TOPICS IN STRUCTURE DETERMINATION OF SUBMICRON SIZED OBJECTS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Hyung Joo Park

August 2015

TOPICS IN STRUCTURE DETERMINATION OF SUBMICRON SIZED OBJECTS

Hyung Joo Park, Ph.D.

Cornell University 2015

This dissertation documents work done in two different fields, both of which are related through their studies of objects measured on the order of nanometers. The first part discusses efforts at automating image reconstruction of objects studied through coherent X-ray imaging experiments. Data collected at the Linac Coherent Light Source are used to reconstruct hundreds of images of soot particles in flight. The second part discusses efforts at protein structure prediction and generating collections of building blocks necessary for the prediction algorithm. Data from the Protein Data Bank are processed to generate large numbers of building blocks, which are then used to predict protein structures with some initial success.

BIOGRAPHICAL SKETCH

Hyung Joo Park was born in 1986 in Seoul, South Korea. During his childhood, his parents moved the family to various foreign countries such as Bangladesh and Costa Rica before finally settling in Guatemala. There, he attended Colegio Maya, a small school made up of students from all over the world and teachers who taught beyond the standard subject matters and frequently imparted different perspectives on the things they taught.

Much to the dismay of his social studies teachers, Hyung Joo developed an interest in science and mathematics. To further his education in these areas, he enrolled in Harvey Mudd College, where he obtained a Bachelor of Science degree in Physics in 2008. Naïvely thinking he had enough of the year-round, sunny California weather, he came to Cornell to pursue a graduate degree in something mathematical. At first, he started in the field of Theoretical and Applied Mechanics, but later transferred to the Center for Applied Mathematics. Aside from a yearlong pause which took him to exotic locales in Asia, he spent his time in Ithaca, earning a Master of Science in 2014 and then working towards his PhD.

To my family

ACKNOWLEDGEMENTS

I would first like to start by thanking my adviser, Veit Elser. I feel extremely fortunate to have worked with someone who is as devoted to his students and possesses deep knowledge over many fields as Veit. Life as a graduate student has been full of highs and lows, and Veit was supportive and encouraging during all points of the journey. The past four years were intellectually challenging and stimulating thanks to his guidance, and I will be forever grateful for the many important lessons he taught me to shape me into a better researcher. I would also like to thank David Bindel and Steven Strogatz for serving on my committee and providing me with valuable advice over various points in my graduate career.

Studying the soot diffraction data would not have been possible without Duane Loh. I would like to thank him for his guidance and the weekly Skype sessions, despite the vast time difference. I would not have imagined what it would be like to perform diffraction experiments on cows until he sent me simulated results of such efforts. I would also like to thank the other students in the Elser group I have gotten to know over the years, with whom many engaging conversations were had.

I would like to acknowledge Alex Alemi for generating the first version of the "4+1" litemotif collection discussed in Chapter 4. This was subsequently modified and built on as the litemotif specifications were updated over the course of the project.

Life in Ithaca was made pleasant thanks to the many wonderful people I got to know during my time here. The TAM and CAM communities have been filled with quirky and interesting people. Through the many months of snow and grey, drinking with them (as well as many other non-alcoholic activities)

helped me cope. Also, I am grateful that some of my Harvey Mudd friends decided to come to Cornell around the same time I did.

Lastly, I would like to thank my family for their love and the support they gave me, encouraging me to follow my own path.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This dissertation is composed of two parts stemming from work done on two very different projects. Broadly speaking, the two projects address ways to understand and predict the structures of very small objects measured on the order of nanometers and ångstroms. Both projects have focused on developing tools and methods for processing large datasets with the hopes of streamlining processes which would otherwise be handled by a human researcher.

The two projects make up two parts of this dissertation, each of which is mostly self-contained. Part 1, consisting of Chapters 2 and 3, discusses automating image reconstruction from coherent X-ray imaging (CXI) experiments. Chapter 2 provides an introduction to the field and the necessary details to understand Chapter 3, which delves into our efforts to automate the image reconstruction process using CXI data from the Linac Coherent Light Source (LCLS) at the SLAC National Accelerator Laboratory.

Part 2, consisting of Chapters 4 and 5, discusses efforts at protein structure prediction and building the necessary datasets to assist in these efforts. In Chapter 4, we describe an algorithm that takes a sequence of amino acid residues as input and outputs a prediction of the three-dimensional structures of the protein backbone. Chapter 5 details efforts at generating collections of building blocks needed by the prediction algorithm.

