HYPO: COMPUTATIONAL PREDICTION OF (HY)DROGEN (PO)SITIONS IN PROTEIN STRUCTURES FROM X-RAY AND NEUTRON CRYSTALLOGRAPHIC DATA

THESIS SUBMITTED TO THE GRADUATE DIVISION OF
THE UNIVERSITY OF HAWAI'I AT MĀNOA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN CHEMISTRY

DECEMBER 2015

BY FRANCISCO E JIMENEZ

THESIS COMMITTEE:
HO LEUNG NG, CHAIRPERSON
JOSEPH JARRETT
TOM APPLE
Acknowledgements:

For my Mother & Father, without whom

I would never have been given the spark,

and for their undying love and support.

To my wife, for always being my muse and my support.

To my son, for being my inspiration when things got difficult.


Ten thousand flowers in spring, the moon in autumn,

a cool breeze in summer, snow in winter.

If your mind isn’t clouded by unnecessary things,

this is the best season of your life.
Abstract. X-ray crystallography has become the standard means for gathering experimental structural data from proteins. However, there are problems inherent with X-ray crystallography, most especially with the explicit location of hydrogen atoms. This poses a problem, in that many critical functions carried out by protein active sites rely on fine positioning of key hydrogen atoms. Another experimental technique is neutron crystallography, which is especially good at determining hydrogen positions with a high level of accuracy, but which carries extreme difficulty. Our project aims to allow for a more systematic, accurate, and quantitative methodology for computationally locating hydrogen atom positions within proteins, using data gathered from high resolution X-ray and neutron crystallographic structures. Use of these methods will make crystallographers less dependent upon ultra-high resolution crystal structures and experimentally challenging neutron data when determining locations of critical hydrogen atoms within a protein structure.
LIST OF TABLES

Table 3.1. Primary Data Set ................................................................. 19
Table 3.2. Secondary Data Set ................................................................. 22
LIST OF FIGURES

Figure 2.1. Structure taken from PDB entry 1F0Y  ........................................... 2
Figure 2.2. Hydrogen bond network pd-Toho-1 R274/R276N-BZB  ...................... 3
Figure 2.3. Relative scattering amplitudes for common biological elements  ........ 4
Figure 2.4. The electric field vector $\mathbf{E}$ is perpendicular to the magnetic field vector $\mathbf{B}$ ................................. 6
Figure 2.5  
  Figure 2.5a. Representation of a plane wave in the complex and real planes  ........ 8
  Figure 2.5b. Addition of plane waves results in different amplitudes and phases ........ 8
Figure 2.6  
  Figure 2.6a. Scattering of an incoming electric field vector $\mathbf{E}$  ...................... 10
  Figure 2.6b. Scattering vector diagram for partial waves originating at two different points  .. 10
Figure 2.7. Gaussian scattering function for a single atom  .................................. 12
Figure 2.8. Laue conditions for scattering & constructive interference .................... 14
Figure 2.9. Graphical representation of Bragg’s Law. .................................................. 15
Figure 3.1  
  Figure 3.1a. Good global and local alignment of structures  ............................. 21
  Figure 3.1b. Good global and local alignment of structures  ............................. 21
  Figure 3.1c. Good global and poor local alignment of structures  .................... 21
  Figure 3.1d. Good global and poor local alignment of structures .................... 21
Figure 3.2. Overall processing algorithm for HyPo data collection.  ......................... 23
Figure 3.3. Processing algorithm for HyPo data. ....................................................... 24
Figure 3.4  
  Figure 3.4a. Definition of the torsional angle for HyPo data collection ............. 25
  Figure 3.4b. Effects of varying the bond length and bond angle  .................... 25
Figure 4.1  
  Figure 4.1a. Ser-46, Pyrococcus furiosus rubredoxin, 0 sigma cutoff .................. 32
  Figure 4.1b. Ser-46, Pyrococcus furiosus rubredoxin, 1 sigma cutoff .................. 32
  Figure 4.1c. Ser-46, Pyrococcus furiosus rubredoxin, 2 sigma cutoff .................. 32
  Figure 4.1d. Ser-46, Pyrococcus furiosus rubredoxin, 3 sigma cutoff .................. 32
  Figure 4.1e. Ser-46, Pyrococcus furiosus rubredoxin, Neutron Density ............... 32
  Figure 4.1f. Ser-46, Pyrococcus furiosus rubredoxin, X-ray & neutron density .......... 32
Figure 4.2  
  Figure 4.2a. Correlation between neutron resolution and number of hydrogen peaks detected 33
  Figure 4.2b. Correlation between X-ray resolution and number of hydrogen peaks detected ... 33
Figure 4.3. Correlation between hydrogen detection rate, x-ray, and neutron resolution  .... 34
Figure 4.4.  

Figure 4.4a. plot of scattering density vs. torsion angle for X-ray and neutron data for the ser-46 hydroxyl of oxy-myoglobin ................................................................. 35

Figure 4.4a. plot of scattering density vs. torsion angle for X-ray and neutron data for the ser-46 hydroxyl of oxy-myoglobin ................................................................. 35

Figure 4.5.  

Figure 4.5a. Correlation between neutron resolution and peak density match rate ............ 37

Figure 4.5b. Correlation between X-ray resolution and peak density match rate ............ 37

Figure 4.6. Correlation between bond length and positive match rate for a hydrogen density peak between related neutron and X-ray structures ......................................................... 38

Figure 4.7.  

Figure 4.7a. Total number of successfully identified hydrogen density peaks .................. 39

Figure 4.7b. Percentage of successfully identified hydrogen density peaks .................... 39
LIST OF SYMBOLS
(In order of appearance.)

NAD+ Nicotinamide adenine dinucleotide, de-protonated form
NADH Nicotinamide adenine dinucleotide, protonated form
CoA Coenzyme A
His Histidine
$F_{obs}$ Fourier observed
$F_{calc}$ Fourier calculated
$E$ Electric field vector
$\nabla^2$ LaPlace operator
$\mu$ Reduced mass
$\partial$ Legendre partial derivative
$t$ Time
$B$ Magnetic field vector
$A$ Amplitude vector
$\lambda$ Lambda, wavelength
$\otimes$ Cross product operator
$s_0$ Incoming electric field vector
$\| \|$ Magnitude or absolute value
$E$ Energy
$h$ Planck’s Constant, $6.626 \times 10^{-34}$ J · s
$\nu$ Frequency
$c$ Speed of light in vacuo, $2.99 \times 10^8$ m · s$^{-1}$
$z$ Position
$\pi$ Pi, 3.1415....
$\cos$ Cosine
$\omega$ Omega = $2\pi\nu$
$\varphi$ Phase of a plane wave
$\Psi$ Psi, wavefunction
$i$ \[ \sqrt{-1} \]
$\sum_{j=1}^{n} f(x)$ Sigma, summation from j to n of f(x)
\( \rho \)  \hspace{1cm} \text{Rho, density}

\( r \)  \hspace{1cm} \text{Radial vector}

\( \Delta \)  \hspace{1cm} \text{Delta, difference}

\( \cdot \)  \hspace{1cm} \text{Dot product operator}

\( f \)  \hspace{1cm} \text{Integration operator}

\( s_1 \)  \hspace{1cm} \text{Scattered vector}

\( S \)  \hspace{1cm} \text{Total scattering vector}

\( F_S \)  \hspace{1cm} \text{Structure factor}

\( K_S \)  \hspace{1cm} \text{Total wave for a crystal}

\( \theta \)  \hspace{1cm} \text{Theta, angle}

\( d \)  \hspace{1cm} \text{Bragg plane spacing}

\( hkl \)  \hspace{1cm} \text{Miller indices}

\( d^* \)  \hspace{1cm} \text{Reciprocal space Bragg spacing}

PDB  \hspace{1cm} \text{Proten Data Bank}

CIF  \hspace{1cm} \text{Crystallographic Information File}

MTZ  \hspace{1cm} \text{Reflection data file}

fig.  \hspace{1cm} \text{Figure}

CSV  \hspace{1cm} \text{Comma Separated Value}

CCP4  \hspace{1cm} \text{Collaborative Computational Project No. 4}

\( \text{Å} \)  \hspace{1cm} \text{Angstroms}

Ser  \hspace{1cm} \text{Serine}

Gln  \hspace{1cm} \text{Glycine}

Asn  \hspace{1cm} \text{Asparagine}

Glu  \hspace{1cm} \text{Gluamate / Glutamic Acid}

Asp  \hspace{1cm} \text{Aspartate}

Arg  \hspace{1cm} \text{Arginine}

Lys  \hspace{1cm} \text{Lysine}
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Met</td>
<td>Methionine</td>
</tr>
<tr>
<td>Cys</td>
<td>Cysteine</td>
</tr>
<tr>
<td>Leu</td>
<td>Leucine</td>
</tr>
<tr>
<td>Phe</td>
<td>Phenolalanine</td>
</tr>
<tr>
<td>Tyr</td>
<td>Tryosine</td>
</tr>
<tr>
<td>Trp</td>
<td>Tryptophan</td>
</tr>
<tr>
<td>His</td>
<td>Histidine</td>
</tr>
</tbody>
</table>
1. Introduction

Protein structure and function elucidation have become increasingly important, as more and more improvements are being made to the experimental methodology for collecting structural data from proteins. The most ubiquitous method for structural data collection, X-ray crystallography, has become widespread in its use as a means for gathering experimental structural data from proteins. However, there are problems inherent with X-ray crystallography, most especially with the explicit location of hydrogen atoms. This poses a problem, in that many critical functions carried out by protein active sites rely on fine positioning of key hydrogen atoms. Another experimental technique which has gained in popularity recently is neutron crystallography, which has its own strengths and weaknesses. Neutron crystallography is especially good at determining hydrogen positions with a high level of accuracy. Ideally, crystallographers would like to combine the two techniques, to get a more complete view of a given protein’s structure, but this is not always realistic from an experimental viewpoint. Our project aims to allow for a more systematic, accurate, and quantitative methodology for computationally locating hydrogen atom positions within proteins, using data gathered from high resolution X-ray and neutron crystallographic structures. Use of these methods will make crystallographers less dependent upon ultra-high resolution crystal structures and experimentally challenging neutron data when determining locations of critical hydrogen atoms within a protein structure.
2. Background

Hydrogen atoms are an indispensable part of understanding the structure and function of proteins. In most proteins, the average ratio of hydrogen atoms to non-hydrogen atoms—weighted to account for amino acid frequency—is 1.01. This means that hydrogen account for roughly 50% of all atoms in an average protein. Hydrogen atoms are a critical element of protein structure, as many biochemical reactions depend upon hydrogen atoms in some way. It is often essential to understand hydrogen positioning within a specific part of a protein as completely as possible, in order to understand its function and mechanism. In addition, it is often important to know the protonation state of a specific amino acid during different stages of a chemical reaction. For example, in some L-3-hydroxyacyl-CoA dehydrogenases, an alcohol oxidation is carried out which requires NAD$^+$ as a coenzyme, and yields reduced NADH/H$^+$ as a byproduct. Deprotonation of the hydroxyl group on CoA is facilitated by a His-158 residue, and a hydride ion is added to the nicotinamide ring of NAD$^+$. In this case, knowledge of the physical orientation and protonation states of the substrate, cofactor, and His-158 residue proved to be critical factors in mechanistic elucidation. Another case in which it is important to have discrete information on proton orientation involves solvent networks of water molecules, in which hydrogen bonding helps to stabilize a highly networked conformation. In many cases, an intimate understanding of the dynamics of hydrogen atoms was tantamount to mechanistic elucidation.

