could be modeled in a relatively tight homomeric trimer. This TMH trimer showed good geometric correspondence with individual lobes of the rosette CSC and, when replicated six times, with the 6-lobed view of the

CSC where the TMH cross the plasma membrane. The predicted tight clustering of the TMH regions suggests that in each lobe up to three β-1,4-glucan chains traverse the membrane in close rather than distant proximity, which is reasonable given the need for cellulose chains to interact without folding during cellulose microfibril formation. Using this information and the alignment of our computational models with the atomic structure of *R. sphaeroides* cellulose synthase, inclusive of a glucan chain passing from catalytic core through its TMH region, we are able to suggest that four of the candidate models for ATCESACatD trimers were most likely to be correct. Of these, we chose ATCESACatD-m12 as the most reasonable model for cytosolic domain trimer based on its fit to the experimental data.

Transmembrane Domain of Cellulose Synthase

Multiple plant models have been proposed to explain the detailed molecular events of cellulose biosynthesis[128, 134]. Both of these models have been based on the comparison with the bacterial version of CesA (BcsA) that has been crystallized [130]. Although we have evidence for oligermization into trimer structures from the cystolic domain, we do not fully understand how the plant CesA subunits oligermize to form rosettes[1]. The transmembrane domain of CesA is made of 8 trans-membrane alpha helices (Fig 45).

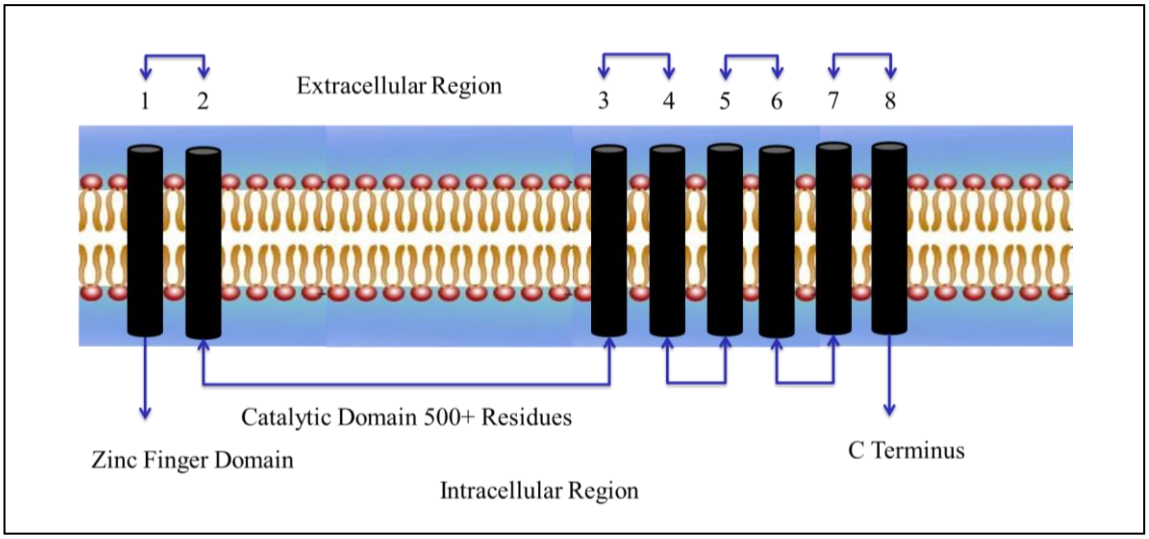


Figure 45: Transmembrane domain topology. The transmembrane domain of atCESA1 consists of 8 transmembrane helices and 300 amino acids. The N-terminal zinc finger domain proceeds TM1. The catalytic domain is in the intracellular region between TM2 and TM3.

From this arrangement the relative orientation of transmembrane helices can be inferred (Table 19). We hypothesize that BCL::Fold is capable of sampling the topology space for the transmembrane domain of CesA. We further expect that limited experimental data from SAXS/SANS and other studies will enable us to select the correct topology model for this domain. Our model would then be integrated into a holistic structural model of CesA for further verification.

Table 19: Inferred orientation of TM helices. Side represents either the n-terminal (N) or the c-terminal (C) end of a given secondary structure element. Orientation represents either the intracellular side of the lipid bilayer (I) or the extracellular side of the lipid bilayer (E).

Helix	Side	Orientation	Helix	Side	Orientation
TM1	N	I	TM5	N	I
TM1	C	E	TM5	C	E
TM2	N	E	TM6	N	E
TM2	C	I	TM6	C	I
TM3	N	I	TM7	N	I
TM3	C	E	TM7	C	E
TM4	N	E	TM8	N	E
TM4	C	I	TM8	C	I

Secondary Structure prediction of the TM regions of atCESA3

To begin the prediction we obtained the FASTA sequence of atCESA3 (Q941L0.fasta). Using the secondary structure prediction methods of PSIPRED[99], MASP, Jufo9d[101], Octopus[104, 105] and ProfPHD[97], we built a secondary structure pool of the eight transmembrane helices based of the consensus predictions of these methods.

Table 20: Consensus transmembrane secondary structure pool for atCESA3. Start is the sequence location of the first residue of the transmembrane helix. End is the sequence location of the last residue of the transmembrane helix. Length is the number of residues in the helix.

Helix	Start	End	Length	Helix	Start	End	Length
TM1	259	277	19	TM5	914	926	13
TM2	287	307	21	TM6	966	981	16
TM3	830	863	34	TM7	996	1014	19
TM4	872	893	22	TM8	1028	1042	13

Generate de Novo models of the transmembrane region of atCESA3

Using cyclic C3 and C6 trimer symmetry, BCL::Fold was used to generate 10,000 different configurations of the eight transmembrane helices in both symmetries. The SSE predictions were obtained through MASP, OCTOPUS, and JUFO9D. The score weights and stage files were obtained from the protocols previously published[55]. The BCL::Fold software suite was compiled for Titan at Oak Ridge National Labs. Folding was performed on Titan

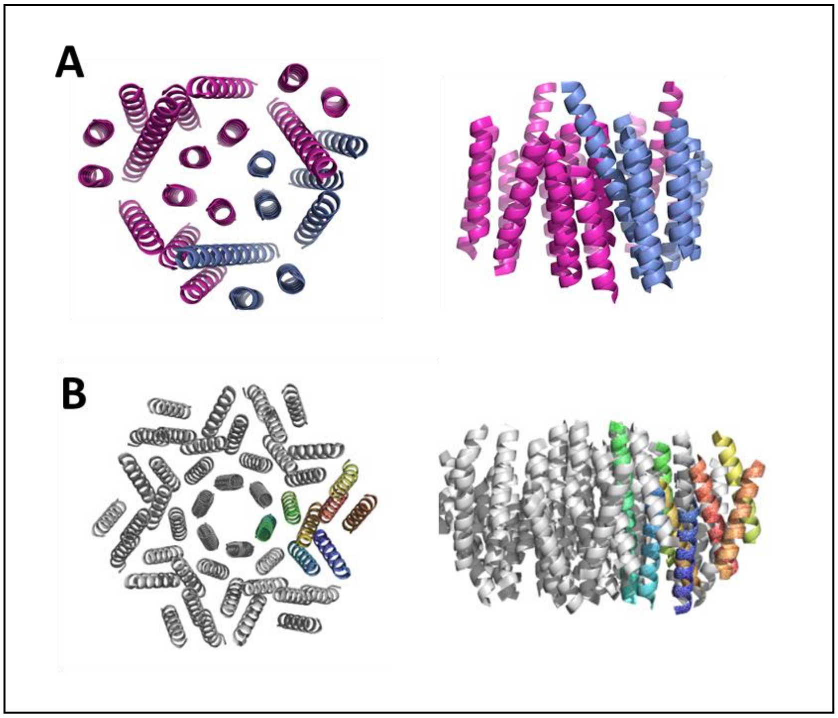


Figure 46: Representative Example of C3 (A) and C6 (B) symmetries of 8 transmembrane helices using BCL::Fold on Titan

Align the TM helices from atCESA3 with the RS_BcsA bacterial analog

Because there is a crystal structure of the bacterial version of CESA, we wanted to thread the coordinates of the TM helices from the bacterial version of RS_BcsA to their analog on the atCESA3 transmembrane helices. To perform threading, we first must align the TM helices between atCESA3 and RS_BcsA. The BCL was used to perform these alignments. (See Table 21)

Table 21: Transmembrane alignments of atCESA3 with bacterial counterpart RS_BcsA. An alignment was performed over the entire sequence with a gap extension penalty of -0.1 and an open gap penalty of 10 (left). Each individual TM helix from atCESA3 was aligned to the sequence of RS_BcsA. (center) Proposed alignment based on the inferred topology of atCESA3 and the known topology of RS_BcsA.

Entire alignment		Single Helix Alignment		Proposed Alignment	
atCESA3 TM	RS_BcsA TM	atCESA3 TM	RS_BcsA TM	atCESA3 TM	RS_BcsA TM
1	3	1	3	1	3
2	4	2	8	2	4
3	5	3	5	3	5
4	6	4	5	4	6
5	-	5	-	5	-
6	7	6	7	6	-
7	8	7	3	7	7
8	-	8	8	8	8

After the alignments were performed, we visually inspected the alignments and alpha helices to ensure the ends of the helices were correctly place on the extracellular or intracellular side of the lipid bilayer. Our proposed alignment is shown in table 21 (right) and Figure 47

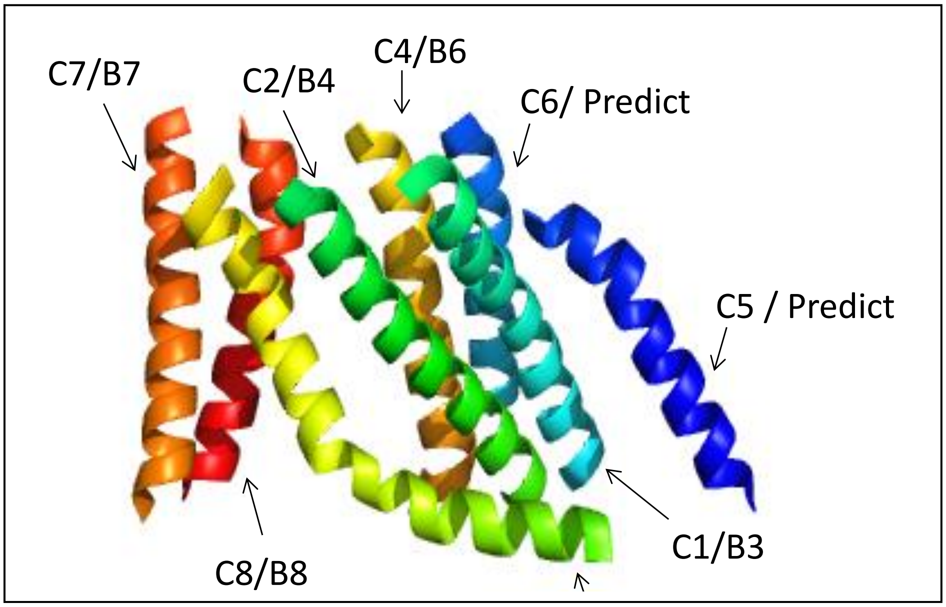


Figure 47: Correspondence between TM helices of atCESA3 (C) and TM helices of RS_BcsA (B)

Transmembrane domain discussion

Our experimental collaborators at Oak Ridge National Laboratories were unable to express and purify the transmembrane region of atCESA3. Without experimental SAS data they did not want to pursue modeling the transmembrane domain. Rather, we focused our attention on the previously described catalytic domain and n-terminal zinc finger domain.

Once experimental data is obtained, the models should be filtered by their agreement with experimental data, leaving a small subset for further analysis. The loops and side chains should be added back to the model to create a complete protein model. Because the cystolic catalytic domain is very large, it should be replaced by a small loop connecting TM2 with TM3. Completed models could then be explored using molecular dynamics to determine stability in both the lipid bilayer and solvent outside the bilayer.

CHAPTER VII

Conclusion

To understand this field, I wrote a review article that was praised by leading scientists in the field as: "Excellent review on SAXs, this should be required reading material for anyone wanting to learn SAXS". This review equipped me with understanding of how to reconstruct SAXS profiles from atomic coordinates. Importantly it provided an understanding of how to use the Debye implementation. In this innovative 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 and Experimental Data. 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 experimental scattering profiles of 13 proteins were used as a restraint in BCL::Fold.

Because of the low resolution of the SAXS / SANS, they cannot be used exclusively to identify the native protein configuration from a set of similar protein configurations. We have shown however through our work with the cytosolic region of cellulose synthase, that this type of data 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 generated SAXS profiles from complete protein models, it was apparent that the shape of the SAXS profile is important when comparing two structures. To account for this behavior, we computed the derivative of the profiles and then computed the χ similarity score 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 88%. 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. 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."

Computation of SAXS profiles can be used to validate high-resolution models in solution and to identify biologically active protein conformations. This was used extensively in my work on the n-terminal and cytosolic domains of Cellulose Synthase 1 and 3 in *Arabidopsis Thaliana.* SAXS was used to characterize trimer complexes whose components have known monomeric structure. 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 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 using ensemble optimization methods.

Future Work

My work to incorporate small angle X-Ray and Neutron scattering into the BCL::Fold suite provides a strong foundation for further optimization and expansion in this field. We can improve the speed of profile reconstruction, the SAS method for protein folding, loop modeling, ensemble modeling, and public use. I will discuss each of these directions.