Other than the study of small objects, another common thread tying the two parts together is the approach taken to solve various constraint satisfaction problems. There are many examples in the subsequent chapters of problems

whose solutions are defined by meeting a number of different constraints. In all these instances, we use an iterative method that searches the solution space for regions where elements satisfy all the constraints. This approach to constraint satisfaction lies at the heart of the image reconstruction and protein structure prediction algorithms. The method is detailed in Chapter 2, specifically in subsections 2.3.1 through 2.3.3. It is recommended that these two subsections be read before reading Part 2.

Part I

Coherent X-ray Imaging

CHAPTER 2

PRINCIPLES OF COHERENT X-RAY IMAGING

The field of coherent X-ray imaging (CXI) is an active area of research whose aim is to generate high resolution images of nanometer scale structures such as viruses and proteins. While the theoretical foundations have been developed over decades with advances in X-ray crystallography, it was in 1980 when the idea of extending the same methods to noncrystalline structures was proposed [1]. It took two more decades before the technique was experimentally demonstrated [2]. Since then, the field has seen tremendous advances where microscopic particles such as yeast cells, viruses, and particulate matter have successfully been imaged [3, 4, 5].

The basic premise of CXI involves two steps: scattering X-rays off an object and using the scattered information, in the form of recorded diffraction patterns, to reconstruct images (either in two or three dimensions) of the scattering object. The scattering experiments generate tremendous amounts of data which necessitate automated tools at many stages of the process from data collection to reconstruction. Currently, there are considerable efforts devoted to building these tools [6, 7, 8, 9, 10, 11, 12].

In this chapter, we discuss an overview of the experimental techniques behind CXI as well as its theoretical foundations. We then discuss the methods required to reconstruct images of the scattering objects. These discussions provide the context and background for understanding Chapter 3, which will delve into efforts on automating reconstruction of two-dimensional images of small, strongly scattering objects.

2.1 Technique

Conceptually, the experiments are simple. A monochromatic X-ray pulse is incident upon a scattering object, as shown in Figure 2.1. A small fraction of photons in the pulse scatters off the object, while the rest propagates as if it never interacted with the object. The principle behind the interaction is not exclusive to light in the X-ray regime, but resolving details on objects on the order of nanometers in size requires light with extremely small wavelengths, making X-rays the ideal candidates. A detector is set up some distance away from where the pulse-object interaction takes place to record the scattered photons. The detector usually has a gap in the middle in order to let the unscattered pulse through without damaging the electronics.

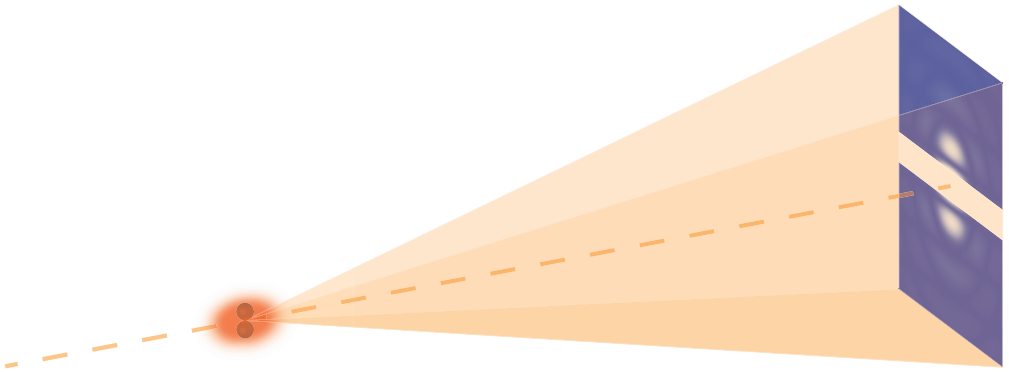


Figure 2.1: A sketch of a CXI experiment. A high energy X-ray pulse, shown in orange, interacts with a pair of nano-spheres. During the process, the spheres will vaporize. A fraction of the photons in the X-ray pulse will scatter off and land on the detectors, shown on the right, while the unscattered pulse will pass through a gap. The diffraction pattern can be used to recover a two-dimensional image of the nano-spheres.