Unfortunately, determination of hydrogen atom data has proven experimentally problematic and challenging, especially at lower resolutions. The most common technique for determining protein structural
Figure 2.2. Hydrogen bond network of the active site region of the pd-Toho-1 R274/R276N-BZB complex showing important catalytic residues, a catalytic water molecule (wat1), and a second bound water molecule (wat2). In this particular case, knowledge of the exact orientations of the hydrogen atoms involved was critical in achieving an understanding of this system.

Data is X-ray crystallography. This technique has improved considerably in recent years, and more and more structures are becoming available that have been solved at extremely high resolutions. X-ray radiation interacts with the electronic structure of atoms, which causes it to scatter. More intense scattering is observed when atoms have more dense electron clouds. Thus, atoms with a larger atomic number tend to interact more strongly with X-rays and display a stronger scattering signal. This makes hydrogen atoms all but invisible to X-ray scattering techniques. Hydrogens do start to become indirectly visible at very high resolutions, but resolution alone does not seem to be the only factor governing hydrogen visibility in X-ray crystal structures. Also, increasing the intensity of the radiation does not show an increase in hydrogen signal. There are experimental cases in which a higher resolution structure yielded poorer $F_{\text{obs}} - F_{\text{calc}}$ maps than a lower resolution counterpart. In addition to the weak interaction of hydrogen atoms with X-rays, there are other difficulties which can make X-ray data acquisition problematic. X-ray bombardment is an extremely high energy technique, and is therefore destructive to protein crystals. The technique also requires very cold temperatures, which add another layer of experimental difficulty. It may not always be easy experimentally to create a host of large, pure crystals for data collection.

Neutron diffraction is a technique that has been gaining popularity for some time amongst crystallographers interested in hydrogen position data. Neutrons have a number of properties that make them behave very differently from X-rays. They have a very weak interaction with matter because they are uncharged, and are therefore able to penetrate very deeply into solid matter. Interestingly, atomic scattering amplitude of neutrons is not tied to an atom’s electron density. One side effect of this is that the magnitude of interaction for neutrons is different for different isotopes of a given element. In addition, neutron diffraction amplitudes follow no simple pattern for scattering interaction the way X-rays do.
Hydrogen has a neutron scattering amplitude similar to that of uranium\textsuperscript{48}. Unfortunately for protein crystallographers, the incredibly strong neutron scattering by hydrogen is incoherent. Neutrons are spin $\frac{1}{2}$, and they interact almost exclusively with the nucleus, which can greatly affect the spin-dependent cross-section\textsuperscript{59}. Incoherent scattering manifests itself as an isotropically-based random scattering effect, which in the case of hydrogen, gives very poor signal to noise ratios, with enough noise to interfere with signal from neighboring atoms. Fortunately, this effect can be overcome by substituting with deuterium\textsuperscript{50}. Of course, this comes at a high experimental price. Perdeuteration of proteins adds an entirely new layer of experimental cost and difficulty. Extremely large, pure crystals are required in order to run neutron experiments, and these can be very challenging to grow experimentally. Also, because the flux of neutron beams is much weaker than that of X-rays, data collection takes much longer—many days as opposed to the seconds of collection time experienced with synchrotron X-ray radiation. Fortunately, neutron crystallography is a non-destructive method, so the sample is preserved and collection can be performed at ambient temperature. These conditions make neutron crystallography nearly impossible to carry out for most proteins. There are approximately 50 neutron crystal structures in the protein data bank, as compared to over 100,000 X-ray structures.

Computational methods exist for determining positions of hydrogen atoms\textsuperscript{2,13,60}, and for many of the hydrogen atoms in a protein, work very well. A high percentage of hydrogen atoms are constrained geometrically and sterically by their surrounding neighbors such that they have very easily determined bond lengths and dihedral angles. Often, it is sufficient to add hydrogens using a fixed bond length and subsequently minimize the protein’s energy using force field methods. However, there are cases for which these simple kinetic and electrostatic calculations fall short. Rotamerically unhindered hydroxyl hydrogens, such as that pictured in fig. 2.1, have too much freedom of movement to be confidently modeled computationally. Other factors can be taken into account, such as orientation with a neighboring residue, but it would be far preferable to have positional data gleaned from the experimental data directly. Similarly, water molecules can adopt a great many orientations, and it can be easy to overlook an important water-based interaction without solid experimental data to back the calculation. Additionally,
the lowest energy conformation is not necessarily the correct one in reality. Knowing the difference between a lowest energy conformation and the actual observed conformation can provide deep insight into thermodynamic energy barriers for reactions. Additionally, being able to explicitly verify hydrogen positional data based solely on experimental data could provide a means of verifying the accuracy of existing approaches of a purely computational nature, and lead to more robust parameterization of empirically based computational methods in the future.
2.1. X-Ray Diffraction - A Very Basic Review. Scattering of X-rays by electrons is the fundamental process by which crystallographers collect their data. It is important to understand the mathematics behind the emergence of signal from X-rays scattered by electrons in a crystal structure, in order to understand part of what makes a quality data set. It is also very useful in achieving an understanding of what we are talking about when we discuss scattering density throughout this paper.

Waves scattered by a crystal must be summed in order to determine the scattering pattern of diffracted X-rays. Each single wave in the final summation is scattered by a single electron. Each unit cell of a crystal can have thousands of electrons. Each crystal can have millions of unit cells. It is therefore essential to understand the means by which we make this problem of summation simple and fast so that we may easily relate the final scattering pattern back to the electron density of the crystal and the units cells therein.

The underlying physical process for X-ray scattering is the interaction of the electric field vector $E$ of a propagating electromagnetic wave packet from each photon with the electrons in a given crystal. Light propagates as a 3D vector field with both magnetic ($B$) and electric ($F$) fields. The 3D wave equation for the electric field is

$$\nabla^2 E - \mu_e \frac{\partial^2 E}{\partial t^2} = 0$$ (2.1)

The electric field vector is normal to the propagation direction of the wave vector, and oscillates with a frequency corresponding to the energy and the inverse of the wavelength of the X-rays (fig. 2.4). When these photons travel through a crystal, 99% of the time nothing happens. However, the electric field vector $E$ can induce coherent oscillations in the crystal’s electrons along the photon’s coherence length by photon absorption. The electrons can then re-emit virtual waves, which will overlap either destructively or constructively, resulting in a re-emitted photon that has been “scattered” in a different direction from the original wave vector $E$. The probability of this photon being re-emitted is proportional to the amplitude of the overlapping waves in a given direction. This is why sums become so important.

*Figure 2.4.* The electric field vector $E$ is perpendicular to the magnetic field vector $B$. Both oscillate normal to the propagation vector $s_o$, which lies along the x-axis. In X-ray crystallography, we are mostly concerned with the electric field vector, as it is the interaction between $E$ and the electron cloud of individual atoms that yields a scattering pattern.
in diffraction mathematics: the sum of all scattering events of independent photon re-emissions is what generates the diffraction pattern and relates it back to the electron density of the crystal and the unit cell.

The cross product of the magnetic field vector $\mathbf{B}$ and the electric field vector $\mathbf{E}$ returns the incident propagation vector, as both waves propagate normal to $\mathbf{s}_0$. In addition, the magnitude of $\mathbf{s}_0$ is defined as $1/\lambda$, which becomes very important in later calculations.

$$E \otimes B = \mathbf{s}_0 \text{ with } ||\mathbf{s}_0|| = \frac{1}{\lambda}$$

(2.2)

Fortunately, we can discard the magnetic field vector $\mathbf{B}$ for the most part, as its contribution to scattering effects of X-rays on electrons is negligible. We can therefore treat the electric field vector $\mathbf{E}$ as a classical, plane, transversal sine wave function with amplitude $A$, given by the magnitude of the electric field vector $\mathbf{E}$, oscillating at a given frequency $\nu$. Here frequency represents the frequency of the photon with electric field vector $\mathbf{E}$ and frequency defined by $E = h\nu$, with $\nu = c/\lambda$, where $\nu$ is the frequency of the light, $h$ is Planck’s constant, and $\lambda$ represents the wavelength. Using these definitions, for a plane sine wave at time $t=0$ and position $z$, the electric field strength can be represented thus

$$E(t = 0; z) = A \cdot \cos(2\pi) \cdot \left(\frac{z}{\lambda}\right)$$

(2.3)

During a given time $t$, the wave travels over a distance $t \times c = t \times \lambda \times \nu$. Because of the periodicity of sine and cosine functions, at time $t$ the field strength at $(t, z)$ will be equal to the field strength at $(t=0, z)$, and the position will have shifted by $z - t \times \lambda \times \nu$. This results in the following equation

$$E(t; z) = A \cdot \cos(2\pi) \cdot \left(\frac{1}{\lambda}\right) \cdot (z - t \times \lambda \times \nu) = A \cdot \cos(2\pi) \cdot \left(\frac{z}{\lambda} - \nu \cdot t\right)$$

(2.4)

$$= A \cdot \cos(2\pi \nu) \cdot \left(t - \frac{z}{c}\right)$$

(2.5)

Using $\omega$ in place of $2\pi \nu$ for convenience, we have the classic equation for a wave: $E(t; z = 0) = A \cos \omega t$. The magnitude of the incoming electric field vector is only part of the picture we need in order to gather the information we desire. Also of importance is the phase of the wave, which allows determination of destructive or constructive interference resulting from the superposition of many waves, and the amount of each type of interference occurring. If we examine a wave identical to that used in equation 2.4, and that has been displaced over a distance $\varphi = 2\pi(z/\lambda)$ from the original wave, we see that we end up with

$$A \cos(\omega t + \varphi) = A \cos(\varphi) A \cos(\omega t) + A \sin(\varphi) A \sin(\omega t)$$

(2.6)

$$= A \cos(\varphi) A \cos(\omega t) + A \sin(\varphi) A \sin(\omega t + \frac{\pi}{2})$$