Profile Reconstruction

The initial benchmarks of BCL::SAS were performed using SAS profiles computed from CRYSOL / CRYSON in lieu of experimental profiles obtained in the lab. The simulated experimental profiles were instrumental in validating my early attempts at SAXS profile reconstruction. Once I obtained experimental data obtained in the lab, I had to carefully consider the excluded volume parameter (C_1) and the hydration shell parameter (C_2). The only way I could replicate the results of CRYSOL and FOXS was to optimize these parameters by adjusting them and repeatedly computing the χ similarity score.

The excluded volume and hydration shell parameters were optimized by minimization algorithms that do not use the derivative. To compute the SAXS profile of a model and fit the model with experimental data, my algorithm requires 410 evaluations to find the optimal C_1, C_2 combination that minimizes the χ agreement. Future work should employ optimization routines that use the derivative of the minimization function to arrive at the optimal combination of these parameters more rapidly. Specifically I recommend a Levenberg-Marquardt optimization method.

Folding with BCL::Fold

In order to achieve the optimal fit between experimental SAXS profiles and SAXS profiles computed from a model, the excluded volume and hydration shell parameters must be optimized. This optimization requires 410 χ evaluations and the solvent accessible surface area (SASA) value for each model. This computational demand is not feasible during protein folding. Furthermore, I have shown that I cannot match the SANS profile without proper C_1 and C_2 optimization.

To address these limitations, I propose an innovative approach to use SAS during protein folding simulations. I suggest that we explore the fit between experimental and computed pair wise distance distribution functions. The P(r) fit should be benchmarked against a protein set to determine if 1) It can be used exclusively during folding with BCL::Fold or 2) Used in combination with the previously published SAXS score during folding with BCL::Fold.

The P(r) function is computed from the experimental SAXS profile I(q) through an indirect Fourier transformation. Theoretically the information content of both functions is identical. This transformation is routinely computed using the software GNOM [117] from the ATSAS suite. The advantages of using this form of the data are 1) SAXS and SANS input data can be treated identically, 2) Profile reconstruction is not necessary, 3) GPU acceleration is not necessary, 4) C_1 and C_2 optimization is not necessary, 5) SASA computation is not necessary, 6) The P(R) can be computed directly from BCL Models by using actual Euclidean distances.

In my view, the benefits of using the P(r) function during folding merit a study to determine the loss in accuracy due to the transformation of the I(q) into the P(r) via GNOM. A scoring metric must be developed to compare the experimental P(r) function with the P(r) function computed from BCL::Fold models. These functions should be normalized, with an area under the curve set to 1. The comparison metric should identify important features of the curve such as 1) Smoothness, 2) D_{max} cutoff, 3) Area under the curve cutoff, 4) Area under the curve similarity, 5) morphology of curves.

Loop modeling improvement

In this work, I implemented a rapid method to approximate the residues of loop regions between SSEs. I demonstrated that modeling these loop regions improves the SAXS χ agreement score with experimental data. This loop modeling process can be vastly improved. I recommend a loop hash method. The idea is to create a repository of loops from the protein data bank of a given sequence length, angle between SSEs, and Euclidean distance between SSEs. Once a model is produced with BCL::Fold, the loops would be would be built from the database of existing loops that are keyed by the sequence length, angle between SSEs, and

Euclidean distance. This would enable the production of an ensemble of proteins with different native loop configurations. I expect this approach to improve agreement by SAXS score.

Ensemble modeling

An exciting feature in modern SAXS is identifying and modeling protein flexibility from an ensemble set of different conformers to fit experimental SAXS data. This requires a large library of starting conformers as input to the algorithm. After a suitable library of conformers has been generated or found, the experimental SAXS data are used as a constraint in an algorithm to determine which combination of conformers optimally fit the data. The scattering intensity (I) is represented by a linear combination of the selected conformers. In this process the algorithm must decide 1) Which conformers to use and 2) How many conformers are required to accurately recreate the experimental SAXS profile. I propose that we develop a method to generate conformers of a given protein and then an algorithm to construct SAXS profiles from a weighted linear combination of conformers similar to the way the BCL::Scoring function is setup.

Create online webserver

BCL::SAXS is a tool that is of interest to the scientific community because of the direct use of the Debye formula. I proposed that we explore the P(r) function metric during folding simulations. For the web server I propose that we use C_1 and C_2 optimization and SASA values to fit experimental SAXS data with rigid models. There are no online methods using this approach and would be of interest to the SAXS community.

APPENDIX

APPENDIX I: CHAPTER 2 COMMANDLINES

Loop approximation

```
/hd0/putnamdk/workspace/bcl-testing/build/linux64_release/bin/bcl-apps-
static.exe restraint:SaxsPrep -pdb_file 3hz7_bcl.pdb -output_model -
min_sse_size 5 3 999
```

Visualize model in Pymol

In the pdb file the temperature factor on the atom line is in column 61-66. This value will be 0.0 for approximated loop residues

In pymol use this command:

```
select loops, b < 0.0001
show loops as spheres
```

To alter the size of the spheres:

```
alter loops, vdw=1
rebuild
```

To change transparency of cartoon:

```
set cartoon_transparency, 0.5, 6lyz_bcl
```

Generate clean BCL protein model

```
/hd0/putnamdk/workspace/bcl-testing/build/linux64_release/bin/bcl-apps-
static.exe protein:PDBConvert input.pdb -bcl_pdb -output_prefix input_bcl
```

The bcl file must be adjusted to remove any residues from the sequence lines that are not specified in the atom lines
Once the missing residues have been removed from the pdbfile, rerun PDBConvert to renumber the file.

For multimers, PDB Convert adds a TER line which uses one of the line id slots. MSMS removes the TER and labels the atoms sequentially. This results in an offset in numbering by the number of
TER lines present. Run a script to remove the TER lines and then renumber the atom lines consecutively to resolve this conflict

label the cleaned pdb file: #_????_bcl.pdb

The pdb filename must be put on the first line of a text file called pdbs.ls

example:
original input file: pdb_model.pdb

Step 1: Run pdb convert to generate clean BCL file

```
/hd0/putnamdk/workspace/bcl-testing/build/linux64_release/bin/bcl-apps-
static.exe protein:PDBConvert input.pdb -bcl_pdb -output_prefix
input_bcl
```

Step 2: Run script to identify missing residues on original pdbfile

```
perl
../identify_missing_residues.pl pdb_model.pdb > missing_residues.txt
```

Step 3: Remove identified missing residues by hand (haven't written script yet) from the SEQRES lines of the output in Step 1

Step 4: Run pdb convert on the manipulated file from step 3 to renumber the atoms correctly

```
/hd0/putnamdk/workspace/bcl-testing/build/linux64_release/bin/bcl-apps-static.exe protein:PDBConvert input.pdb -bcl_pdb -output_prefix input_bcl
```

Step 5: For Multimer Processing, the BCL adds TER lines after each chain. Run Script to removed TER lines and renumber atoms

```
perl ../renumber_atom_lines.pl file.pdb ( the pdb file is the output from step

4)
```

Generate MSMS file for solvent accessible surface area

convert pdb files to xyzr files:

```
pdb_to_xyzr *.pdb > *.xyzr
```

run msms on .xyzr file to get the .area file:

```
msms -if *.xyzr -af *.area -probe_radius 1.399
```

Generate SAXS Profile with c1 and c2 optimization and fit on Log10 scale

```
/hd0/putnamdk/workspace/bcl-testing/build/linux64_release/bin/bcl-apps-
static.exe restraint:AnalyzeAgreement -analysis_prefix 6lyz -
analysis_type_enumerated "AnalyzeSas( c1=1, c2=0,
experimental_profile=iofq_data_file.dat, sasa_profile=6lyz_bcl.area,
optimize_hydration_parameters=true, default_search_grid=true,
scoring_function=chi, use_errors=0, cpu=false, sans=false,
approximate_side_chains=false, approximate_loops=false, transformations(
Normalize, Log10, Scale ), print_transformations=true, y_max=1.0)"
```

Loess (locally weighted scatter plot smoothing) data regression in R

Import data set into R

```
data <- read.delim("1_SAXS.dat", header = T, sep="")
```

plot the raw data:

```
p <- qplot(Q, log10(Intensity), data=data))
```

Save an image of the data

```
ggsave(filename="Samp1_a.png", plot=p)
```

Add LOESS Regression line with a span of 0.3. The span controls the jaggedness of the line.

```
p1 <- p + geom_smooth( method = "loess", span = 0.3, size = 1.5))
```

Write the values of the data fit to a variable:

```
fit = predict(loess(log10(Intensity)~Q, data, span=0.2, degree=1), data$Q)
```

Compute and plot the derivative of the fit

```
fx.spline <- splinefun(data$Q, data$fit)
```

```
plot(data$Q, fx.spline(data$Q, deriv=1), type='p')


Write the fit values to the data frame
data = cbind.data.frame( data, fit)

Write the data to a file
write.table( data, file="fit.out", sep =" ", quote=FALSE)
```

Generate SAXS profile with c1 and c2 optimization and fit on derivative scale

```
/hd0/putnamdk/workspace/bcl-testing/build/linux64_release/bin/bcl-apps-
static.exe restraint:AnalyzeAgreement -analysis_prefix 6lyz -
analysis_type_enumerated "AnalyzeSas( c1=1, c2=0,
experimental_profile=iofq_data_file.dat, sasa_profile=6lyz_bcl.area,
optimize_hydration_parameters=true, default_search_grid=true,
scoring_function=chi, use_errors=0, cpu=false, sans=false,
approximate_side_chains=false, approximate_loops=false, transformations(
Normalize, Log10, Scale, Derivative ), print_transformations=true, y_max=1.0)"
```

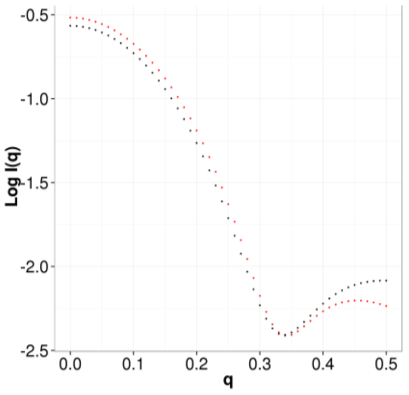
APPENDIX II – CHAPTER 4 SANS BENCHMARK COMMANDLINES

SANS profile reconstruction

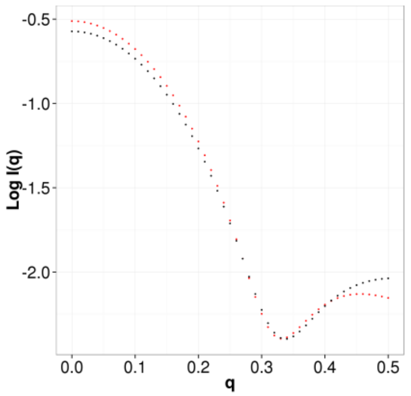
```
/home/putnamdk/Vanderbilt/Sans_Project/bcl/oanh/bcl-apps-static.exe
restraint:AnalyzeAgreement -analysis_prefix 1ENH -analysis_type_enumerated
"AnalyzeSas( c1=1, c2=0,
experimental_profile=/home/putnamdk/Vanderbilt/Sans_Project/cryson/d20_30/1
ENH00.int, sasa_profile=1ENH_bcl.area, optimize_hydration_parameters=true,
default_search_grid=true, scoring_function=chi, use_errors=0, cpu=true,
approximate_side_chains=false, approximate_loops=false, use_sans=true,
deuterium_percent=0.3, transformations( Normalize, Log10, Scale ),
print_transformations=true, y_max=423.937)
```

APPENDIX III – SANS BENCHMARK DATA OF 1ENH, PRE-OPTIMIZATION, CONTINUUM MODEL

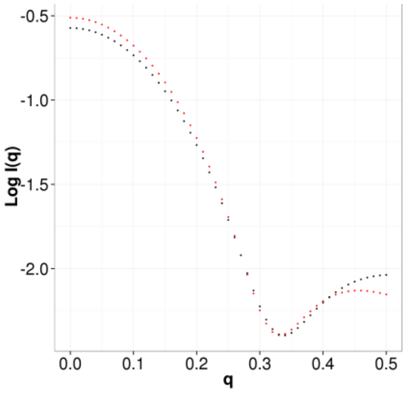
0% D_20



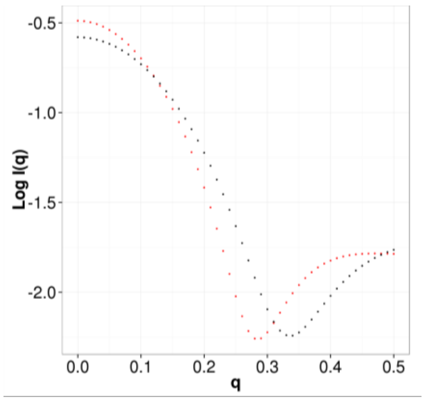
10% D_2O



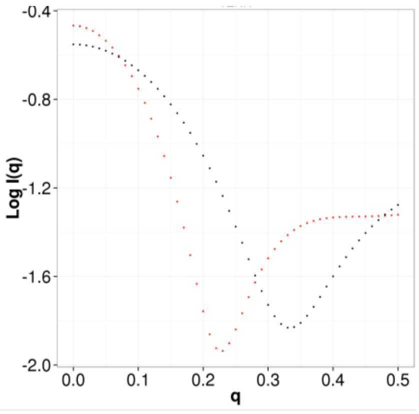
20% D_2O



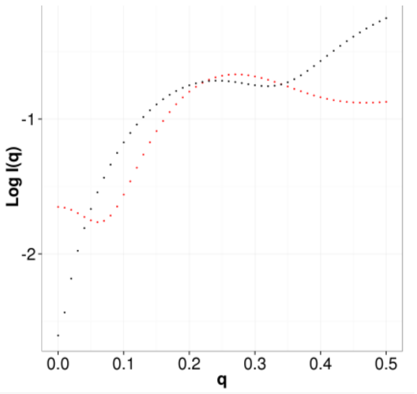
30% D_2O



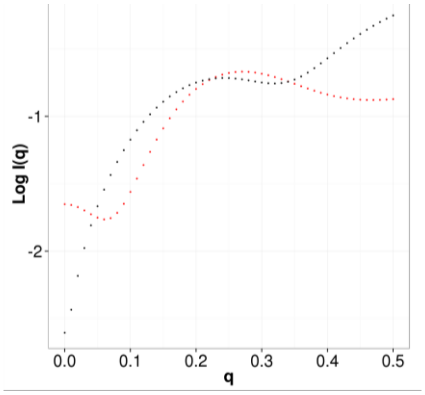
40% D_2O



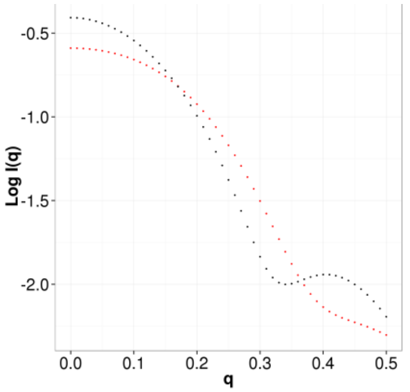
50% D_2O



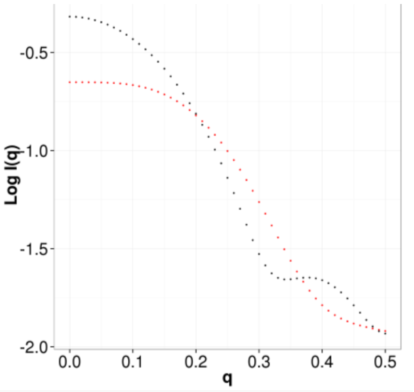
60% D_2O



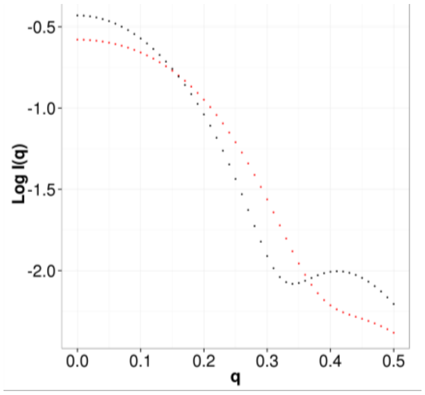
90% D_20



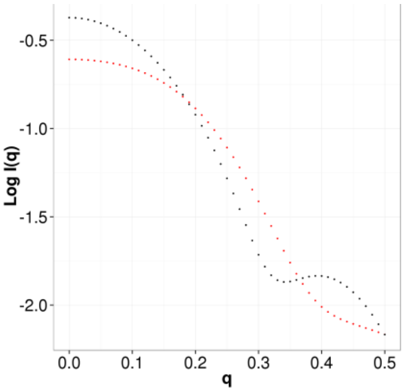
70% D_2O



100% D_20

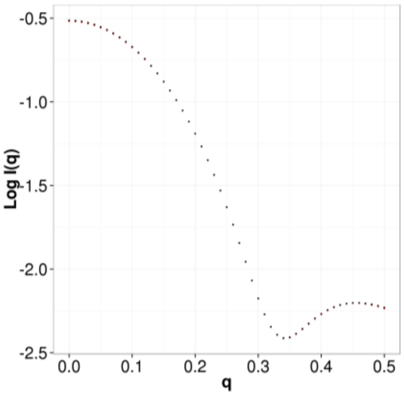


80% D_2O

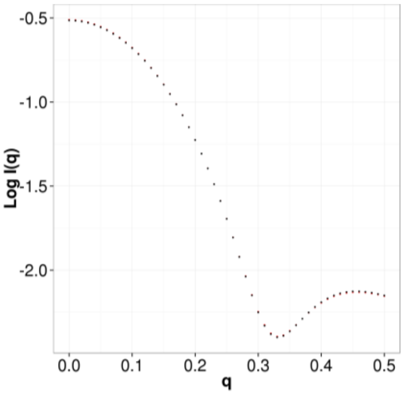


APPENDIX IV – SANS BENCHMARK DATA OF 1ENH, OPTIMIZED, CONTINUUM MODEL

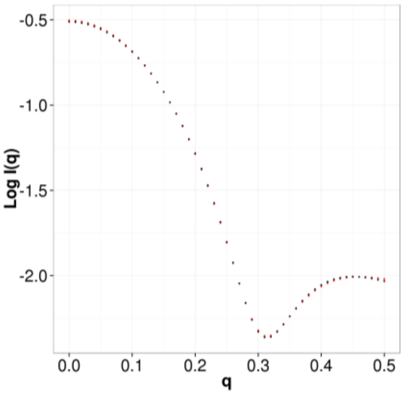
0% D_2O



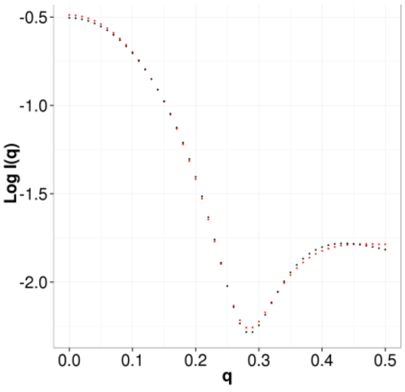
10% D_2O



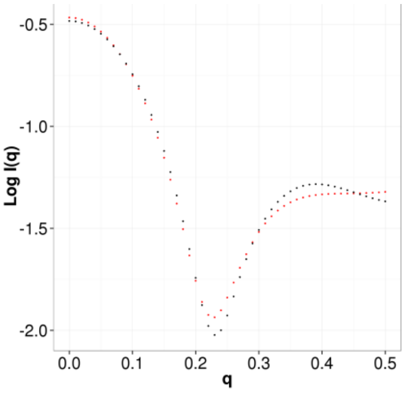
20% D_2O



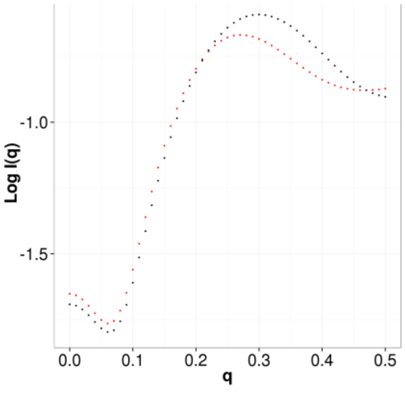
30% D_2O



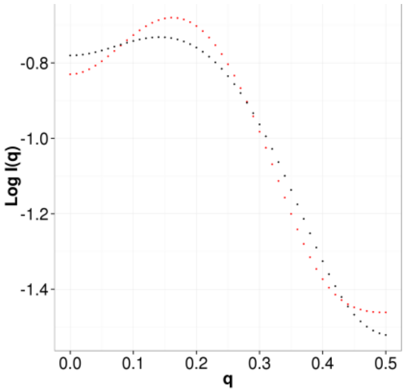
40% D_2O



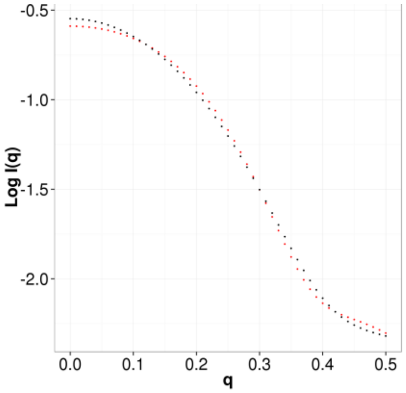
50% D_2O



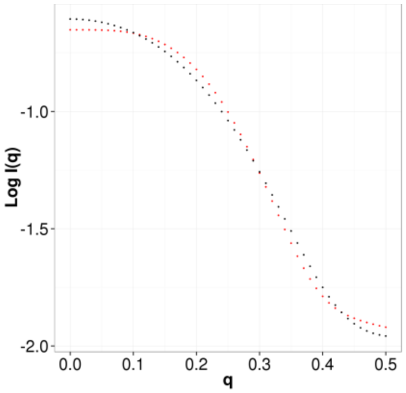
60% D_2O



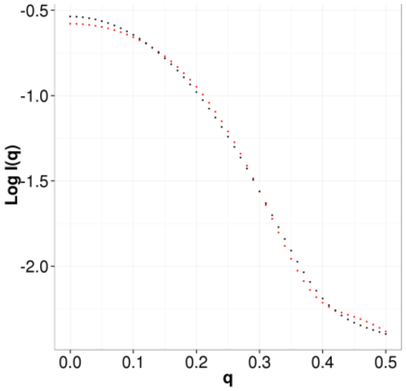
90% D_2O



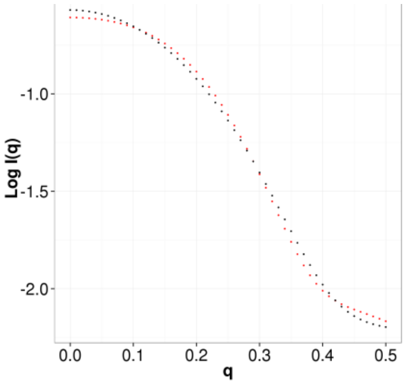
70% D_2O



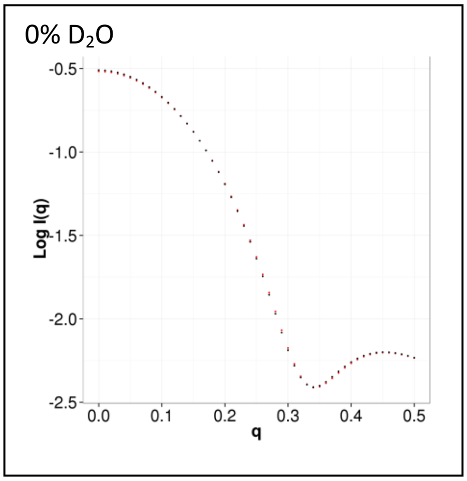
100% D_2O

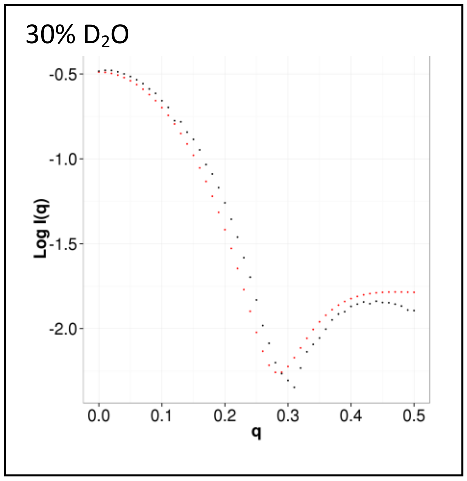


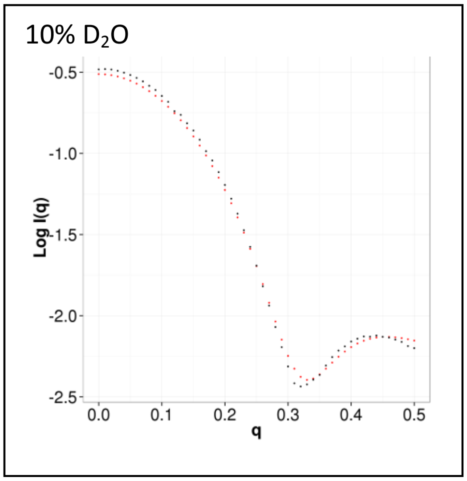
80% D_2O

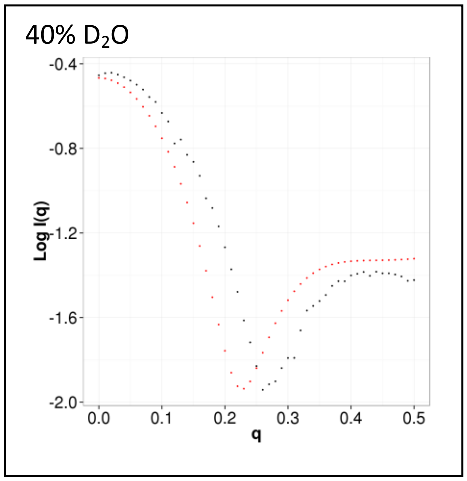


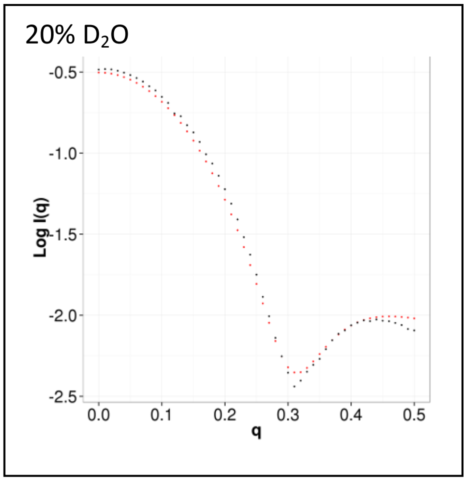
APPENDIX V – SANS BENCHMARK DATA OF 1ENH, PRE-OPTIMIZATION, QUANTIZED MODEL

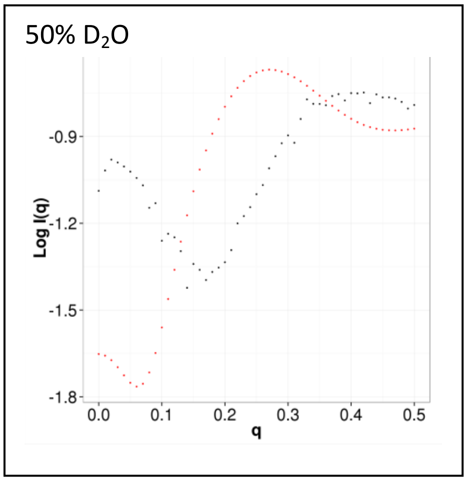

0% D_2O


30% D_2O

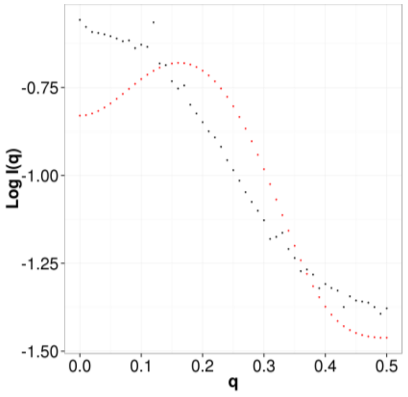

10% D_2O


40% D_2O

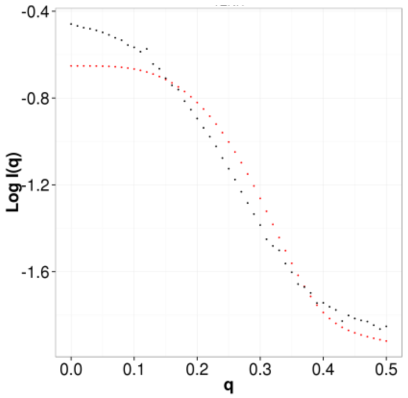

20% D_2O


50% D_2O

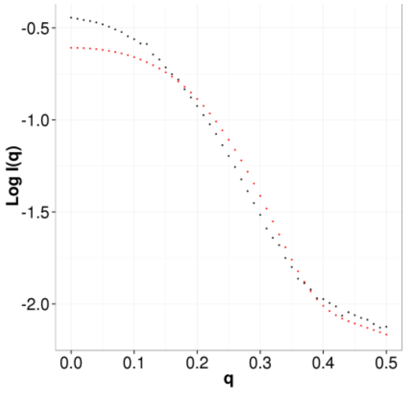
60% D_2O



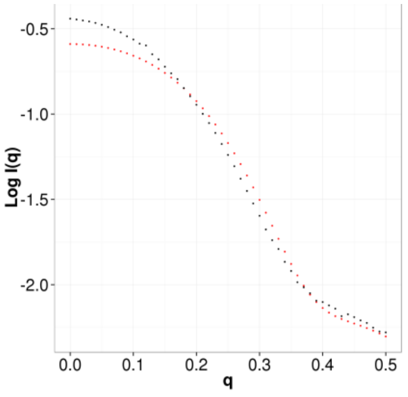
70% D_2O



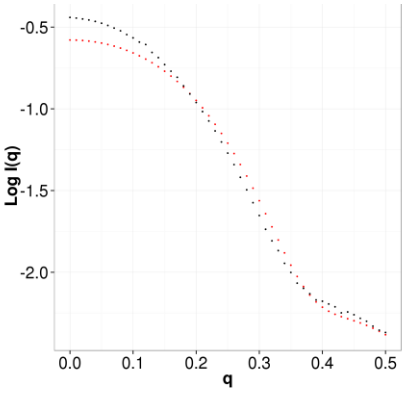
80% D_2O



90% D_2O

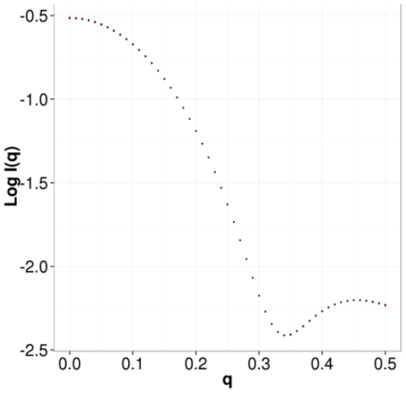