(2.7)
For this purpose, it is convenient to represent a plane wave of amplitude $A$ and phase angle $\varphi$ relative to the origin as a vector with real and complex parts orthogonal to one another, which is represented by the phase shift of $\pi/2$. This allows for an elegant relationship between phase and magnitude of a plane wave with electric field vector $\mathbf{E}$ (fig. 2.5A). By representing plane waves with real and imaginary vector components, we are able to establish an efficient way to add plane waves in order to obtain the superposition of many waves and the resulting phase angles. We can see in fig. 2.5b a visual representation of this process. By converting each wave to be superposed to two parts—a real part and an imaginary part—we can obtain an equation similar to 2.6 in the form $A = \cos(\varphi) + i \sin(\varphi)$, where $i = \sqrt{-1}$. This is also evident in fig. 2.5a, where the vector $\mathbf{A}$ represents the wave itself, and the vector projection of $\mathbf{A}$ onto the real axis is represented by the component $A \cos(\varphi)$ with a phase of zero, and the vector projection of $\mathbf{A}$ onto the imaginary axis is represented by the component $A \sin(\varphi)$ with a phase of $\pi/2$. What this allows us to do is to use Euler’s famous formula to describe the complete wave vector with phase and amplitude information in a compact form:

$$e^{i\varphi} = \cos(\varphi) + i \sin(\varphi) \quad (2.8)$$

We are able to use Euler’s formula because of the periodic nature of our wave system on the interval $[0, 2\pi]$. Using the graphical representation in fig. 2.5A, we can see that for the wave vector $\mathbf{A}$

$$\mathbf{A} = Q + i P = ||\mathbf{A}|| \cos(\varphi) + i ||\mathbf{A}|| \sin(\varphi) = A \cos(\varphi) + i A \sin(\varphi) \quad (2.9)$$

and it follows that

$$\mathbf{A} = ||\mathbf{A}|| \cdot e^{i \varphi} = A \cdot e^{i \varphi} \quad (2.10)$$
Equation 2.9 gives us a powerful way to sum over \( n \) waves \( \mathbf{A} = \mathbf{A}^1 + \mathbf{A}^2 + \mathbf{A}^3 + \ldots + \mathbf{A}^n \) using simple exponential summation

\[
\mathbf{A} = \sum_{j=1}^{n} \|\mathbf{A}_j\| \cdot e^{i\varphi_j} = \sum_{j=1}^{n} A_j \cdot e^{i\varphi_j}
\]  

(2.11)

What we have done is to represent each wave as a 2D vector, where the length of each vector represents the magnitude of the wave, and the vector angle corresponds to the phase. By summing all the separate wave vectors for a given system, we can calculate the vector representing the total wave of the system.

Now that we have established the mathematics for the propagation and superposition of electric field waves \( \mathbf{E} \), we can begin to look at how these waves actually behave when they interact with objects. The DeBroglie wavelength of X-ray radiation makes them well suited for resolving atoms that are spaced by standard covalent bonds. This means that when an X-ray wave encounters an object of ideal size in its path, those objects can act as points of propagation for new waves. It is these new waves that are what we think of as scattered waves, and they have some interesting properties that will allow us to determine the three-dimensional position of the objects by which they are scattered. Huygen’s principle states that every point of interaction along a wavefront in turn becomes the source of an entirely new wavefront. The velocity of this new wavefront is essentially equal to that of the original in the case of X-ray scattering because the scattering process is mostly elastic, without the transfer of energy between electron and photon. It is often useful to think of the incoming wave vector as the scattering vector \( \mathbf{S} \), with \( \mathbf{s}_0 \) representing the the incoming electric field vector, and \( \mathbf{s}_1 \) representing the scattered wave vector.

\[
\mathbf{S} = \mathbf{s}_1 - \mathbf{s}_0 \text{ with } \|\mathbf{S}\| = \|\mathbf{s}_1 - \mathbf{s}_0\|
\]  

(2.12)

Also of benefit is examination of a simple model system first. Before looking at the scattering behavior of a complex system such as a crystal lattice, we can examine the scattering behavior of a single atom. The electron cloud of an atom exists as stable, well-defined orbitals that are represented by probability distributions related to the square of each atom’s corresponding wave function, \( \Psi \). The amplitude of a wave scattered by an atom will be related to the electron cloud’s probability distribution in space. It follows then, that the electron density \( \rho(\mathbf{r}) \) with respect to a given position \( \mathbf{r} \) in any atom will be of importance when examining scattering behavior.

In order to do this easily, we need to assume the electron density of an atom \( \rho(\mathbf{r}) \) as being spherical, which is not necessarily true, especially for higher level orbitals. In this case, the spherical approximation is a reasonable approximation to use\(^{551}\), and the complexity of calculations will be reduced dramatically as we will be able to leverage symmetry. As we can see in fig. 2.6a, if the wavelength of the incoming radiation and the size of an object are of comparable size, scattering will occur in a mostly forward direction. The scattering function is proportional to the square root of the intensity of the scattered X-ray photons, which is represented by a gaussian-like distribution in fig. 2.6a. One effect that occurs
as the result of high similarity between the DeBroglie wavelength of the incoming radiation and the size of the scattering object is that the partial waves scattered from the rear of the object experience a phase shift from those in scattered by the front, meaning that the further the phase angle shifts away from that of the incoming propagation vector \( s_0 \), the smaller the amplitudes of the scattered partial waves become. This is why we see a peak in the scattering function when \( \varphi(s_0) = \varphi(s_1) \).

This weakening of backward scattering as a function of phase angle can be quantified if we take a closer look at the relationship between partial waves emitted from different points of an atom’s electron density cloud \( \rho(r) \). If we look at fig. 2.6b, we can see that by placing a point \( R_1 \) at the center of the atom and a point \( R_2 \) somewhere near the edge of the electron density cloud \( \rho(p) \), we can calculate the path difference \( \Delta r \) between two waves emitted from different parts of the atom: \( R_1 \) and \( R_2 \). If we take \( r_{12} \) as the distance between \( R_1 \) and \( R_2 \), we can see that the path difference is related to the difference between \( AR_2 \) and \( R_1B \) in fig. 2.6b, which represent the vector projections of \( r \) onto both \( s_0 \) and \( s_1 \). We can relate these lengths using the scalar products \( r \cdot s_0 \) and \( r \cdot s_1 \). If we maintain the units of \( 1/\lambda \) for all
scattering vectors we now have:

\[ \Delta \rho = (\lambda \mathbf{r} \cdot \mathbf{s}_1 - \lambda \mathbf{r} \cdot \mathbf{s}_0) = (\mathbf{s}_1 \cdot \mathbf{s}_0) \cdot \mathbf{r} \lambda = \mathbf{S} \cdot \mathbf{r} \lambda \]  

(2.13)

We can easily convert \( \Delta \rho \) to the more useful \( \Delta \varphi \) by converting to radians using \( 2\pi/\lambda \) and substitute \( \mathbf{S} = \mathbf{s}_1 - \mathbf{s}_0 \) to give:

\[ \Delta \varphi = 2\pi \mathbf{S} \cdot \mathbf{r} \]  

(2.14)

as a general expression for the phase difference for a relative phase angle of a scattered wave with respect to a fixed point of origin. What this now allows us to do is to calculate an atomic scattering factor \( f_s \) based on the sum of partial waves scattered by atomic electron density volume element \( \rho(\mathbf{r}) \) in exponential vector form \( e^{i\varphi} \), and with relative phase \( \Delta \varphi = 2\pi \mathbf{S} \cdot \mathbf{r} \). What happens is that the magnitude of each wave is proportional to the electron density of each particular volume element, and integration gives the complete scattering function. Assuming a spherical density function allows us to also assume that \( \rho(r) = -\rho(r) \) because of centrosymmetry about the nucleus of the atom. Integration over the entire space \( \mathbf{r} \) gives:

\[ f_s = \int \rho(\mathbf{r}) \left[ e^{i(2\pi \mathbf{S} \cdot \mathbf{r})} + e^{i(-2\pi \mathbf{S} \cdot \mathbf{r})} \right] d\mathbf{r} \]  

(2.15)

and because \( \rho(r) = -\rho(r) \)

\[ f_s = 2 \int \rho(\mathbf{r}) e^{i(2\pi \mathbf{S} \cdot \mathbf{r})} d\mathbf{r} \]  

(2.16)

which is a real result that is independent of the direction of \( \mathbf{S} \), but does depend on the magnitude of \( \mathbf{S} \):

\[ \|\mathbf{S}\| = \frac{2 \sin \theta}{\lambda} \]  

(2.17)

As seen in fig. 2.6a, the scattering function for a single atom is Gaussian and is centered on \( \mathbf{S}=0 \), is symmetric because of our spherical approximation, and is directly related to the number of electrons in the atom’s electron cloud, meaning \( f_{S=0} = z \). Also of importance, is that the atomic scattering factor turns out to be directly related to the electron density of an atom by way of a Fourier transform.

We are now ready to examine how diffraction of X-rays by a system of two or more atoms might change the properties we have already established. Two or more scattering objects will behave similarly to the famous double slit experiment performed by Thomas Young in 1801, in that the scattering function will begin to show the effects of constructive and destructive interference. Interference by the atoms in a crystal lattice is of extreme importance to us, because it allows for the scattering function obtained experimentally to be amplified tremendously, thus increasing signal to noise and improving the quality of data collection. In order to accomplish this, it is important to understand the conditions required to achieve maximum constructive interference. The maximum of the scattering function will still occur at \( \mathbf{S}=0 \), with secondary maxima occurring at intervals of \( \Delta \varphi = 2\pi \) with integer multiples \( n = 1, 2, 3, ... \), and
Figure 2.7. Here we see the Gaussian scattering function for a single atom as a red, dashed line, and the scattering function for two atoms in black. Below we see a scattering vector diagram for two atoms as a blown-up view from the scattering source in the top portion of the figure. The constructive interference causes the magnitude of the $S=0$ scattering function to be twice as large as for the single atom function. However, we now have an added layer of complexity in that we need to be able to determine the phase difference at which diffraction maxima appear with respect to $S=0$.

minima at half that phase angle, or at $n \cdot \pi$. What this means is that we now have more strict conditions for maximum scattering to occur versus a single atom. The maximum constructive interference for a system of two atoms will occur at:

$$\Delta \varphi_{\text{max}} = 2\pi S \cdot r = n \cdot 2\pi$$

(2.18)

If we were to look at our two atom system as representing a real crystal lattice in three dimensions, we could see that the vector $r_{12}$ in fig. 2.7 corresponds to the crystal’s unit cell basis vector $a$ in one dimension. By adding basis vectors in the other two dimensions as $b$ and $c$, we would then have a way to describe the conditions for maximum constructive interference in three dimensions using three independent equations.

$$S \cdot a = h, \ S \cdot b = k, \ S \cdot c = l$$

(2.19)

We can start to see that the conditions for maximum constructive interference of forward-scattered waves is very important, and that understanding the conditions necessary to create this experimentally is also important. We will see that eventually, these three equations will become very important with respect
to the diffraction maxima. The three equations are known as the Laue equations, and they establish the conditions that must be satisfied for maximum scattering signal to be obtained from a crystal. We shall revisit the Laue equations shortly, but first we must know a bit more about where they come from and what they mean.