100% D_2O

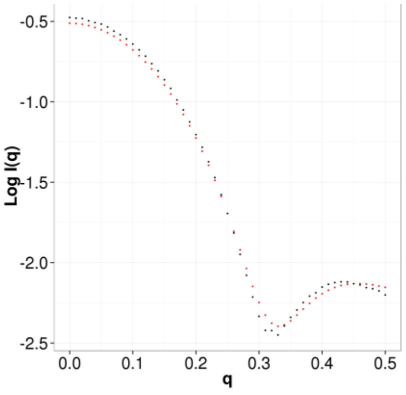


APPENDIX VI – SANS BENCHMARK DATA OF 1ENH, OPTIMIZED, QUANTIZED MODEL

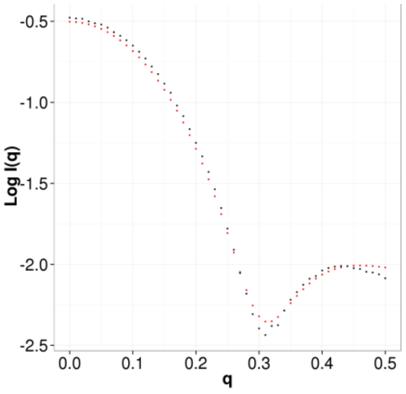
0% D_2O



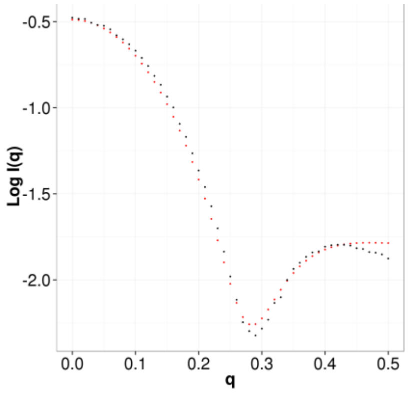
10% D_2O



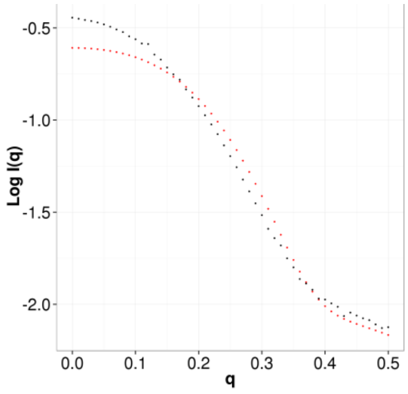
20% D_2O



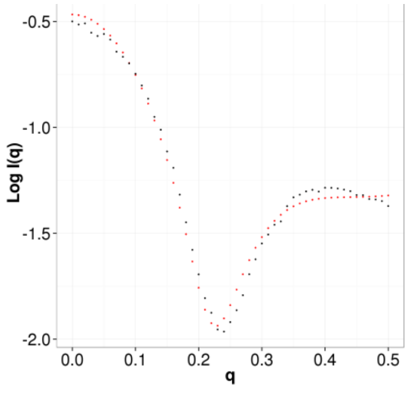
30% D_2O



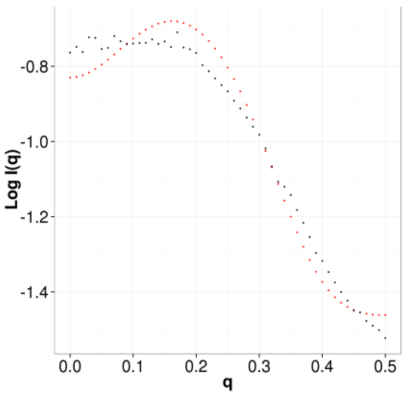
40% D_2O



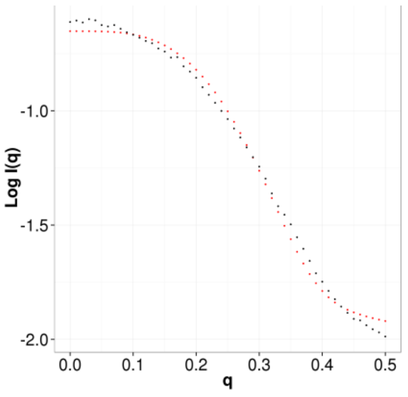
50% D_2O



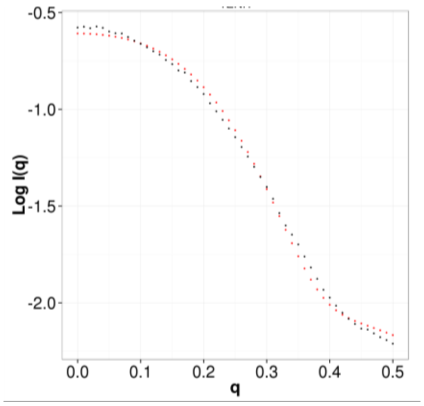
60% D_2O



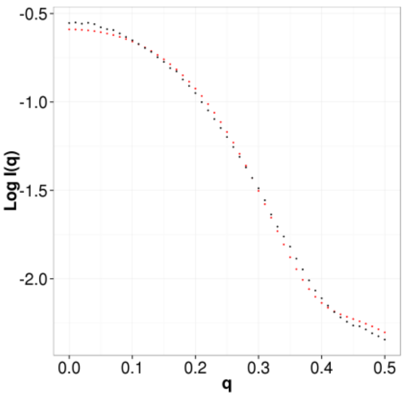
70% D_2O



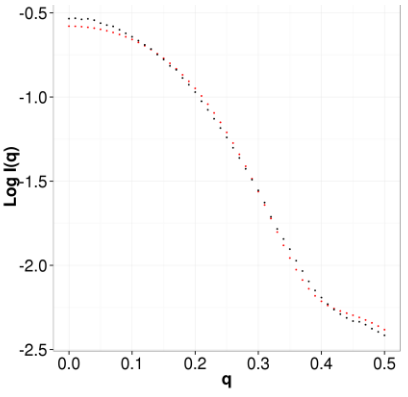
80% D_2O



90% D_2O



100% D_2O



APPENDIX VI – CHAPTER 5 SAXS FOLDING COMMANDLINES

Rosetta loop modeling commands

1. Download FASTA file from PDB page.
2. Create fragment file:
   request a job at: http://robetta.bakerlab.org/fragmentsubmit.jsp => download 3.bin and 9.bin
3. Add a loop definition file: which residues to model,

eg:

```
LOOP   1   4 0  0.0 0
LOOP   92 101 0 0.0 0
```

4.Write an option file into the working directory

The option (.options) file specified the set-up of the loop modeling such as location of the fragment files, fragment sizes, refinement, extension, and relaxation. An example is provided below:

eg:

```
-loops:frag_sizes 9 3 1
-loops:frag_files /basepath/9.bin /basepath/3.bin none
-loops:build_initial true
-loops:remodel quick_ccd
-loops:refine refine_ccd
-loops:extended true
-loops:relax relax

-ex1
-ex2

-out:output true
-out:pdb true
```

5. Rosetta Loop modeling command

Eg:

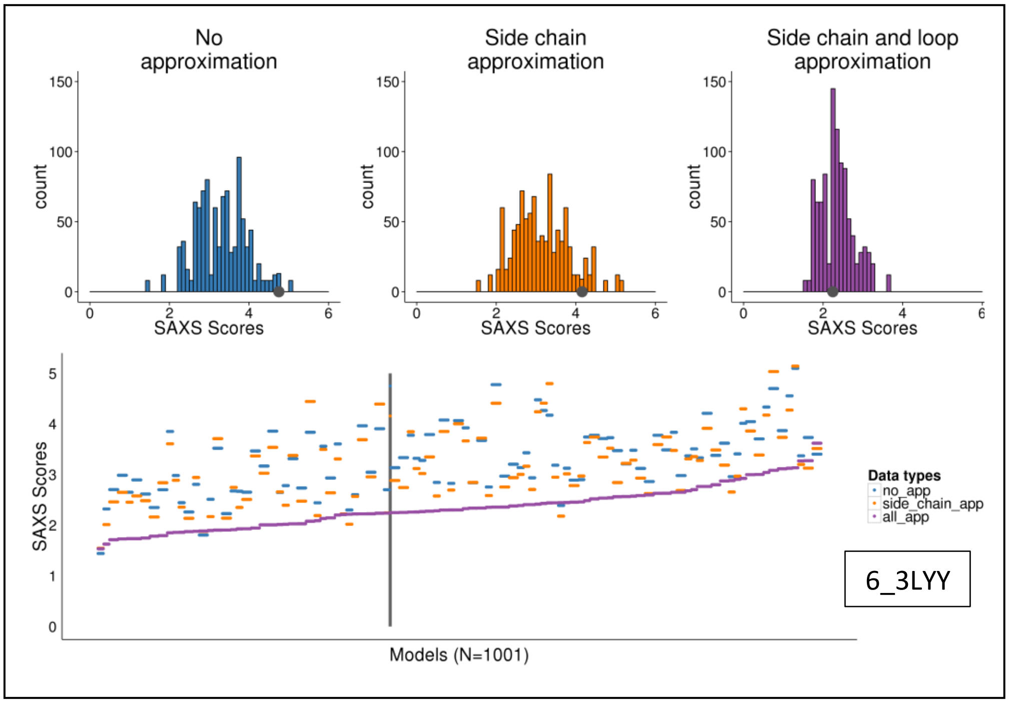
```
/dors/meilerlab/apps/rosetta/rosetta_2015.12.57698/main/source/bin/loopmodel.de
fault.linuxgccrelease @<option file> -nstruct <number of output models> -
loops:loop_file <loop definition file> -s <input crystallographic PDB file> -
out:prefix <prefix of output files> -out:path <directory storing output pdb
files>
```