If we imagine a single unit cell—the most basic and smallest volume repeated structure—within a crystal lattice, with \( n \) atoms at positions \( \mathbf{r}_j \), where \( j = 1, 2, 3, \ldots, n \) with respect to the unit cell’s origin, we can begin to describe the scattering function for a unit cell in two dimensions. Each atom with its nucleus at position \( \mathbf{r}_j \), will scatter individually according to its individual atomic scattering factor \( f_{s,j} \), as described in eq. 2.16. If we assume the complex wave vectors for each atomic scattering function to originate at the origin of the unit cell, the phase angles for each can be described using an Argand diagram similar to that used in fig. 2.5a, with each wave vector from the origin to a given atom \( \mathbf{r}_j \) being described by:

\[
f_j = f_{s,j} e^{(2\pi i \mathbf{r}_j \cdot \mathbf{S})}
\]

where \( f_{s,j} \) is the atomic scattering function corresponding to the atom specified by \( \mathbf{r}_j \). We can then describe the total scattering for the unit cell by summing over all the scattered wave vectors. This is also known as the structure factor \( F(S) \), because it is dependent on the arrangement—or structure—of the unit cell atoms in space. The structure factor \( F(S) \) is described by:

\[
F_S = \sum_{j=1}^{n} f_{s,j} e^{(2\pi i \mathbf{r}_j \cdot \mathbf{S})}
\]

It is fairly straightforward to extend the scattering function for a single unit cell to describe the scattering by the entire crystal. In so doing, we can also gain a clearer understanding as to why the Laue equations are important in determining maximum diffraction conditions. We can assume that a given crystal has a very large number of repeated unit cells, \( n \). We can also assume that \( n \) is very large in each of three dimensions: \( n_a, n_b, n_c \). Once again, the scattering for the entire crystal is simply the summation of all the complex scattered wave vectors for all the unit cells in the crystal. We can represent each unit cell’s individual origin with respect to the crystal’s origin as being translated by \( x \cdot \mathbf{a} + y \cdot \mathbf{b} + z \cdot \mathbf{c} \), where \( x, y \) and \( z \) are whole numbers.

\[
F_S \times e^{(2\pi i x \cdot \mathbf{a} \cdot \mathbf{S})} \times e^{(2\pi i y \cdot \mathbf{b} \cdot \mathbf{S})} \times e^{(2\pi i z \cdot \mathbf{c} \cdot \mathbf{S})}
\]

It follows then that the total wave for the crystal, \( K(S) \), can be calculated by summing over all the unit cells:

\[
K_S = F_S \times \sum_{x=0}^{n_a} e^{(2\pi i x \cdot \mathbf{a} \cdot \mathbf{S})} \times \sum_{y=0}^{n_b} e^{(2\pi i y \cdot \mathbf{b} \cdot \mathbf{S})} \times \sum_{z=0}^{n_c} e^{(2\pi i z \cdot \mathbf{c} \cdot \mathbf{S})}
\]

Because in reality the number of unit cells is very, very large, we can treat these summations as limits in which \( n_a, n_b \) and \( n_c \) are approaching infinity. Here we can see the origins of the Laue conditions.
Figure 2.8. On the left, we can see that if the Laue conditions for scattering are not satisfied, there will not be sufficient scattering to produce a meaningful signal. On the Right, we can see that if the Laue conditions are satisfied, there will be significant constructive interference, and there will be excellent signal to noise. These are important factors contributing to the quality of a neutron or electron density map.

Unless the dot products in each of the exponential functions are positive or negative integers, or they equal zero, the sum $K(S)$ will go to zero because the unit cells are not scattering in phase. If we recall from eqn. 2.19 that $S \cdot a = h$, $S \cdot b = k$, and $S \cdot c = l$, we can re-write 2.23 in the form:

$$K_S = F_S \times \sum_{x=0}^{n_a} e^{(2\pi i x h)} \times \sum_{y=0}^{n_b} e^{(2\pi i y k)} \times \sum_{z=0}^{n_c} e^{(2\pi i z l)}$$

(2.24)

and we now have the Laue conditions for scattering, where $h$, $k$, and $l$ must be whole numbers, either negative or positive. The scattering vector $S$ for each unit cell is contained within the $h$, $k$, $l$ portion of the Laue equations, and also contains the phase and magnitude information for the scattered waves. The phases of these vectors must line up such that each summation enumerates to 1. They can also be equal to zero. If the Laue conditions are not satisfied, a real world crystal will not scatter with very strong signal, and a one-dimensional crystal lattice will not scatter at all, as $K(S)$ will trend toward zero. This is illustrated in fig. 2.8. Another important property of the total wave equation for a crystal that we can easily see from eqn. 2.24 is that the amplitude is proportional to the structure factor $F(S)$, and the number of unit cells in the crystal, which is represented by the sums over $[0, n_a]$, etc.

Establishing the Laue conditions gives us knowledge of the crystal lattice’s orientation, so that when diffraction does occur, we know the conditions were satisfied. What we also need is a way to relate the diffraction angle to the incident beam in a more straightforward and more elegant way. We can now begin to think of the crystal lattice as being divided into planes. Each plane contains a number of atoms. We can also begin to think of the incident and scattered electromagnetic field vectors as being reflections of one another, with the reflections originating from a set of planes spaced by the same distance as the diffracting atoms in the lattice.

This was how W. Lawrence Bragg thought about the problem when doing his Nobel Prize winning work at the beginning of the 19th century, and this is also why one of the file types we work with is
Figure 2.9. Graphical representation of Bragg’s Law. By rotating fig. a, we can get a much more elegant representation of a two atom system in fig. b. Horizontal lines represent reflection planes made up of atoms in the crystal lattice. Maximum constructive interference between two excited partial waves occurs when the path difference between $s_0$ and $s_1$ is $2 \cdot d \sin \theta$, which must also be equal to a multiple of $\lambda$.

called a reflections file to this day. If we revisit our two-atom system in fig. 2.9, we can see that by orienting the system such that the scattering planes are horizontal and the corresponding orthogonal scattering vectors are vertical, we are able to relate the scattering angle directly to the distance between scattering planes, and consequently the distance between atoms. Incident X-ray waves with electric field vector $s_0$ are now scattered by reflections from planes at an identical angle $\theta$, and reflected wave vector $s_1$. Only waves that are perfectly in phase will result in constructive interference, because the distance to the detector is very large compared to the spacing between atoms/reflection planes, making the scattered light wave vectors essentially parallel. We can see that the difference in the length of the path between $s_0$ and $s_1$ is related to the spacing of the reflection planes by the trigonometric relationship:

$$d \cdot \sin \theta$$

(2.25)

and we can also see that for the waves to be perfectly in phase, the incident and reflected wave vectors must be integer multiples of the incoming wavelength. Accounting for the reflection of the scattered waves, we arrive at:

$$2 \cdot d_{hkl} \sin \theta = n \lambda \text{ or } 2 \sin \theta = \frac{n \lambda}{d_{hkl}}$$

(2.26)

which is Bragg’s Law, where $h$, $k$ and $l$ represent Miller indices. The Laue conditions are directly related to the Bragg equation through the Miller indices. For each value of $l$ in eqn. 2.24, equivalent to $n$ in eqn. 2.26, there is a series of planes that generates a continuous set of scattered waves. For each value of $l$, the scattered waves form a set of cones in which the angle of the cone relative to the incident beam is the diffraction angle of $2\theta$. The Laue equations define the number of wavelengths that result in an observed reflection from a given crystal, while the Bragg equation defines those reflection planes with respect to the incoming X-ray beam and the observed scattering angle.
While the relationship in eqn. 2.26 appears very simple, it has some profound properties. There is a reciprocal relationship between the Bragg angle \( \theta \) and the spacing between the atoms on a reflection plane in a crystal lattice. It also means that we can determine the spacing between the reflection planes in the crystal lattice by measuring the Bragg angle, as \( \theta \) is often referred to. However, this does not mean that all atoms lie on reflection planes, nor does it mean that atoms not in reflection planes do not contribute to scattering. One of the most important and useful properties of the Bragg equation for our purposes is that it relates the scattering vectors in real space to those in reciprocal space. This relationship is direct and inverse:

\[
\frac{1}{d_{hkl}} = \frac{2 \sin \theta}{n \lambda} = d^{\ast}_{hkl}
\]  

(2.27)

where \( d^{\ast}_{hkl} \) represents a part of the crystal lattice in reciprocal space. This is very useful to us because X-ray crystallography does not allow us to directly observe the atom’s positions. We must use the diffraction data, which now includes the Bragg angles, and intensities for our diffracted waves. By working with this information in reciprocal space, we can eventually get the three-dimensional picture of the atomic positions. We can then translate this back into real space. In order to do this, we need to take a look at some of the properties of reciprocal space and some of the techniques used to translate our diffraction data using the theory we have developed up to this point. Up until now, we have been looking at how the actual experimental portion of X-ray crystallography works. This is the data that ultimately determines the quality of HyPo’s predictions. From here, our software will use a Fourier transform to convert the experimental data from real space to reciprocal space, and we will use the resulting electron density map to gather our data in conjunction with the solved protein coordinates.

Depending on the coefficients used in the Fourier transform, we can generate different types of scattering density maps that have been converted from experimental reciprocal space to real space. In X-ray crystallography, a difference density map shows the spatial distribution of the difference between the measured electron density of the crystal and the electron density explained by the current model. Conventionally, they are displayed as isosurfaces with positive density. This is electron density where there’s nothing in the model, usually corresponding to some constituent of the crystal that hasn’t been modeled, for example a ligand. Positive density is usually modeled in green, and negative density—parts of the model not backed up by electron density, indicating either that an atom has been disordered by radiation damage or that it is modeled in the wrong place—in red.