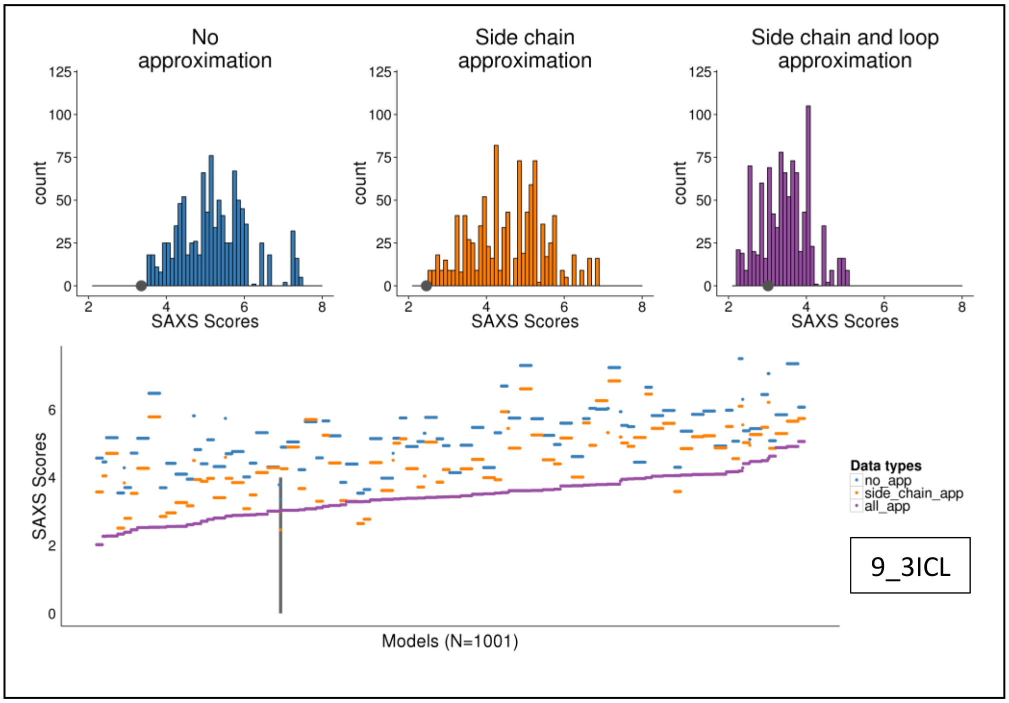
6. Use Rosetta to simulate missing loop regions in crystallographic structures

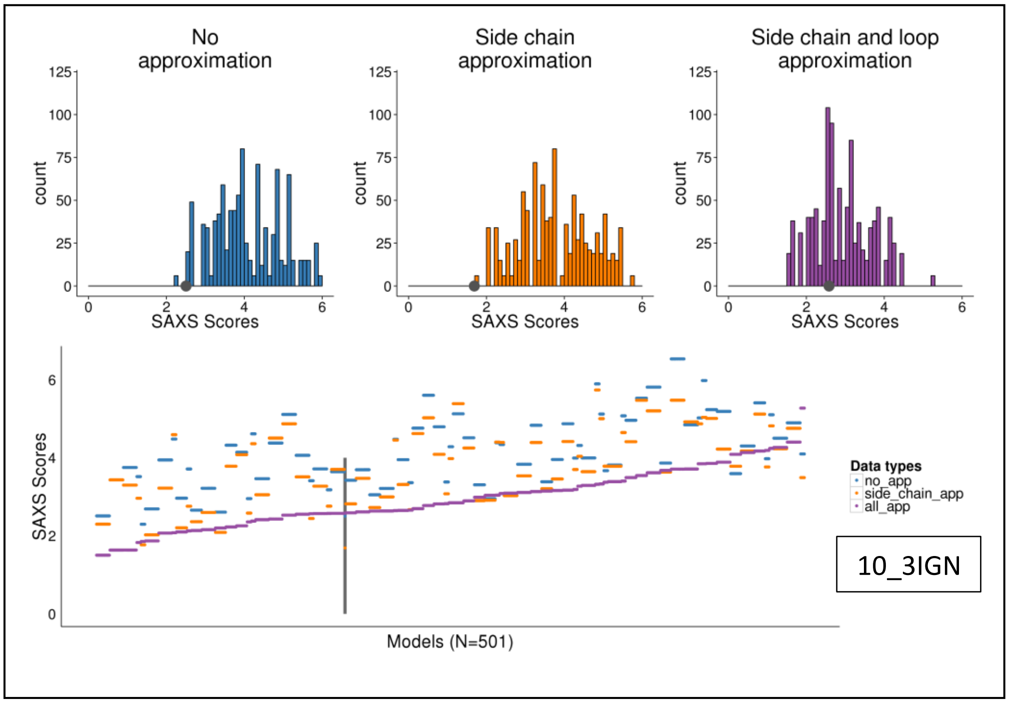
Eg:

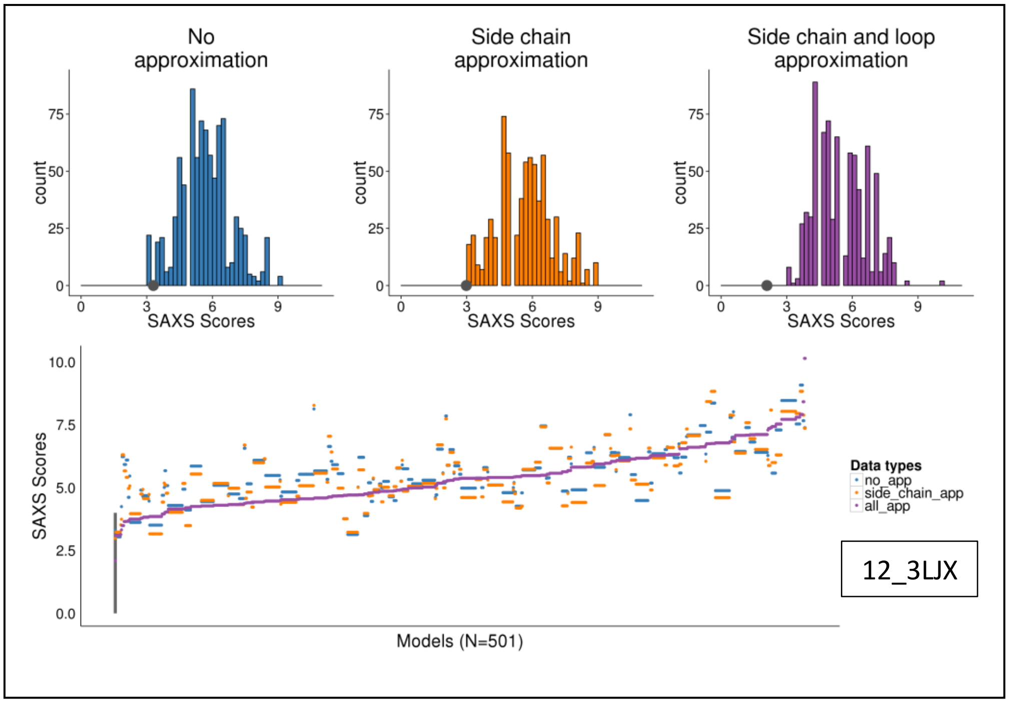
```
/dors/meilerlab/apps/rosetta/rosetta_2014.35.57232/main/source/bin/relax.defaul
t.linuxgccrelease1 -relax:constrain_relax_to_native_coords -in:file:native <PDB
file> -relax:coord_constrain_sidechains -relax:ramp_constraints false -s <PDB
file> -out:prefix <prefix> -nstruct 500
```