Difference density maps are usually calculated using Fourier coefficients which are the differences between the observed structure factor amplitudes from the X-ray diffraction experiment and the calculated structure factor amplitudes from the current model, using the phase from the model for both terms (since no phases are available for the observed data). The two sets of structure factors must be on the same scale. It is now normal to also include weighting terms which take into account the estimated errors in the current model. In the case of our maps, the weighting coefficient is 2 for the 2Fo-Fc maps, and 1 for
the Fo-Fc maps. The equations for these transforms are as follows:

\[ 2F_o - F_c = \rho(x, y, z) = \frac{1}{v} \sum_{hkl} (2|F_{obs}| - |F_{calc}|)e^{-2\pi i(hx + ky + lz) + i\alpha_{calc}} \]  \hspace{1cm} (2.28)

\[ F_o - F_c = \Delta \rho(x, y, z) = \frac{1}{v} \sum_{hkl} (1|F_{obs}| - |F_{calc}|)e^{-2\pi i(hx + ky + lz) + i\alpha_{calc}} \]  \hspace{1cm} (2.29)

where \( F_{observed} \) represents structure factor amplitudes measured from the experimental diffraction patterns as described above, and \( F_{calculated} \) represents structure factor amplitudes calculated from the model. \( F_o - F_c \) maps typically represent electron density that is not represented by the model. This means that experimental data might exist that is not accounted for by the model built into the electron density by the crystallographer. This is why we can use the \( F_o - F_c \) map in high quality crystal models to search for hydrogens in X-ray maps, as they are not included in the model by crystallographers due to high experimental uncertainty. The \( 2F_o - F_c \) map is the standard map that represents the structural scattering density, and is the standard map used by crystallographers to build models into experimental electron density maps. These maps and the calculations to create them are carried out in the refinement step of our process, as detailed in the methods section.
3. Methods

3.1. Initial Approach & Results. The HyPo project has been purely computational in nature up to this point. Our main goal has been the gathering and generation of a data set to demonstrate the efficacy of direct examination of scattering density maps, rather than the structural data arising from the maps. There are several reasons as to why this is desirable. First, any X-ray structural data that is not of extremely high quality will have poor quality hydrogen data\textsuperscript{57,61}. Often, it will be impossible to determine where hydrogens are, especially hydrogens of water molecules\textsuperscript{3}, because of their high degree of rotational freedom. It is for this reason that most solved crystal structures on the PDB do not include the hydrogen atoms. Second, while there do exist software programs that can guess hydrogen atom position data within the crystal structure\textsuperscript{2,17,47,49,60}, they are still just guesses, and are especially inadequate with respect to hydrogen atoms with high degrees of rotational freedom: serine hydroxyls, waters, and several others. Finally, even where maps are of sufficient quality to allow for determination of hydrogen positions, the process is very time consuming. Additionally, it is subject to the limitations of the human eye and the skill of crystallographer. Most crystallographic structures are hand built by crystallographers in a manner that involves some degree of qualitative decision-making\textsuperscript{24,44}. HyPo’s aim is to help address these issues by examining the experimental electron density data directly, quickly, and automatically.

A secondary goal in this process has been the automation of the complicated steps necessary to convert raw experimental data into usable statistical and graphical information. Because the project involves repetitive and time-consuming tasks to be performed many times on a large number of files, it is extremely beneficial to have a workflow that is as efficient and automated as possible. To this end, we have relied mainly on the crystallographic tool Phenix\textsuperscript{2}. This is an open-source package widely used by crystallographers to help process experimental data into modeled structures. Phenix includes both a graphical user interface and command-line tools. Because Phenix is written in python, it was our desire to program as much of HyPo in the python scripting language as possible.

Our initial approach to the problem was to determine how best to choose experimental data for processing. Our choice of initial data sets was extremely restricted due to the very small number of solved structures in the protein data bank that were obtained via neutron scattering. For the PDB entries with available experimental neutron data, we found the highest resolution complementary X-ray structure. By comparing data obtained from X-ray electron density maps to data from neutron density maps, we would hopefully have a meaningful way to validate our computed hydrogen position data against real experimental data.

Once our test set had been determined, we needed to prepare the raw experimental files for processing. In order to make a comparison between X-ray and neutron scattering data, it was necessary to compare different types of density maps due to the nature of the calculations being performed. For X-ray data, it was necessary to compare a PDB file without added hydrogens, so that the data generated by HyPo
Table 3.1. Table of experimental data set for testing the HyPo method, including PDB entry ID and resolution of solved neutron and X-ray structure. Sorted by resolution of neutron structure.

<table>
<thead>
<tr>
<th>Neutron PDB ID</th>
<th>Neutron Resolution (Å)</th>
<th>X-ray PDB ID</th>
<th>X-ray Resolution (Å)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4ar3</td>
<td>1.05</td>
<td>4ar6</td>
<td>0.92</td>
</tr>
<tr>
<td>4fc1</td>
<td>1.10</td>
<td>3nir</td>
<td>0.48</td>
</tr>
<tr>
<td>2xoi</td>
<td>1.50</td>
<td>2xoh</td>
<td>1.25</td>
</tr>
<tr>
<td>3hgn</td>
<td>1.65</td>
<td>3hgp</td>
<td>1.20</td>
</tr>
<tr>
<td>3a1r</td>
<td>1.70</td>
<td>2e3w</td>
<td>1.05</td>
</tr>
<tr>
<td>3kcj</td>
<td>1.80</td>
<td>1muw</td>
<td>0.86</td>
</tr>
<tr>
<td>3xwb</td>
<td>1.80</td>
<td>2nwd</td>
<td>1.00</td>
</tr>
<tr>
<td>3f45</td>
<td>1.80</td>
<td>2ov0</td>
<td>0.75</td>
</tr>
<tr>
<td>3q6</td>
<td>1.85</td>
<td>1bg7</td>
<td>1.15</td>
</tr>
<tr>
<td>4n3n</td>
<td>1.90</td>
<td>2ba</td>
<td>1.50</td>
</tr>
<tr>
<td>1eq2</td>
<td>2.00</td>
<td>1a6m</td>
<td>1.00</td>
</tr>
<tr>
<td>3u2j</td>
<td>2.00</td>
<td>1bq8</td>
<td>1.10</td>
</tr>
<tr>
<td>3kmf</td>
<td>2.00</td>
<td>2dn2</td>
<td>1.25</td>
</tr>
<tr>
<td>2vs2</td>
<td>2.00</td>
<td>2jjj</td>
<td>1.00</td>
</tr>
<tr>
<td>4bd1</td>
<td>2.00</td>
<td>2eq7</td>
<td>0.94</td>
</tr>
<tr>
<td>4jnc</td>
<td>2.00</td>
<td>3a20</td>
<td>0.88</td>
</tr>
<tr>
<td>3kks</td>
<td>2.00</td>
<td>3k34</td>
<td>1.00</td>
</tr>
<tr>
<td>4gpg</td>
<td>2.00</td>
<td>4gpg</td>
<td>1.90</td>
</tr>
<tr>
<td>4hbs</td>
<td>2.10</td>
<td>3w7y</td>
<td>0.92</td>
</tr>
<tr>
<td>3otj</td>
<td>2.15</td>
<td>4i8h</td>
<td>0.75</td>
</tr>
<tr>
<td>2y4x</td>
<td>2.20</td>
<td>1ns</td>
<td>0.94</td>
</tr>
<tr>
<td>2r24</td>
<td>2.20</td>
<td>1us0</td>
<td>0.66</td>
</tr>
<tr>
<td>2moc</td>
<td>2.20</td>
<td>2rk1</td>
<td>1.26</td>
</tr>
<tr>
<td>3byc</td>
<td>2.20</td>
<td>3o4p</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Represented only experimental scattering data. To this end, we wanted to examine the difference density \((F_{\text{obs}} - F_{\text{calc}})\) map, in order to leverage the fact that hydrogen does not readily scatter X-rays. By applying the HyPo method, we would be able to obtain the scattering density in a 1-D ring about each hydrogen-containing carbon. The peaks from this data would correspond to areas of higher probability of hydrogen atom placement. A similar approach was used for the neutron scattering maps.

The first step in our workflow involved running a refinement on the CIF (Crystallographic Information File). Each CIF contains nearly all the information from a given crystallographic study in a single ASCII file, but must be converted into a format usable by Phenix. To this end, a series of scripts to handle these conversions was written in the cross-platform python scripting language. The script was designed to be pointed to a directory and walked through every sub-directory, returning only those files which were identified as CIF’s. The scripts then standardized all the file names for consistency, and generated and executed custom batch files pointing to the absolute paths of each of the located CIF’s, converting them to MTZ reflection files using python’s command-line tool cif_as_mtz. An example of the shortest and most basic of our first round of python scripts can be seen in the 27 lines of code below:

```python
#!/usr/bin/python
import sys
import os
```
Once the files had been converted to mtz format, they were refined, along with their corresponding pdb structure files, using Phenix’s command line version of phenix_refine. Once the mtz and pdb files were refined by Phenix, they were Fourier transformed into various density maps using the ccp4_ff command line tool. The difference density map created by this final step, along with the refined pdb file were the ones we wanted to examine using the HyPo process. This process created a total of around 12 or so files for each input file. These were then grouped into sub-folders that were named using the four-digit pdb entry code for each corresponding file set. HyPo was then run on each folder, generating a custom batch script with absolute path names for each set of files, and subsequently HyPo would generate the necessary rotameric electron density data for the hydrogen positions using bond lengths of 0.9-1.5Å. The main program comprising the core of HyPo uses some command line tools from Chimera, which was developed at UCSF, in order to read and process the electron density around the protein side chains. This entire process of data acquisition was completely automated, with the overall goals of speed and flexibility. We were easily able to generate large amounts of raw data easily and quickly.
**Figure 3.1.** In fig. 3.2a and b, we can see excellent global and local alignment of the structures. The Fo-Fc difference map (in red) from the X-ray data aligns well with the 2Fo-Fc difference map (green) from the neutron data. We would expect this pair of structures (4AR6 & 4AR3) to yield excellent results via the HyPo method. In Fig. 4.2c and d, we can see the results of good global alignment and poor local alignment. These small differences are likely enough to break our requirement of a match existing to within 30 torsional degrees, and result in poor overall match rates.

### 3.2. Secondary Approach & Results

Unfortunately, because of the poor overlap of many of our X-ray/Neutron comparison pairs, our initial results were less than satisfactory, yielding a maximum of 7:100 hydrogens matching between neutron and X-ray structures. It was determined that this was mainly caused by poor alignment of the proteins and their corresponding electron density maps. It was decided that in order to better control for the many unforeseen variables we were encountering, it would be best to use only proteins that had >90% sequence alignment. Even more preferable would be if the data collected for neutron and X-ray structures were from the same research group and even the same experiment. It is often the case where a research group will produce a matching set of structures, as the two data collection methods are very complimentary, and because having an extremely high quality crystal for neutron experimental data collection lends itself naturally to collection of an extremely high quality X-ray experimental data set. One of the best examples of this can be seen in a data set from Cuypers, et. al. in their Protein Data Bank entries: *X-ray crystallographic structure of the reduced form...*
Table 3.2. Secondary data set, including PDB entry ID’s for 18 matching neutron and X-ray experimental data sets. These were chosen because of similar sequence length and experimental conditions, in order to try and minimize the number of differences between the two data sets for direct comparison of neutron and X-ray experimental data.

<table>
<thead>
<tr>
<th>Neutron ID</th>
<th>No. Residues</th>
<th>Resolution (Å)</th>
<th>Xray ID</th>
<th>No. Residues</th>
<th>Resolution (Å)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1cq2</td>
<td>153</td>
<td>2.00</td>
<td>1naz</td>
<td>154</td>
<td>1.04</td>
</tr>
<tr>
<td>3u2j</td>
<td>117</td>
<td>2.00</td>
<td>3u2i</td>
<td>117</td>
<td>1.7</td>
</tr>
<tr>
<td>3q6j</td>
<td>66</td>
<td>1.85</td>
<td>1kg7</td>
<td>66</td>
<td>1.15</td>
</tr>
<tr>
<td>4ar3</td>
<td>54</td>
<td>1.27</td>
<td>4ar6</td>
<td>54</td>
<td>1.5</td>
</tr>
<tr>
<td>3ucj</td>
<td>388</td>
<td>1.80</td>
<td>3kbn</td>
<td>388</td>
<td>1.53</td>
</tr>
<tr>
<td>2yq4</td>
<td>237</td>
<td>2.20</td>
<td>1nls</td>
<td>237</td>
<td>0.94</td>
</tr>
<tr>
<td>2mqj</td>
<td>159</td>
<td>2.17</td>
<td>4psy</td>
<td>159</td>
<td>0.85</td>
</tr>
<tr>
<td>2xoi</td>
<td>125</td>
<td>1.50</td>
<td>2nol</td>
<td>125</td>
<td>1.25</td>
</tr>
<tr>
<td>4bd1</td>
<td>261</td>
<td>2.00</td>
<td>4bd0</td>
<td>261</td>
<td>1.21</td>
</tr>
<tr>
<td>3hgn</td>
<td>240</td>
<td>1.65</td>
<td>3hgp</td>
<td>240</td>
<td>1.2</td>
</tr>
<tr>
<td>3kex</td>
<td>260</td>
<td>3.20</td>
<td>2lll</td>
<td>259</td>
<td>1.05</td>
</tr>
<tr>
<td>4fc1</td>
<td>46</td>
<td>1.10</td>
<td>1ejg</td>
<td>48</td>
<td>0.54</td>
</tr>
<tr>
<td>1ntp</td>
<td>223</td>
<td>1.80</td>
<td>4il8h</td>
<td>223</td>
<td>0.75</td>
</tr>
<tr>
<td>4qcd</td>
<td>248</td>
<td>1.55</td>
<td>2d1e</td>
<td>248</td>
<td>1.51</td>
</tr>
<tr>
<td>4y0j</td>
<td>258</td>
<td>2.00</td>
<td>3kcs3</td>
<td>260</td>
<td>0.9</td>
</tr>
<tr>
<td>4ndq</td>
<td>159</td>
<td>1.60</td>
<td>4psy</td>
<td>159</td>
<td>0.85</td>
</tr>
<tr>
<td>4ny6</td>
<td>65</td>
<td>1.05</td>
<td>1kg7</td>
<td>66</td>
<td>1.15</td>
</tr>
</tbody>
</table>

perdeuterated Pyrococcus furiosus rubredoxin at 295 K (in quartz capillary) to 0.92 Angstroms resolution\textsuperscript{16} and Near-atomic resolution neutron crystallography on the oxidised form perdeuterated Pyrococcus furiosus rubredoxin\textsuperscript{16}. These entries were created by the same research group, and at the same time, being discussed in the same publication. Consequently, the two sets of data matched extremely well and aligned with excellent results, as can be seen in fig. 3.2.

In addition to choosing an entirely new data set, it was also decided to change the processing algorithm and data preparation of the files from processing by HyPo. Unfortunately, despite being the de facto standard for sharing experimental data, the CIF file format has a tendency to be very inconsistent with respect to the type and amount of information it contains. As these files, in addition to their corresponding coordinate files, are the basis of our entire project, we needed a way that yielded more consistent results across the entire data set. Additionally, many of the solved data sets we employed in our second test set had non-standard ligands and coordinated metals within the protein, which made preparation and refinement of each structure a unique situation. Because of these factors, it was necessary to prepare and refine the majority of our second data set files manually. Also, we were forced to throw out all of our original python scripts and start from scratch, re-writing them to better suit the new approach.

As a means of testing the efficacy of our automated preparation and refinement script, the automated process was run on the second data set, and the results were compared to those obtained manually. Because of the many variables in the protein structures which required manual attention, the automated refinement success rate was only slightly above 50%. This could easily be improved with the full automation of the algorithm pictured in fig. 3.3. Our manual workflow consisted of first preparing the CIF
files for refinement by generating any ligand restraints and metal coordination information using Phenix ReadySet. A script could be written to replace the manual workflow, as ReadySet’s default options are usually adequate for the majority of files. Using a script that calls ReadySet prior to auto-refinement would likely increase the success rate of our batch refinement to 80-90% successful. Refinement is a necessary step in our processing algorithm for two reasons. First, it allows us to generate our MTZ reflections file, transforming the experimental reflections to electron or neutron density maps as described in section 2. Second, it ensures that the PDB model we have fits within the calculated electron density as well as possible. After all files were refined successfully, the resulting MTZ reflections files needed to be prepared for comparison by HyPo. In order to accomplish this, a newer tool in the Phenix suite was used: superpose maps.

Superpose maps is a utility for transforming two maps to the same frame of reference, allowing direct comparison of molecular features in electron density. The program actually performs a superposition of the PDB coordinate files, then reorients the associated maps to follow them. It should be noted that superpose maps uses only one chain for the molecular superposition, and attempts to determine the selected chain automatically. We found this worked well by default, and did not need to change the default selections. Input models are transformed, such that all coordinates are positive, and once the maps have been superposed, there is no way to recover the initial frames of reference. Additionally, maps
1. Using Chimera’s command-line tools, HyPo ‘walks’ along the protein backbone and samples torsion angles around side-chain carbon atoms where hydrogens are expected to be. HyPo returns a list of electron density vs. torsion angle for a range of bond lengths and bond angles. All parameters are tunable.

2. HyPo dumps all data into a number of different files. A massive text file containing torsion vs. density lists for every residue in the sampled protein is converted to CSV format to leverage python’s CSV tools for data analysis.

3. The converted CSV files contain a great deal of redundant data. The extra data are truncated, and the data in the files are condensed. In addition, all electron density values below a designated threshold are removed, including negative values. For all our experiments, this value was chosen to be 0.3 sigma.

4. The CSV files initially output by HyPo contain a theta angle column for every density column. This format was chosen for graphing purposes. These columns are now removed, leaving a single theta column in each CSV file.

5. All residues in the CSV file are now grouped together in alphabetical and numeric order.

6. A new CSV file is created, containing only the highest electron density value and its corresponding theta angle for each residue.

7. The number of hydrogens detected for each protein are calculated based on peaks identified above the 0.3 sigma threshold.

8. All folders and subfolders are searched, and xray structures are matched to their corresponding neutron structures automatically by name. A new folder is created with these final peak-picked CSV files renamed for comparison.

9. All matching xray and neutron files, at each bond length and bond angle, are compared using electron density and torsion angle data. Matches are returned for any torsion angle that corresponds to within 30 degrees. Summary of matches is appended to filenames.

**Figure 3.3.** Processing algorithm for HyPo data. This entire process is completely automated and able to handle thousands of data files in a very short amount of time.

are sampled along a simple grid, and reorientation of the map requires re-sampling for the new grid. This requires interpolation of the new grid values, and some accuracy is lost in the process, meaning the grid point values are not precise. Maps were superposed such that the X-ray structure and its refined $F_{obs} - F_{calc}$ map was superposed with the Neutron structure and its corresponding $2F_{obs} - F_{calc}$ map. Some examples of superposed structure and map files can be seen in *fig. 3.2*. It is important to note that the incorporation of superpose maps made a dramatic change in our workflow. Not only did it allow us to generate meaningful data sets that were prepared for examination by HyPo, but it also allowed us to stop using any of the command line tools included in the CCP4\textsuperscript{60} distribution. This meant that our entire preparation workflow was based on tools provided in the Phenix\textsuperscript{2} distribution. This is highly desirable, as Phenix is already written almost entirely in python, as is HyPo.

Once all the chosen input files had been successfully downloaded, prepared, refined, and superposed, we were ready to run our program on the prepared files to collect our data. Figure 3.4 outlines the current HyPo data collection and processing algorithm. The first step of the HyPo algorithm is also the most computationally intensive. This is also the step that utilizes Chimera to help read in electron density maps and to relate the values to the torsion angle about a given carbon on a side chain residue. Eventually, we would like to completely re-write the sampling algorithm, so that this step of the process...
In fig. 3.5a, we can see the definition of the torsional angle chi1 for ALA. HyPo uses various rules for determining the chi angle for different residues. In fig. 3.5b, we can see the effects of varying the bond length and bond angle used to define the torsional sampling of electron density, and the resulting data collection. We can take the beta carbon from fig. 3.5a as being represented below the sampling cone.

HyPo does not depend on Chimera at all. Currently, HyPo reads electron density in rings about carbons on amino acid side chains at various bond lengths and bond angles, and returns a list of residues as torsion angle vs. electron density, with torsion being incremented in steps of 10 degrees. This process was automated using a complicated script that searches the contents of a folder and all the subfolders contained within that folder. HyPo has been written with automation and ease of use in mind. To this effect, HyPo has only a single configuration file that needs to be edited before running the program on any system. In addition, there is a single environment variable that needs to be set prior to running the script. The configuration files contain the paths to various parts of the script that run in sequence. The environment variable HYPO_WORKING_DIR points to the working directory, which contains all the pre-processed files for HyPo to examine. The first step in the process finds all instances of superposed PDB coordinate files and electron density maps within the folder indicated by the working directory environment variable. An example of one of the many scripts written to automate the HyPo data collection and refinement process is provided below. HyPo currently consists of about 23 such scripts, usually run in sequence. Parallelization of these has proven unnecessary, as calculation time has not been a major time constraint thus far.

```
1 #!/usr/bin/python
2 import sys
3 import os
4 import fnmatch
5 import time
6 import dateutil
7 import subprocess
8 import shutil
9
10 srcpath = os.environ['HYPO_WORKING_DIR']
11 hypodir = os.environ['HYPO_AUTO_PATH']

12 def progress(count, total, suffix=''):  
13     bar_len = 60  
14
```
filled_len = int(round(bar_len * count / float(total)))
percents = round(100.0 * count / float(total), 1)
bar = '=' * filled_len + '-' * (bar_len - filled_len)
sys.stdout.write('{}%{}...
' % (bar, percents, suffix))
sys.stdout.flush()

def id_gen():  # Generates format USERNAME_DATE to append to files
    i = datetime.datetime.now()
    long_date = str(i.isoformat())
    date = long_date[:10]
    unix_id = os.getenv("LOGNAME")
    return str(date) + '_' + unix_id + '_'

map_list = []
map_count = 0
map_reset = 0
ringer_sort = 0
bond_length = 0.90
bond_angle = 90.00
bl_str = '%.2f' % bond_length
job_name = id_gen()