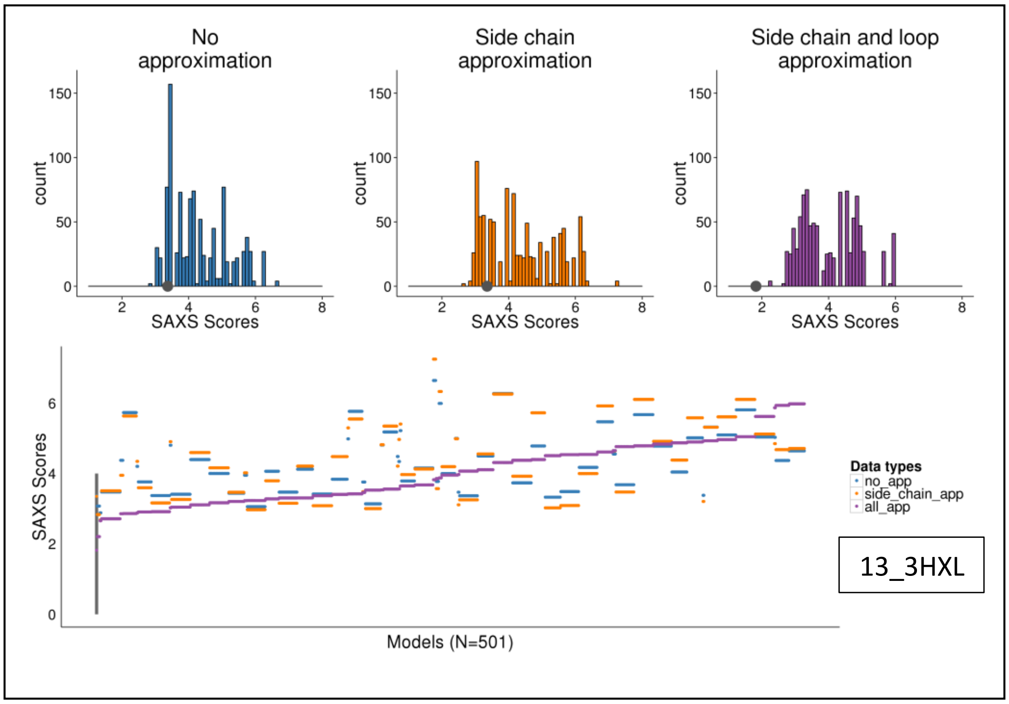
APPENDIX VIII – SAXS SCORE DISTRIBUTION

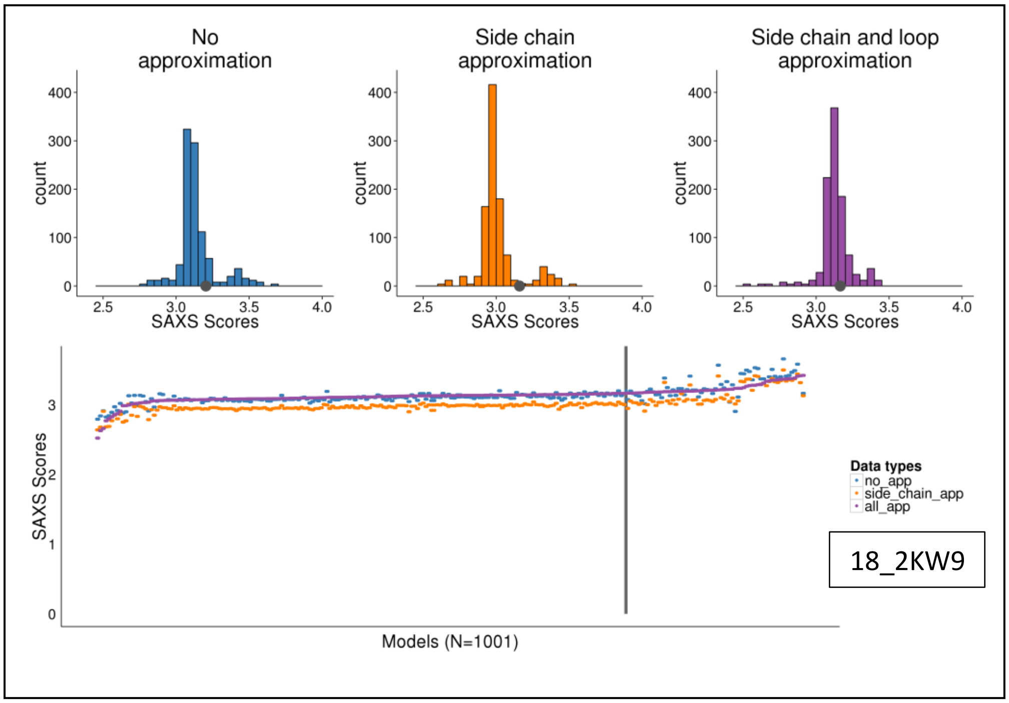


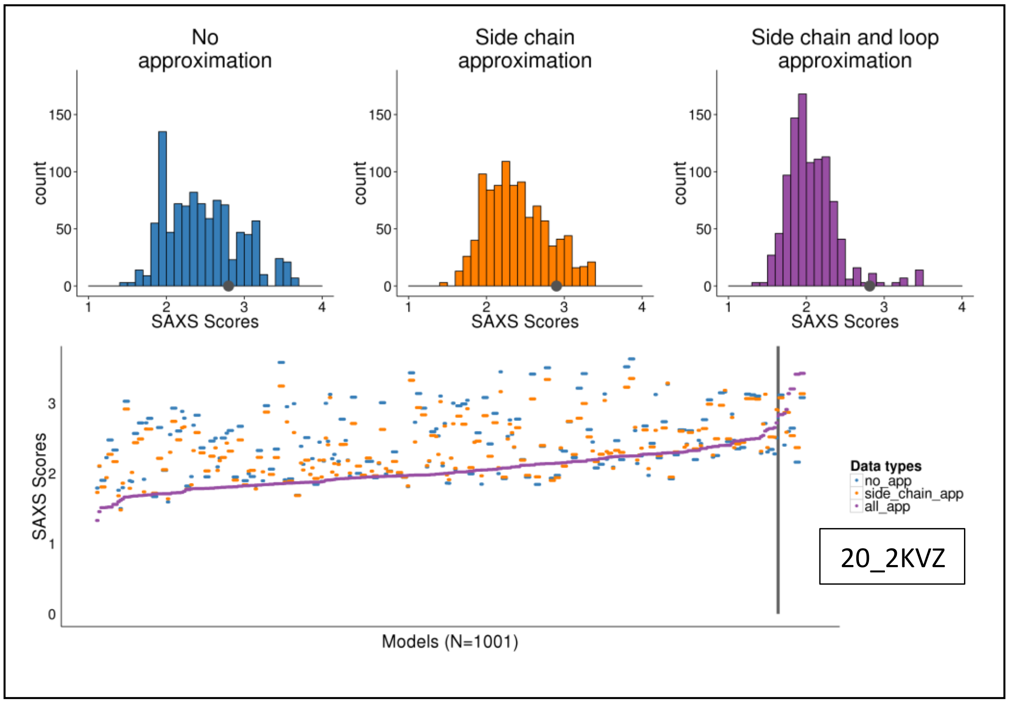


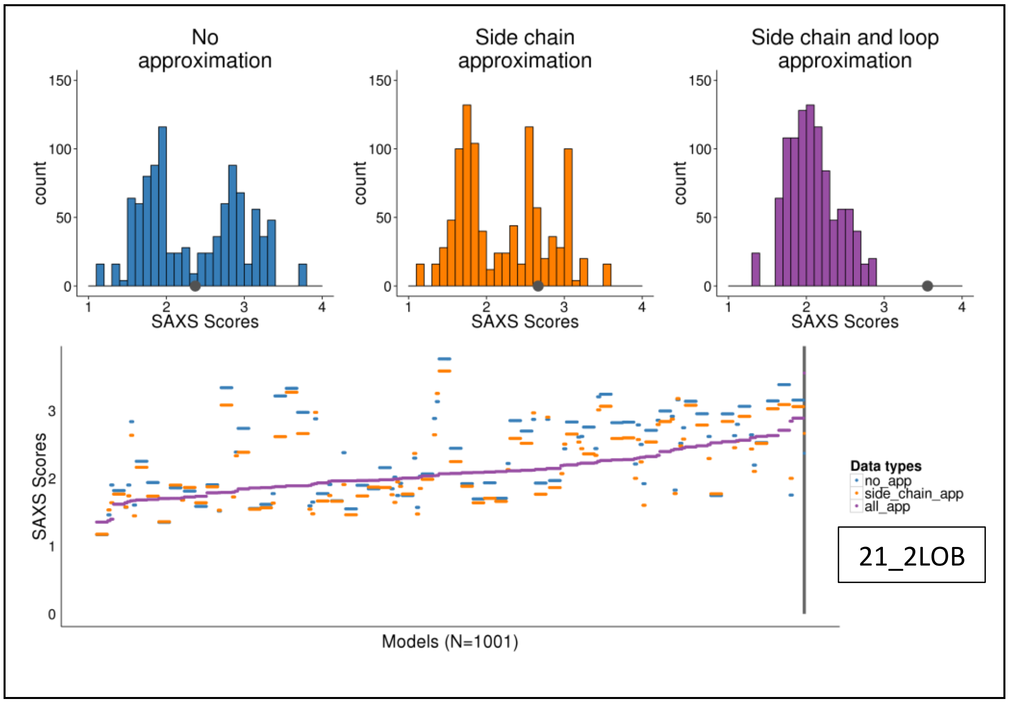

No approximation
Side chain approximation
Side chain and loop approximation
count
125
100
75
50
25
0
0
2
4
6
SAXS Scores
SAXS Scores
6
4
2
0
Models (N=501)
Data types
no_app
side_chain_app
all_app
10_3IGN

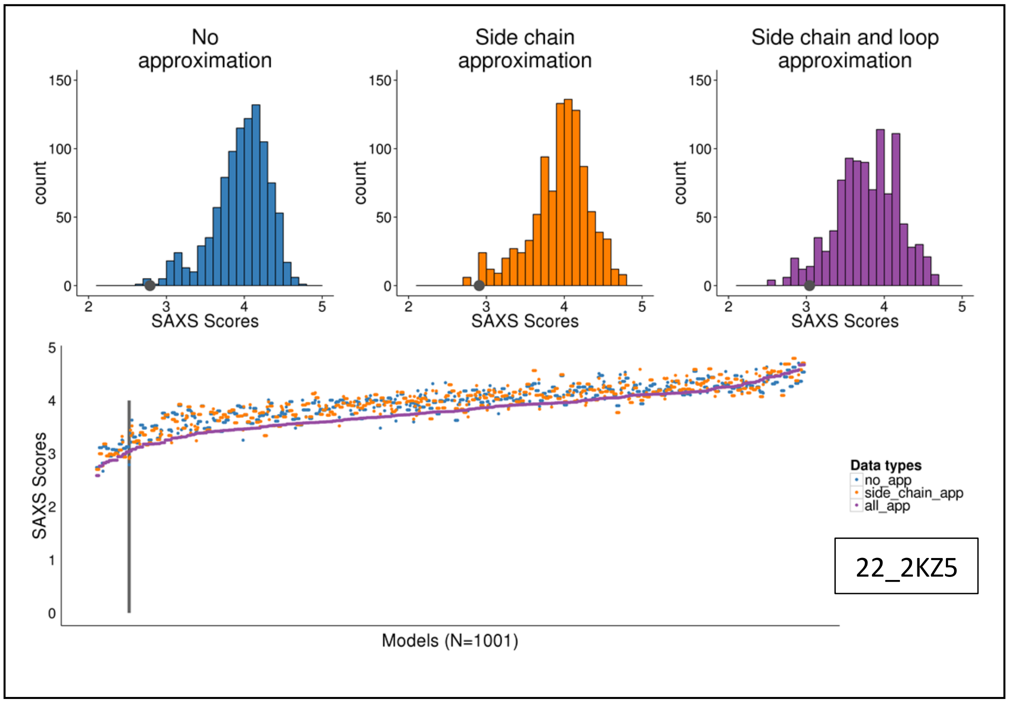

No approximation
Side chain approximation
Side chain and loop approximation
count
75
50
25
0
0
3
6
9
SAXS Scores
SAXS Scores
10.0
7.5
5.0
2.5
0.0
Models (N=501)
Data types
no_app
side_chain_app
all_app
12_3LJX

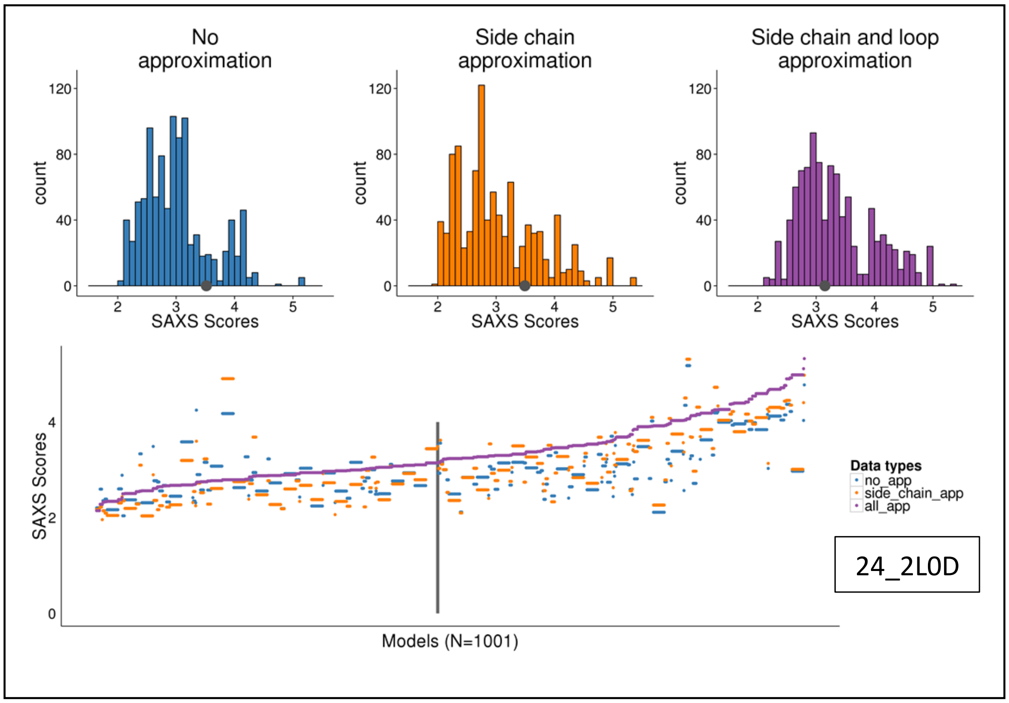

No approximation
Side chain approximation
Side chain and loop approximation
count
150
100
50
0
2
4
6
8
SAXS Scores
SAXS Scores
0
2
4
6
Models (N=501)
Data types
no_app
side_chain_app
all_app
13_3HXL

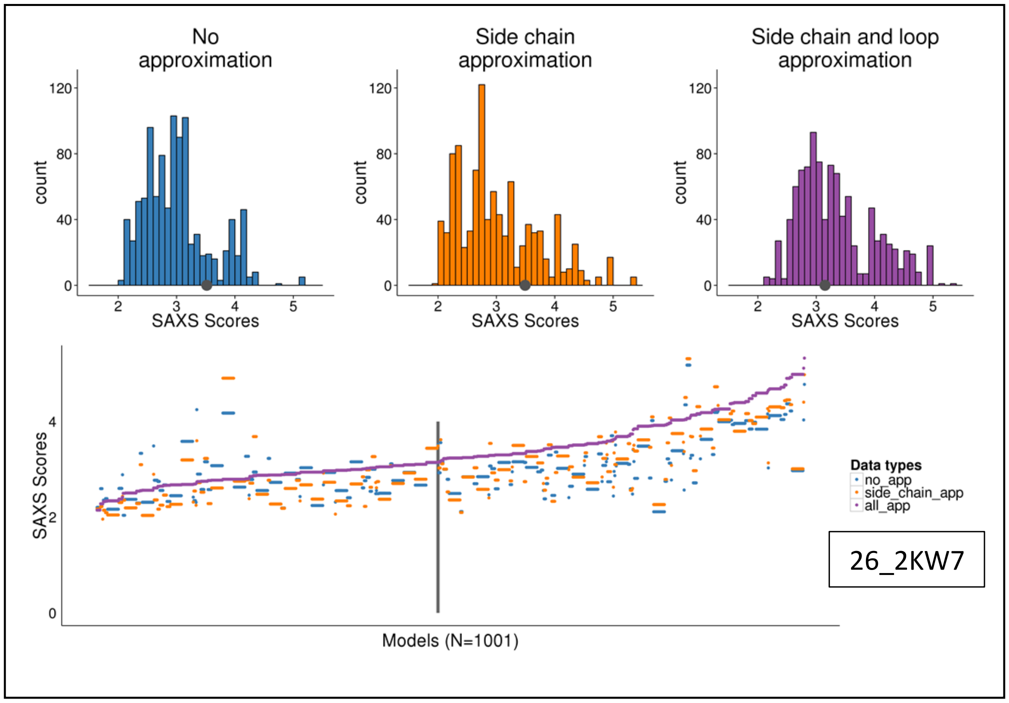

No approximation
Side chain approximation
Side chain and loop approximation
count
400
300
200
100
0
2.5
3.0
3.5
4.0
SAXS Scores
SAXS Scores
0
1
2
3
Models (N=1001)
Data types
no_app
side_chain_app
all_app
18_2KW9

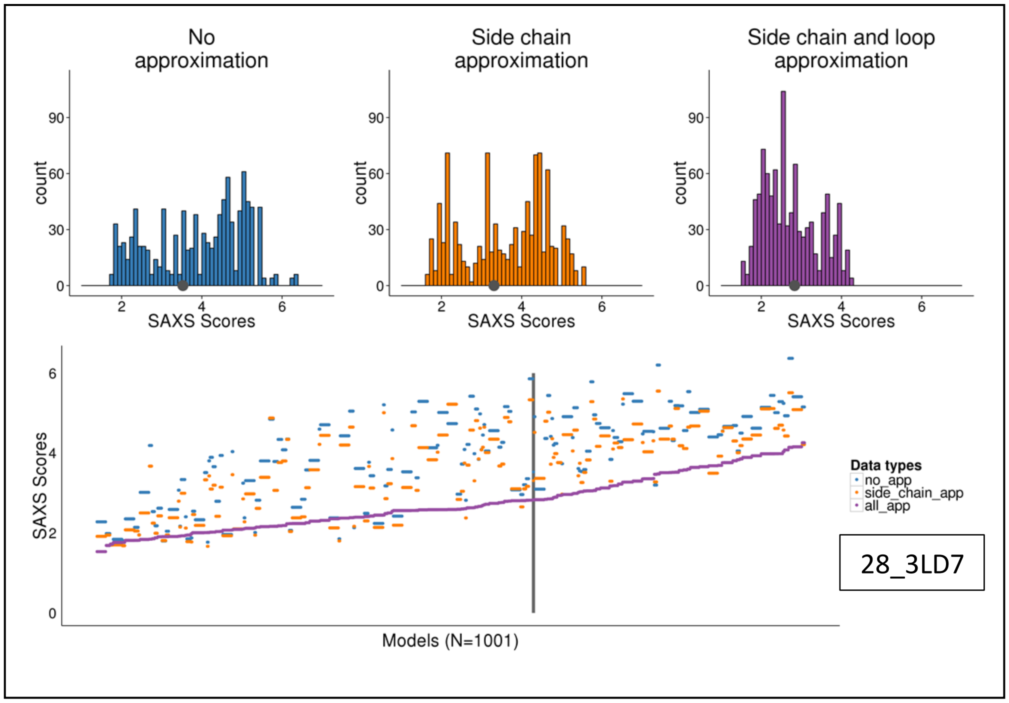

No approximation
Side chain approximation
Side chain and loop approximation
count
150
100
50
0
1
2
3
4
SAXS Scores
SAXS Scores
3
2
1
0
Models (N=1001)
Data types
• no_app
• side_chain_app
• all_app
20_2KVZ


No approximation
Side chain approximation
Side chain and loop approximation
count
150
100
50
0
1
2
3
4
SAXS Scores
SAXS Scores
3
2
1
0
Models (N=1001)
Data types
• no_app
• side_chain_app
• all_app
21_2LOB


No approximation
Side chain approximation
Side chain and loop approximation
count
0
50
100
150
2
3
4
5
SAXS Scores
SAXS Scores
0
1
2
3
4
5
Models (N=1001)
Data types
no_app
side_chain_app
all_app
22_2KZ5


No approximation
Side chain approximation
Side chain and loop approximation
count
0
40
80
120
2
3
4
5
SAXS Scores
SAXS Scores
0
2
4
Models (N=1001)
Data types
no_app
side_chain_app
all_app
24_2L0D


No approximation
Side chain approximation
Side chain and loop approximation
count
120
80
40
0
2
3
4
5
SAXS Scores
SAXS Scores
4
2
0
Models (N=1001)
Data types
no_app
side_chain_app
all_app
26_2KW7


No approximation
Side chain approximation
Side chain and loop approximation
count
90
60
30
0
2
4
6
SAXS Scores
SAXS Scores
6
4
2
0
Models (N=1001)
Data types
no_app
side_chain_app
all_app
28_3LD7

APPENDIX IX –BCL::SAXS COMMAND LINES

Comparing SAXS profiles

The BCL application, “restraint:AnalyzeAgreement” 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. Add the name of the input pdb file into the file pdbs.ls

Complete models

```
bcl-apps-static.exe restraint:AnalyzeAgreement -analysis_prefix 3HZ7 -aaclass
AAComplete -analysis_type_enumerated "AnalyzeSas( c1=1, c2=0,
experimental_profile=01_SAXS.dat, optimize_hydration_parameters=false,
default_search_grid=true, scoring_function=chi, use_errors=1, cpu=false,
use_sans=false, approximate_side_chains=false, approximate_loops=false,
transformations( Normalize, Log10, Scale ), print_transformations=false,
y_max=1)"
```

Approximating side chains

```
bcl-apps-static.exe restraint:AnalyzeAgreement -analysis_prefix 3HZ7 -aaclass
AABackBone -analysis_type_enumerated "AnalyzeSas( c1=1, c2=0,
experimental_profile=01_SAXS.dat, optimize_hydration_parameters=false,
default_search_grid=true, scoring_function=chi, use_errors=1, cpu=false,
use_sans=false, approximate_side_chains=true, approximate_loops=false,
transformations( Normalize, Log10, Scale ), print_transformations=false,
y_max=1)"
```

Approximating side chains and loop regions

```
bcl-apps-static.exe restraint:AnalyzeAgreement -analysis_prefix 3HZ7 -aaclass
AABackBone -analysis_type_enumerated "AnalyzeSas( c1=1, c2=0,
experimental_profile=01_SAXS.dat, optimize_hydration_parameters=false,
default_search_grid=true, scoring_function=chi, use_errors=1, cpu=false,
use_sans=false, approximate_side_chains=true, approximate_loops=true,
transformations( Normalize, Log10, Scale ), print_transformations=false,
y_max=1)"
```

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.

APPENDIX X –TRANSMEMBRANE DOMAIN COMMANDLINES

Secondary structure prediction with MASP

```
run_command( "cd ".$blue_dir_sspred.";
../../scripts/MembraneAssociationAndSecondaryStructurePredictor.py
".$fasta_link_name." > ".$log_file."; cd - > /dev/null"); # run script
```

Visualize secondary structure prediction

```
run_command( "visualize_sspred.pl --target ".$target." --blue_dir ".$blue_dir);
```

Fold cyclic trimer with BCL::Fold

```
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/bcl-apps-
static.exe protein:Fold -fasta
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/Q941L0/dat
a/Q941L0.fasta -sequence_data
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/Q941L0/ssp
red/ Q941L0 -sspred MASP PSIPRED OCTOPUS -pool
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/Q941L0/ssp
red/short.pool -pool_separate -stages_read
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/materials/
no_restraint/membrane_stages.txt -protein_storage
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/Q941L0/fol
d/no_restraint Overwrite -prefix test_ -nmodels 1 -opencl Disable -random_seed -
membrane -tm_helices
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/Q941L0/ssp
red/short.pool -symmetry C3 -fasta_chain_id A
```

Fold monomer with BCL::Fold

```
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/bcl-apps-
static.exe protein:Fold -fasta
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/Q941L0/dat
a/Q941L0.fasta -sequence_data
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/Q941L0/ssp
red/ Q941L0 -sspred MASP PSIPRED OCTOPUS -pool
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/Q941L0/ssp
red/short.pool -pool_separate -stages_read
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/materials/
no_restraint/membrane_stages.txt -protein_storage
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/Q941L0/fol
d/no_restraint Overwrite -prefix test_ -nmodels 1 -opencl Disable -random_seed -
membrane -tm_helices
/dors/meilerlab/home/putnamdk/Oakridge/transmembrane_modeling/denovo/Q941L0/ssp
red/short.pool -fasta_chain_id A
```

Titan commandline for BCL::Fold

```
#!/bin/bash

seed=`cat /dev/urandom|od -N4 -An -t u`

$MEMBERWORK/bip124/bcl-apps-compute-node.exe protein:Fold -fasta
$MEMBERWORK/bip124/denovo/Q941L0/data/Q941L0.fasta -sequence_data
$MEMBERWORK/bip124/denovo/Q941L0/sspred/ Q941L0 -sspred MASP PSIPRED OCTOPUS -
```

```
pool $MEMBERWORK/bip124/denovo/Q941L0/sspred/short.pool -pool_separate -
stages_read
$MEMBERWORK/bip124/denovo/materials/no_restraint/membrane_stages.txt -
protein_storage $MEMBERWORK/bip124/denovo/Q941L0/fold/no_restraint Overwrite -
prefix test_ -nmodels 1 -random_seed -membrane -tm_helices
$MEMBERWORK/bip124/denovo/Q941L0/sspred/short.pool -symmetry C3 -fasta_chain_id
A -histogram_path $MEMBERWORK/bip124/histogram/rev_4782/
```

Titan PBS Script

```
PBS Script of Production Run

#!/bin/bash
#PBS -l walltime=4:00:00
#PBS -o output/
#PBS -l nodes=126
#PBS -A BIP124
#PBS -j oe

cd $MEMBERWORK/bip124/

aprun -n 2016 $MEMBERWORK/bip124/CESA_tm_trimer.sh
```

APPENDIX XI –ZINC FINGER N-TERMINAL DOMAIN COMMANDLINES

Denovo folding with Rosetta

```
#!/bin/bash
#SBATCH --mem=2000mb
#SBATCH --time=4:00:00
#SBATCH --nodes=1
#SBATCH -o
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/ZfvrA/denovo_61_247/outp
ut/log.txt

export
LD_LIBRARY_PATH=/dors/meilerlab/apps/Linux2/x86_64/gcc/4.8.2/lib64/:/dors/meile
rlab/apps/rosetta/rosetta_2014.35.57232/main/source/build/external/release/linu
x/2.6/64/x86/gcc/4.8/default/:/dors/meilerlab/apps/Linux2/x86_64/lib64/:$LD_LIB
RARY_PATH

seed=$SLURM_JOBID

/dors/meilerlab/apps/rosetta/rosetta_2014.35.57232/main/source/bin/AbinitioRela
x.linuxgccrelease -in:file:fasta
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/ZfvrA/denovo_61_247/CESA
.fasta -in:file:frag3
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/ZfvrA/denovo_61_247/aat0
00_03_05.200_v1_3 -in:file:frag9
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/ZfvrA/denovo_61_247/aat0
00_09_05.200_v1_3 -abinitio:relax -relax:fast -abinitio::increase_cycles 10 -
abinitio::rg_reweight 0.5 -abinitio::rsd_wt_helix 0.5 -abinitio::rsd_wt_loop 0.5
-use_filters true -psipred_ss2
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/ZfvrA/denovo_61_247/t000
_.psipred_ss2 -kill_hairpins
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/ZfvrA/denovo_61_247/t000
_.psipred_ss2 -nstruct 10 -out:file:silent
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/ZfvrA/denovo_61_247/mode
ls/${seed}_silent.out
```

Extract PDBs from silent files

```
cat binary.ls | awk '{system("
/dors/meilerlab/apps/rosetta/rosetta_2014.35.57232/main/source/bin/score.linuxg
ccrelease -in:file:silent "$1" -in:file:silent_struct_type binary -
in:file:fullatom -out:output -out:pdb -out:file:fullatom -out:prefix "$2" ")}'
```

Add SEQRES lines to the PDB

```
cat pdbs.ls | awk '{system("dssp2 -i "$1" -o "$2".dssp ")}'
cat pdbs.ls | awk '{system("dssp2pdb "$2".dssp "$2".pdb > "$2".dssp.pdb ")}'
cat pdbs.ls | awk '{system("/hd0/putnamdk/workspace/bcl-
testing/build/linux64_release/bin/bcl-apps-static.exe protein:PDBConvert
"$2".dssp.pdb -bcl_pdb -output_prefix "$2"_")}'
```

Score Models with Q3 metric

```
cat bcl_models.ls | awk '{system("/dors/meilerlab/home/heinzes1/bin/ssstat -r
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/r61247/fold/denovo_61_24
7/models/"$1" -s
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/r61247/fold/r61247A.SSPr
edHighest_CONSENSUS.pool &> "$1".out")}'
```

Homology modeling with Rosetta

```
#!/bin/bash
#SBATCH --mem=2000mb
#SBATCH --time=3:00:00
#SBATCH --nodes=1
#SBATCH -o
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/ZfvrA/homology/output/lo
g.txt

export
LD_LIBRARY_PATH=/dors/meilerlab/apps/Linux2/x86_64/gcc/4.8.2/lib64/:/dors/meile
rlab/apps/rosetta/rosetta-
3.5/rosetta_source/build/src/release/linux/2.6/64/x86/gcc/4.7/default/:$LD_LIBR
ARY_PATH

seed=$SLURM_JOBID
model="8"

/dors/meilerlab/apps/rosetta/rosetta-
3.5/rosetta_source/bin/loopmodel.default.linuxgccrelease -s
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/ZfvrA/homology/model_0${
model}/threaded_1weo_mod${model}_A.pdb -loops:fa_input -loops:loop_file
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/ZfvrA/homology/Cesa_.loo
ps -loops:frag_sizes 9 3 1 -loops:frag_files
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/ZfvrA/homology/aat000_09
_05.200_v1_3
/dors/meilerlab/home/putnamdk/Oakridge/zinc_finger_new/ZfvrA/homology/aat000_03
_05.200_v1_3 none -loops:remodel quick_ccd -loops:refine refine_kic -
loops:extended true -loops:idealize_after_loop_close -loops:relax fastrelax -
loops:fast -ex1 -ex2 -database /dors/meilerlab/apps/rosetta/rosetta-
3.5/rosetta_database -nstruct 100
```

Rosetta on Titan

```
#!/bin/bash
#PBS -l walltime=2:00:00
#PBS -o output/
#PBS -l nodes=5
#PBS -A BIP124
#PBS -j oe

cd $MEMBERWORK/bip124/
seed=`cat /dev/urandom|od -N4 -An -t u`

aprun -n 80
$MEMBERWORK/bip124/rosetta_bin_linux_2015.39.58186_bundle/main/source/bin/minir
osetta.linuxgccrelease -run:protocol broker -broker:setup
$MEMBERWORK/bip124/setup_init.tpb -nstruct 80 -out:file:scorefile score.fsc -
in:file:fasta $MEMBERWORK/bip124/ZfvrA.fasta -in:file:frag3
$MEMBERWORK/bip124/aat000_03_05.200_v1_3.txt -in:file:frag9
$MEMBERWORK/bip124/aat000_09_05.200_v1_3.txt -symmetry:symmetry_definition
$MEMBERWORK/bip124/c2_denovo.sym -database
$MEMBERWORK/bip124/rosetta_bin_linux_2015.39.58186_bundle/main/database/ -
out:pdb -out:prefix ${seed} -relax:fast -relax:jump_move -
```

```
symmetry:initialize_rigid_body_dofs -fold_and_dock::rotate_anchor_to_x -
rg_reweight 0.001 -rigid_body_cycles 1 -abinitio::recover_low_in_stages 0 -
rigid_body_frequency 5 -rigid_body_disable_mc -
run:reinitialize_mover_for_each_job
```

BIBLIOGRAPHY

1. Gopal Vandavasi, V., et al., *A Structural Study of CESA1 catalytic domain of Arabidopsis thaliana Cellulose Synthesis Complex: Evidence for CESA trimers.* Plant Physiol, 2015(169).

2. Glatter, O. and O. Kratky, *Small Angle X-Ray Scattering*. 1982, New York: Academic Press Inc. 515.

3. 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.

4. Putnam, D.K., E.W. Lowe, and J. Meiler, *Reconstruction of Saxs Profiles from Protein Structures.* Computational and Structural Biotechnology Journal, 2013. **8**(11): p. 1-12.

5. 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.

6. 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.

7. 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.

8. 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.

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

10. 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.

11. Stuhrmann, H.B., *Interpretation of small-angle scattering functions of dilute solutions and gases. A representation of the structures related to a one-particle scattering function.* Acta Crystallographica Section A, 1970. **26**(3): p. 297-306.