for root, dir, files in os.walk(srcpath):
    for items in fnmatch.filter(files, '*superposed*.map'):
        map_list.append(os.path.join(root, items))
        map_count = map_count + 1
        map_reset = map_count
        counter_total = map_count
while bond_angle < 130.00:
    map_count = map_reset
    bond_length = 0.90
    bl_str = '%.2f' % bond_length
    if bond_angle == 90.00:
        bl_ba = '%.3f' % bond_angle
    else:
        bl_ba = '%.2f' % bond_angle
    os.environ["BOND_ANGLE"] = bl_ba
    while bond_length < 1.60:
        os.environ["BOND_LENGTH"] = bl_str
        while map_count > 0:
            status = counter_total - map_count
            progress(status, counter_total, suffix='complete.'
            list_extract = os.path.split(map_list[map_count - 1])
            directory_name = list_extract[0] + '/'
            pdb_split = os.path.splitext(map_list[map_count - 1])
            pdb_name = pdb_split[0] + '.pdb'
map_name = directory_name + list_extract[1]
map_file = list_extract[1]
pdb_code = map_file[:4]
ba_directory = directory_name + bl_ba
bl_directory = ba_directory + '/' + bl_str
if not os.path.exists(ba_directory):
    os.makedirs(ba_directory)

ringer_input_file = (
    'map_name ' + map_name + '
' +
    'pdb_name ' + pdb_name + '
' +
    'skip_multi_conf on
' +
    'chi_sample_degree 10
' +
    'aton_sample_type dynamic
' +
    'lower_sigma_cutoff 0.30
' +
    'write_plot on
' +
    'write_peak_list on
' +
    'write_chi2chi1 on
')

inputfilename = directory_name + job_name + 'ringer_' + pdb_code + '_' + bl_str + 'A_' + bl_ba + 'deg.in'
file = open(inputfilename, 'w')
file.write(ringer_input_file)
file.close()

log_name = job_name + 'ringer_' + bl_str + '.log'
logfile = open(log_name, 'w')

hypo = hypodir + '/ringer/ringer'
ringer_run = subprocess.Popen([hypo, '-i ' + inputfilename], stdout=logfile, stderr=logfile)
ringer_run.wait()

file.close()

if not os.path.exists(bl_directory):
    os.makedirs(bl_directory)

move_command = 'mv *ringer* ' + bl_directory

if bond_length < 1.6:
    print '\nProcessing at ' + bl_str + ' & bond length.'
    bond_angle += 10.00

print '\nRun complete!'
very easily and quickly. We were able to process 36 hand-prepared proteins using HyPo, to produce over
1,400 data files in a matter of minutes. The number of files grows very quickly because each file is being
processed at bond angles from 90-120 degrees at each bond length from 0.9 to 1.5 angstroms. Each of
these runs is nested with the appropriately named folders. This means that a folder for each bond angle
is nested within the folders for each bond length. For this reason, a few starting files can quickly balloon
into hundreds, or even thousands of data files. This is the reason that HyPo needed to be automated.
For each file in the many hundreds, there are hundreds of individual data sets for residues and even
multiple data sets for residues with multiple chi torsion angles. By automating this task, HyPo is able
to peak pick maximum electron density for thousands of hydrogens in a matter of seconds. This could
potentially save crystallographers tremendous amounts of time, as well as possibly providing a means of
verifying current computational techniques using purely experimental data.

These files are then paired up and a list is generated, containing all the absolute paths to these files.
The program then generates input files for each pair of maps so that they can be processed with HyPo.
The arguments given to HyPo are completely tunable and customizable. HyPo has been programmed
to sample electron density in rings about certain carbon atoms in protein side chains. For different
residues, this means that there will be different torsional angles, and often, multiple torsion angles. The
sampling mechanism HyPo uses is detailed in fig. 3.5a. As seen in fig. 3.5b HyPo samples in rings that
have various radii and locations in space with respect to the carbon being sampled. This is the result of
HyPo’s sampling ring being adjusted by varying both the bond length and bond angle arguments used
to create the torsional sampling ring.

The tunable parameters for HyPo are as follows:

- Path to the PDB coordinate file must be defined:
  - pdb_name [None] (string) map_name [None] (string)

- The absolute path to the model and map files must be defined.
  - pdb_file_location [./] (string) map_file_location [./] (string)

- Side Chain Perception
  - Each chi angle is defined based on the modeled side chains. Chi 1 angles are built starting
    at the backbone nitrogen. For residues with alternate conformations included in the model,
    HyPo either samples using the higher-occupancy conformation (off) or excludes the side
    chain from the analysis (on).
  - skip_multi_conf [ON] (off.on)

- The dihedral angles are systematically sampled by a changeable absolute torsion angle
  - chi_sample_degree [10] (integer)

- The electron density at each point is extrapolated from the map electron density in Cartesian
  space by trilinear interpolation. The electron density (in units of sigma) is plotted versus the
sampled chi angles for peak identification. Peaks are identified as the maxima in the plots of sigma vs. chi angle above the user-defined lower cutoff in units of sigma.

- lower_sigma_cutoff [0.3] (float)

- Peaks also can be restricted below an upper threshold of 0.8 sigma. This filter helps to focus on peaks that represent low population conformations below the standard 1.0 sigma noise cutoff.
  - upper_sigma_cutoff [off] (on,off)

- Verbose Output Options
  - The user has the option to write out various verbose portions of data from the calculation.
    - The prefix of these files can be modified
      - verbose_outfile_prefix [HyPo] (string)
    - The raw data to generate the sigma vs chi angle plots can be printed for amino acid types.
      - write_sigma_plot [ON] (on,off)

- HyPo can also print out the list of peaks identified from the sigma vs chi angle plots above the user-defined cutoff. Because HyPo identifies alternate conformations based on density peaks beyond the primary peak, only unbranched torsion angles are chosen for each chi angle category. The identified peaks are divided into chi angle-based categories:
  - chi 1 = Ser, Gln, Asn, Glu, Asp, Arg, Lys, Met, Cys, and Leu ring = Phe, Tyr, Trp, and His
  - chi 2 = Gln, Glu, Arg, Lys, Met, and Ile
  - chi 3 = Lys, Arg, and Met
  - chi 4 = Lys and Arg
  - write_peak_list [ON] (on,off)

The initial output is in an unformatted text file format, which is not very useful for data parsing. Additionally, the raw text file contains a great deal of information that is redundant and unnecessary. The next step in the processing algorithm involves taking the original text files and converting them to CSV format. CSV stands for Comma Separated Values, and it is a very common, compatible and widely used file format for storing numeric data. CSV files are directly compatible with many data processing programs, most notably Microsoft Excel. They can also be used in Sigma Plot, Origin Pro, and many others. This step of the program takes the data and converts it from text file to CSV format automatically, and renames all the files accordingly. All paths are preserved within the working folder. Next, HyPo processes each CSV individually by removing all redundant data, and also by removing all electron density values below the 0.3 threshold. This includes negative values. The entire process is completely non-destructive, meaning the original input files are all preserved, and the newly edited files are all named to reflect the changes. Up to this point, the CSV files contained an individual torsion angle column for each residue. This was initially made this way for the purposes of graphing each individual
residue. The removal of redundant data included the removal of all theta columns save the first. This allows for easier processing of the data in our remaining steps. In future project goals, when creating the plotting script, we can come back to the file created before truncation, so that the creation of multiple plots for different residues will be much easier. This is one of the main reasons it was decided to make the entire process non-destructive. It results in many more files being created, but retains much more data.

The processed CSV files thus far have all the residues for each protein at each bond angle and each bond length. The residues are in order according to the sequence of the protein, being numbered from the beginning and continuing until the end of the primary backbone. We found that for further processing, it would be easier to have residues grouped by type. The next step of HyPo performs this action, editing each individual residue in the CSV file so that they are re-ordered and grouped by residue type. This allows for plotting/calculation of average density peak values by residue type. The files are changed and re-named accordingly. All path names and original files are preserved. The next step involves the creation of our peak-picked files. HyPo goes through each file and returns only the highest electron density value at each bond length and bond angle for each residue in each protein. A new file is created with only two values per residue: electron density and torsion angle. These are the files we will be using in order to make our comparison between the peaks in neutron density maps and X-ray density maps. These new files are copied into a newly created folder, with the date, username and “compare” appended to the folder name for easy location at a later time and date. The files are named according to their bond length and bond angle sampling by HyPo, and are named for being either neutron or X-ray experimental data. Each pair of corresponding neutron and X-ray peak-picked CSV files are then directly compared for density peaks matching to within 30 torsional degrees. If there is a match, a boolean TRUE value is returned. If there is no match, a boolean FALSE value is returned. The number of matches vs. the total number of peaks is appended to the file name of each new CSV file created for each protein at each bond length and bond angle. Finally, a script goes through each CSV file and determines the total number of hydrogens detected for each protein at each bond length and bond angle for the X-ray structures and for the neutron structures. These data are also appended to a summary CSV that contains statistics for all the detected hydrogen atoms by residue type.
Figure 4.1. In figures 4.1a-4.1d, electron scattering density maps for the ser-46 hydroxyl in perdeuterated Pyrococcus furiosus rubredoxin show the difference density maps at increasingly stringent sigma cutoffs. In figure 4.1e, the neutron diffraction data provides the real conformation. In figure 4.1f, we can see the results of the neutron structure (4ar3) and experimental data (light grey mesh) superimposed with the X-ray (4ar6) difference density data (red mesh). Note that in figures 4.1a-d, no hydrogen atoms are shown, whereas in 4.1e and 4.1f, the hydrogen atoms have been included, based on the neutron data.

4. Results & Discussion

HyPo gives electron density data for a number of 1D rings of pre-determined radius about different atoms. Because we are examining the difference density map in the case of X-rays, a higher positive density correlates to a higher probability of there being a hydrogen atom attached to the rotameric position about that carbon atom correlating to the electron density peak. The best means of doing this before HyPo to examine the residues of a given protein individually. What this involved was to calculate both the $F_{\text{obs}} - F_{\text{calc}}$ and the $2F_{\text{obs}} - F_{\text{calc}}$ difference density maps and perform a visual inspection of the residue in question in coot, pymol, or chimera. HyPo has the potential to greatly speed this process, and automate a great deal of the work necessary to examine these features of interest. Additionally, this method of difference density map comparison will only work for crystal structures of extremely high quality, usually better than 1.0 angstrom resolution. HyPo can eliminate the need for visual inspection
Figure 4.2. Fig. 4.2a shows that there is a weak correlation between neutron resolution and number of hydrogen peaks detected. This relationship is not really statistically significant. Fig. 4.2b shows a reasonable correlation between x-ray resolution and HyPo’s sensitivity to hydrogen electron density peaks. This is to be expected, as typically signal to noise increases at higher (poorer) resolutions.

of the maps, which will make the process much less subjective. In addition, the resolution threshold will be dramatically increased, as the human eye typically needs to look at maps at the 1.0-2.0 sigma cutoff in order to distinguish between signal to noise. HyPo is sensitive to sigma levels down to 0.3, with a sigma cutoff of 0 representing the average electron density over then entire map. For instance, when looking at the serine hydroxyl atoms in the structure of rubredoxin at very high resolution (4ar6)\textsuperscript{16}, HyPo gives the 1D plot of difference density vs. torsion angle of our sampling ring. After this, visual inspection of the model and map can determine the feasibility of the data.
Figure 4.3. Fig. 4.3 shows the correlation between hydrogen detection rate, x-ray, and neutron resolution. We can see that there is a reasonable negative correlation between increasing resolution and number of hydrogen peaks detected, but the correlation between neutron resolution and average number of peaks detected is not very significant statistically.

The completely raw data analyzed by HyPo often appears differently than that shown by the graphical map. This is the result of the complete analysis of available data by the program. In order to get a more confident idea of acceptable prediction, a statistical cutoff must be established for the purposes of data analysis. This is where HyPo really shines. HyPo is able to bypass the signal to noise cutoff typically required by the human eye, and thereby can increase sensitivity, and make maps of lower resolution much more useful. In fig. 4.1, there are different levels of sigma cutoff shown, and we can see the importance of establishing acceptable cutoff limits. By examining the electron scattering density shown in fig. 4.1, we can see that the secondary peaks can be eliminated by visual inspection only after a given sigma cutoff. HyPo is able to sample to a much lower sigma cutoff than that currently employed by crystallographers using visual inspection of difference density maps. Typically, a crystallographer using traditional identification techniques will be able to find only about 50% of hydrogens in a given crystal structure of very high quality (<1.0Å). In addition, hydrogen detection must be done by hand, one residue at a time, and can be very challenging and time-consuming. HyPo seeks to make this process much faster and more automated, and also to raise the resolution cutoff for using this technique, in order to make more structures from the PDB available for analysis using the experimental data. HyPo was able to detect hydrogens in our test data set to 1.7 angstroms, raising the number of structures available via the PDB from 488 at 1.0 angstrom, to 5254 at 1.7 angstrom. At 1.7 angstroms, HyPo was able to detect 67% of hydrogen peaks as compared to the neutron structure. This suggests that the upper limit of HyPo’s useful sensitivity may lie outside the range covered by our test set. In addition, the detection rate trend seems to increase with x-ray resolution, but be independent of neutron resolution.
Figure 4.4. Fig. 4.4a shows a plot of scattering density vs. torsion angle for X-ray (blue, 1A6M) and neutron (orange, 1CQ1) data for the ser-46 hydroxyl of oxy-myoglobin. Note that HyPo places the main peak from the X-ray data about 20 degrees past the peak predicted from the neutron scattering data. Fig. 4.4b shows a plot of scattering density vs. torsion angle for X-ray (orange, 4AR6) and neutron (blue, 4AR3) for alanine in perdeuterated Pyrococcus furiosus rubredoxin. Interestingly, the peaks from both structures align fairly well, suggesting perhaps a preferred conformation for even a freely rotating methyl.

This could be because the average of the neutron structures included perdeuterated proteins, in which the scattering effects of hydrogens are clear and the signal strength is similar to that of carbon, as well as non perdeuterated proteins, that have very high incoherent scattering of neutrons, resulting in a negative signal. A further study using a set of perdeuterated proteins and a set of non perdeuterated proteins would likely show that resolution dependence is of greater importance for non perdeuterated structures, as the hydrogen atoms in these structures resemble those in X-ray structures more closely than those in perdeuterated proteins. For our data set, the average match rate was (70 ± 10%). These are promising
data that HyPo will be able to increase detection limits for hydrogens in crystallographic data sets, and detect hydrogen peaks much more quickly and accurately than current conventional methods.

In addition to detecting hydrogen peaks, it is extremely important to locate them in space accurately. Our data rely on leveraging torsional angle vs. electron density in order to determine locations of peak hydrogen density about side chain carbons. Our goal was to directly compare density peak data for hydrogen atoms in both neutron and X-ray scattering experimental data for matched protein sets. We are able to see the results of these data most easily by plotting the torsional angle vs. the scattering density for a given carbon on a given side chain residue. An example of these results can be seen in fig. 4.4. In fig. 4.4a, HyPo’s prediction is off by nearly 20 degrees with respect to the torsion angle of the serine hydrogen, this translates to roughly 0.3 angstroms linear distance, assuming both neutron and x-ray peaks lie on the same 2-dimensional circle. Despite controlling for as many variables as possible in our data, we were still not able to achieve any extremely high match rates between neutron and X-ray pairs. Our data ranged from (16.18±2.02)% to (30.13±0.44)% of average hydrogen scattering density peaks matched in X-ray and neutron structures to within 30 torsional degrees or 0.5 angstroms. Our highest match rate was from the data pair 3HGP and 3HGN, which were sister structures from the study *Structure of porcine pancreatic elastase complexed with a potent peptidyl inhibitor FR130180 determined by high resolution crystallography*. These had resolutions of 0.94 angstroms for the X-ray structure and 1.65 angstroms for the neutron structure. Our lowest match rate was for the pair 4PDJ and 4PSY, which were both structures of *E.coli Dihydrofolate Reductase*, but collected by different groups in different studies. The X-ray structure 4PSY had a resolution of 0.85 angstroms, while 4PDJ had a resolution of 1.6 angstroms. Both sets of data had very similar resolutions, which suggests that factors other than resolution are contributing to the match rate of the rotameric hydrogen density peak locations. Despite these two extreme cases, there was a general correlation between both neutron and X-ray resolution and peak density match rate between structures. There did not appear to be any correlation between variation of bond length and bond angle with respect to detection sensitivity and accuracy. Across all bond lengths and bond angles sampled by HyPo, the results were identical. This does not seem to correlate well with visual inspection of the electron density maps, where clear peaks emerged and aligned at higher sigma cutoffs (fig. 3.2a & b). It could be that many of the hydrogen peaks detected were only slightly above the 0.3 sigma cutoff, meaning they were only slightly more than random noise. This would explain the average match rate of (24.39±1.45)%, as many cases exist where completely random matches occur at around a rate of roughly 25%. As to the causes of this poor match rate of peaks, there could be any number of possible explanations. Firstly, as can be seen in fig. 3.2c, there were many cases for which the superpose maps technique used to align our structures proved to be inadequate. Unfortunately, superpose maps is written to perform a global alignment only, and in maximizing the alignment quality, many times the local alignment of residues suffers as the result. This
Figure 4.5. Fig. 4.5a shows a small correlation between neutron resolution and peak density match rate by HyPo. Fig. 4.5b shows a small correlation between X-ray resolution and peak density match rate. Neither correlation is statistically significant enough to warrant a confident prediction.
Figure 4.6. Correlation between bond length and positive match rate for a hydrogen density peak between related neutron and X-ray structures, showing clearly that there was no statistically significant trend of any kind. It should be noted that the bond angle figure was nearly identical, and so omitted.

could be enough of a perturbation such that hydrogen peaks no longer fall within our requirement of 30 torsional degrees for a positive match. Further, one of the goals of our paper was to help determine positions of rotamERICally unhindered hydrogens. This could also be causing a huge problem in that different crystals will likely adopt different low energy minima, and an alanine or serine hydrogen in one structure may be in a completely different rotameric position than for its 'matching' structure. The method was tested on a control to assure that it was sampling the scattering density correctly and correctly searching for matches. We superposed an identical structure to itself, using the $F_{\text{obs}} - F_{\text{calc}}$ and $2F_{\text{obs}} - F_{\text{calc}}$ maps from the same structure. We then processed these through HyPo, and got a 100% match rate at all bond lengths and bond angles. This suggests that the poor match rate is likely coming from some experimental variable we have yet to control or account for.

We also examined HyPo’s sensitivity and successful match rate with respect to individual amino acids. Normalizing for the number of structures by taking percentage matches, we found alanine to be most often successfully matched between x-ray and neutron structures, while methionine had the lowest success rate. This was an interesting result in that it suggests that despite having a great deal of rotameric freedom, alanine my possibly have a globally similar preferred rotameric conformation across multiple protein structures when it is crystallized and adopts its energetically preferred conformation. It would be interesting to perform further study to pursue this.
Figure 4.7. Fig. 4.7a shows the total number of successfully identified hydrogen density peaks for the entire data set, grouped by residue. The orange represents the total number of identified peaks, and the blue represents the total number of successfully matched peaks. It should be noted that this was across all bond lengths and bond angles, as there was no single bond length/bond angle combination that yielded superior results. Fig. 4.7b shows the percentage of successful matches by residue type.

5. Summary, Future Plans & Goals

In its current state, HyPo is incomplete. Our overall goal was to create an automated package for hydrogen position determination directly from experimental data. Currently, data generation has been programmed in python to be fully automatic, meaning that data generation is restricted only by the size of the data set provided for processing. Statistical analysis of HyPo’s results has been fully automated. We have shown the sigma cutoff for HyPo’s identification of hydrogens, classified by residue, as well as the total number of accurately predicted hydrogen atoms in difference density data as compared to neutron data. A comparison has been made between different types of side chains, so as to evaluate the
sensitivity and accuracy of the method between amino groups. We have increased the sensitivity well beyond that currently available to crystallographers using traditional visual inspection methods. We have also greatly sped up the data acquisition process.

In order to be truly useful, Hypo needs to be fully automatic, and needs to be able to evaluate the quality of its predictions. This is relatively straightforward when data are available allowing us to compare X-ray structures to neutron structures, but becomes more challenging as neutron data are either unavailable or as the resolution of crystal structures becomes lower. HyPo also needs to work and work well for X-ray structures with poor resolutions in order to truly become versatile and valuable. Another feature that we are planning to implement in HyPo is the re-writing of the main program, so that it is not dependent on Chimera. Rather than sampling discrete, pre-determined bond lengths about a 1D torsion angle, we would rather sample in a cone- or sphere-shaped manner, so as to obtain maximum experimental data. As of now, the resolution cutoff is around 1.7Å, which does not make the method useful for the majority of available crystal structures. However, as seen in our data, it is highly likely that HyPo’s sensitivity cutoff is significantly higher. We could plausibly push the resolution sensitivity limit to 2.0 angstroms or greater. We are planning ways in which to extend the applicability of HyPo to a greater set of experimental data. HyPo can also be a very useful technique for generation of parameterizations for predictions in data sets that are outside of operating parameters for HyPo. By using HyPo to generate a large training set from a variety of data, the information can be combined with other methods so that weak and incomplete experimental data can be evaluated with greater accuracy and confidence. The unique feature that HyPo incorporates is its ability to extract information directly from experimental scattering data. Once HyPo’s ability to predict and evaluate hydrogen positions from a greater pool of experimental data has been established, we plan to implement a web server that can take queries from other research groups and process the results quickly and automatically, custom-tailored for specific needs, with an ease-of-use approach. In doing this, we can both aid other research groups in helping elucidate hydrogen location data and also continuously build and refine our parametrization training set.
REFERENCES


3. Pavel Afonine, *Hydrogens , X-rays , neutrons and all that . . .


56. Dale Tronrud, *The Wonderful World of Maps What Kinds of Maps are There ?