12. Pelikan, M., G.L. Hura, and M. Hammel, *Structure and flexibility within proteins as identified through small angle X-ray scattering.* General Physiology and Biophysics, 2009. **28**(2): p. 174-189.

13. Schneidman-Duhovny, D., S.J. Kim, and A. Sali, *Integrative structural modeling with small angle X-ray scattering profiles.* BMC Struct Biol, 2012. **12**: p. 17.

14. Bernado, P. and D.I. Svergun, *Structural analysis of intrinsically disordered proteins by small-angle X-ray scattering.* Mol Biosyst, 2012. **8**(1): p. 151-67.

15. Feigin, L.A. and D.I. Svergun, *Structure Analysis by Small-Angle X-Ray and Neutron Scattering*. 1987, New York: Plenum Press.

16. Blessing, R.H., *Introduction to X-Ray Diffraction Physics*, 2006, Hauptman-Woodward Medical Research Institute. p. 55.

17. 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.

18. 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.

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

20. 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.

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

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. Pantos, E. and J. Bordas, *Supercomputer simulation of small angle X-ray scattering, electron micrographs and X-ray diffraction patterns of macromolecular structures.* Pure and Applied Chemistry, 1994. **66**(1): p. 77-82.

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

29. 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.

30. dos Reis, M.A., R. Aparicio, and Y. Zhang, *Improving protein template recognition by using small-angle x-ray scattering profiles.* Biophys J, 2011. **101**(11): p. 2770-81.

31. Edmonds, A.R., *Angular Momentum in Quantum Mechanics*. 1960, Princeton, New Jersey: Princeton University Press.

32. Liu, H., et al., *Computation of small-angle scattering profiles with three-dimensional Zernike polynomials.* Acta Crystallographica Section A, 2012. **68**(Pt 2): p. 278-85.

33. Rambo, R.P. and J.A. Tainer, *Accurate assessment of mass, models and resolution by small-angle scattering.* Nature, 2013. **496**(7446): p. 477-81.

34. Gumerov, N.A., et al., *A hierarchical algorithm for fast Debye summation with applications to small angle scattering.* J Comput Chem, 2012. **33**(25): p. 1981-96.

35. Grossmann, J.G., et al., *X-Ray-Scattering Using Synchrotron-Radiation Shows Nitrite Reductase from Achromobacter-Xylosoxidans to Be a Trimer in Solution.* Biochemistry, 1993. **32**(29): p. 7360-7366.

36. Svergun, D.I., et al., *Protein hydration in solution: experimental observation by x-ray and neutron scattering.* Proc Natl Acad Sci U S A, 1998. **95**(5): p. 2267-72.

37. Rambo, R.P. and J.A. Tainer, *Super-resolution in solution X-ray scattering and its applications to structural systems biology.* Annu Rev Biophys, 2013. **42**: p. 415-41.

38. 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.

39. Novotni, M. and R. Klein. *3D Zernike Descriptors for Content Based Shape Retrieval*. in *ACM Symposium on Solid Modeling and Applications*. 2003. New York.

40. Antonov, L.D., C. Andreetta, and M. Habeck. *An Efficient Parallel Gpu Evaluation of Small Angle X-Ray Scattering Profiles*. 2012.

41. Putnam, D.K., et al., *BCL::SAXS: GPU accelerated debye method for computation of small angle X Ray scattering profiles.* Proteins, 2015.

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

43. 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.

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

45. 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.

46. 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.

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

48. Lars, N., H. Mark, and P. Jan, *Fast n-body simulation with CUDA, Simulation 3*, in *GPU Gems 3*, H. Nguyen, Editor. 2008, Pearson Education, Inc: Boston MA. p. 677-697.

49. Bonneau, R., et al., *Contact order and ab initio protein structure prediction.* Protein Sci, 2002. **11**(8): p. 1937-44.

50. 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.

51. 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.

52. Leaver-Fay, A., et al., *ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules.* Methods Enzymol, 2011. **487**: p. 545-74.

53. *Performance of a GPU-based Direct Summation Algorithm for Computation of Small Angle Scattering Profile.*

54. Heinze, S., et al., *CASP10-BCL::Fold efficiently samples topologies of large proteins.* Proteins, 2015. **83**(3): p. 547-63.

55. Weiner, B.E., et al., *BCL::MP-Fold: Folding Membrane Proteins through Assembly of Transmembrane Helices.* Structure, 2013. **21**(7): p. 1107-1117.

56. 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.

57. 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.

58. Antonov, L., C. Andreetta, and T. Hamelryck, *Parallel GPGPU Evaluation of Small Angle X-Ray Scattering Profiles in a Markov Chain Monte Carlo Framework*, in *Biomedical Engineering Systems and Technologies*, J. Gabriel, et al., Editors. 2013, Springer Berlin Heidelberg. p. 222-235.

59. Berman, H.M., *The Protein Data Bank.* Nucleic Acids Research, 2000. **28**(1): p. 235-242.

60. Berman, H.M., *The Protein Data Bank: a historical perspective.* Acta Crystallographica Section A, 2008. **64**(Pt 1): p. 88-95.

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

62. 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.

63. 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.

64. Hura, G.L., et al., *Robust, high-throughput solution structural analyses by small angle X-ray scattering (SAXS).* Nat Methods, 2009. **6**(8): p. 606-12.

65. Klose, D.P., B.A. Wallace, and R.W. Janes, *2Struc: the secondary structure server.* Bioinformatics, 2010. **26**(20): p. 2624-5.

66. Kabsch, W. and C. Sander, *Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features.* Biopolymers, 1983. **22**(12): p. 2577-637.

67. Andersen, C.A., et al., *Continuum secondary structure captures protein flexibility.* Structure, 2002. **10**(2): p. 175-84.

68. Frishman, D. and P. Argos, *Knowledge-based protein secondary structure assignment.* Proteins, 1995. **23**(4): p. 566-79.

69. Labesse, G., et al., *P-SEA: a new efficient assignment of secondary structure from C alpha trace of proteins.* Comput Appl Biosci, 1997. **13**(3): p. 291-5.

70. Majumdar, I., S.S. Krishna, and N.V. Grishin, *PALSSE: a program to delineate linear secondary structural elements from protein structures.* BMC Bioinformatics, 2005. **6**: p. 202.

71. Taylor, W.R., *Defining linear segments in protein structure.* J Mol Biol, 2001. **310**(5): p. 1135-50.

72. Martin, J., et al., *Protein secondary structure assignment revisited: a detailed analysis of different assignment methods.* BMC Struct Biol, 2005. **5**: p. 17.

73. Zhang, Y. and J. Skolnick, *TM-align: a protein structure alignment algorithm based on the TM-score.* Nucleic Acids Res, 2005. **33**(7): p. 2302-9.

74. Cleveland, W.S., *Robust Locally Weighted Regression and Smoothing Scatterplots.* Journal of the American Statistical Association, 1979. **74**(368): p. 829-836.

75. Cleveland, W.S. and S.J. Devlin, *Locally Weighted Regression - an Approach to Regression-Analysis by Local Fitting.* Journal of the American Statistical Association, 1988. **83**(403): p. 596-610.

76. 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.

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

78. 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.

79. Dutta, S. and H.M. Berman, *Large macromolecular complexes in the Protein Data Bank: a status report.* Structure, 2005. **13**(3): p. 381-8.

80. Hopf, T.A., et al., *Three-dimensional structures of membrane proteins from genomic sequencing.* Cell, 2012. **149**(7): p. 1607-21.

81. Alber, F., et al., *Determining the architectures of macromolecular assemblies.* Nature, 2007. **450**(7170): p. 683-94.

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

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

84. Chivian, D., et al., *Prediction of CASP6 structures using automated Robetta protocols.* Proteins, 2005. **61 Suppl 7**: p. 157-66.

85. 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.

86. Bradley, P., K.M. Misura, and D. Baker, *Toward high-resolution de novo structure prediction for small proteins.* Science, 2005. **309**(5742): p. 1868-71.

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

88. Kaufmann, K.W., et al., *Practically useful: what the Rosetta protein modeling suite can do for you.* Biochemistry, 2010. **49**(14): p. 2987-98.

89. Baker, D., *A surprising simplicity to protein folding.* Nature, 2000. **405**(6782): p. 39-42.

90. Grantcharova, V., et al., *Mechanisms of protein folding.* Curr Opin Struct Biol, 2001. **11**(1): p. 70-82.

91. Wu, S., J. Skolnick, and Y. Zhang, *Ab initio modeling of small proteins by iterative TASSER simulations.* BMC Biol, 2007. **5**: p. 17.

92. Zhang, Y., *Interplay of I-TASSER and QUARK for template-based and ab initio protein structure prediction in CASP10.* Proteins, 2013.

93. Zhang, Y., *Template-based modeling and free modeling by I-TASSER in CASP7.* Proteins, 2007. **69 Suppl 8**: p. 108-17.

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

95. Rohl, C.A., et al., *Modeling structurally variable regions in homologous proteins with rosetta.* Proteins, 2004. **55**(3): p. 656-77.

96. Sali, A. and T.L. Blundell, *Comparative protein modelling by satisfaction of spatial restraints.* J Mol Biol, 1993. **234**(3): p. 779-815.

97. Rost, B., *PHD: predicting one-dimensional protein structure by profile-based neural networks.* Methods Enzymol, 1996. **266**: p. 525-39.

98. Rost, B. and C. Sander, *Combining evolutionary information and neural networks to predict protein secondary structure.* Proteins, 1994. **19**(1): p. 55-72.

99. Jones, D.T., *Protein secondary structure prediction based on position-specific scoring matrices.* J Mol Biol, 1999. **292**(2): p. 195-202.

100. Ward, J.J., et al., *Secondary structure prediction with support vector machines.* Bioinformatics, 2003. **19**(13): p. 1650-1655.

101. Leman, J.K., et al., *Simultaneous prediction of protein secondary structure and transmembrane spans.* Proteins, 2013. **81**(7): p. 1127-40.

102. Meiler, J. and D. Baker, *Coupled prediction of protein secondary and tertiary structure.* Proc Natl Acad Sci U S A, 2003. **100**(21): p. 12105-10.

103. Meiler, J., et al., *Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks.* J Mol Model, 2001. **7**(9): p. 360-369.

104. Viklund, H., et al., *SPOCTOPUS: a combined predictor of signal peptides and membrane protein topology.* Bioinformatics, 2008. **24**(24): p. 2928-9.

105. Viklund, H. and A. Elofsson, *OCTOPUS: improving topology prediction by two-track ANN-based preference scores and an extended topological grammar.* Bioinformatics, 2008. **24**(15): p. 1662-8.

106. Moult, J., *A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction.* Curr Opin Struct Biol, 2005. **15**(3): p. 285-9.

107. Moult, J., et al., *Critical assessment of methods of protein structure prediction (CASP)--round IX.* Proteins, 2011. **79 Suppl 10**: p. 1-5.

108. Taylor, T.J., et al., *Assessment of CASP10 contact-assisted predictions.* Proteins, 2013.

109. Monastyrskyy, B., et al., *Evaluation of residue-residue contact prediction in CASP10.* Proteins, 2013.

110. Nugent, T., D. Cozzetto, and D.T. Jones, *Evaluation of predictions in the CASP10 model refinement category.* Proteins, 2013.

111. Kryshtafovych, A., et al., *Assessment of the assessment: Evaluation of the model quality estimates in CASP10.* Proteins, 2013.

112. Zemla, A., et al., *Processing and analysis of CASP3 protein structure predictions.* Proteins, 1999. **Suppl 3**: p. 22-9.

113. Ginalski, K., et al., *3D-Jury: a simple approach to improve protein structure predictions.* Bioinformatics, 2003. **19**(8): p. 1015-8.

114. Canutescu, A.A. and R.L. Dunbrack, Jr., *Cyclic coordinate descent: A robotics algorithm for protein loop closure.* Protein Sci, 2003. **12**(5): p. 963-72.

115. Fischer, A.W., et al., *BCL::MP-fold: Membrane protein structure prediction guided by EPR restraints.* Proteins, 2015. **83**(11): p. 1947-62.

116. 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.

117. Svergun, D.I., *Determination of the Regularization Parameter in Indirect-Transform Methods Using Perceptual Criteria.* Journal of Applied Crystallography, 1992. **25**: p. 495-503.

118. 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.

119. 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.

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

121. 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.

122. 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.

123. 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.

124. 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.

125. 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.

126. Grant, T.D., et al., *Small angle X-ray scattering as a complementary tool for high-throughput structural studies.* Biopolymers, 2011. **95**(8): p. 517-30.

127. 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.

128. Sethaphong, L., et al., *Tertiary model of a plant cellulose synthase.* Proc Natl Acad Sci U S A, 2013. **110**(18): p. 7512-7.

129. Andre, I., et al., *Prediction of the structure of symmetrical protein assemblies.* Proc Natl Acad Sci U S A, 2007. **104**(45): p. 17656-61.

130. Morgan, J.L., J. Strumillo, and J. Zimmer, *Crystallographic snapshot of cellulose synthesis and membrane translocation.* Nature, 2013. **493**(7431): p. 181-6.

131. Hasegawa, H. and L. Holm, *Advances and pitfalls of protein structural alignment.* Curr Opin Struct Biol, 2009. **19**(3): p. 341-8.

132. Krissinel, E. and K. Henrick, *Inference of macromolecular assemblies from crystalline state.* J Mol Biol, 2007. **372**(3): p. 774-97.

133. Tsuchiya, Y., *Discrimination between biological interfaces and crystal-packing contacts.* Advances and Applications in Bioinformatics and Chemistry, 2008: p. 99.

134. Olek, A.T., et al., *The structure of the catalytic domain of a plant cellulose synthase and its assembly into dimers.* Plant Cell, 2014. **26**(7): p. 2996-3009.