Early experiments were successfully conducted at synchrotrons on large, strongly scattering objects made of metals such as gold [13]. As interest shifted toward imaging smaller, weakly scattering objects like biological specimens,

synchrotrons could not adequately image them without significantly degrading the objects in the process. X-ray free electron lasers (XFELs), capable of producing ultrafast pulses with fluences many orders of magnitude higher than those generated at synchrotrons, show promise for studying weakly scattering objects. The number of photons packed into a small pulse allows for significant scattering to occur, even for delicate biological samples. There is a caveat, however, in that the XFEL pulses destroy any biological sample (and pretty much any sample) that they pass through. However, if a pulse can traverse through and scatter off an object before the object degrades significantly, the recorded photons are viable for image reconstruction purposes. Studies using molecular dynamics have been done to confirm the viability of the "diffract before destroy" strategy, and found that pulses of 10 femtosecond duration can outrun any significant damage [14].

2.2 Theoretical foundation

The way X-rays scatter off an object depends on the object's morphology and electron density. The next section treats the derivation of these relationships in a mathematical and physical fashion. The discussions in the next subsections are largely based on Chapter 13 of *Principles of Optics* by Born and Wolf [15].

2.2.1 Light scattering

We begin by considering a monochromatic electromagnetic field, $\mathbf{E}(\mathbf{r}, \omega)$, incident on a linear, isotropic, nonmagnetic medium occupying a finite volume Γ.

Note that $\mathbf{r}$ is the spatial variable, $\omega = 2\pi c/\lambda$ is the angular frequency of the wave with corresponding wavelength λ, and c is the speed of light. We can consider this idealized model to be a good approximation of the XFEL pulse interacting with an object of interest. Beginning with Maxwell's equations, the t-Fourier transformed, complex-valued electric field $\mathbf{E}(\mathbf{r}, \omega)$ will satisfy the equation

$$\nabla^2 \mathbf{E}(\mathbf{r}, \omega) + k^2 \epsilon(\mathbf{r}, \omega)\mathbf{E}(\mathbf{r}, \omega) + \nabla[\mathbf{E}(\mathbf{r}, \omega) \cdot \nabla(\log \epsilon(\mathbf{r}, \omega))] = 0 \tag{2.1}$$

where $k = \omega/c$ is the wavenumber and $\epsilon(\mathbf{r}, \omega)$ is the relative permittivity of the medium. If we allow for the variations in $\epsilon(\mathbf{r}, \omega)$ to be minimal over length scales of λ, the third term in Equation 2.1 can effectively be neglected, yielding the equation

$$\nabla^2 \mathbf{E}(\mathbf{r}, \omega) + k^2 n^2(\mathbf{r}, \omega)\mathbf{E}(\mathbf{r}, \omega) = 0 \tag{2.2}$$

where $\epsilon(\mathbf{r}, \omega) = n^2(\mathbf{r}, \omega)$ and $n(\mathbf{r}, \omega)$ is the refractive index of the medium.

We can assume that the electric field is linearly polarized due to how the pulses are generated [16]. This allows us to consider each of the components of $\mathbf{E}$ separately. Let $U(\mathbf{r}, \omega)$ be one such component, and note that studying U is enough to understand the behavior of the other components of $\mathbf{E}$. This yields the scalar inhomogeneous Helmholtz equation

$$\nabla^2 U(\mathbf{r}) + k^2 n^2(\mathbf{r}) U(\mathbf{r}) = 0 \tag{2.3}$$

which we rewrite in the form

$$\nabla^2 U(\mathbf{r}) + k^2 U(\mathbf{r}) = -4\pi F(\mathbf{r}) U(\mathbf{r}) \tag{2.4}$$

where

$$F(\mathbf{r}) = \frac{1}{4\pi} k^2 [n^2(\mathbf{r}) - 1] \tag{2.5}$$

is defined as the *scattering potential*. Note that the ω has been dropped for the sake of convenience as it remains fixed in the case of a monochromatic field.

If we appeal to physical intuition, we can reason that the field U can be decomposed into an incident (*i.e.* unscattered) part $U^{(i)}$ and a scattered part $U^{(s)}$

$$U(\mathbf{r}) = U^{(i)}(\mathbf{r}) + U^{(s)}(\mathbf{r}), \tag{2.6}$$

where the incident field can be described by a plane wave satisfying the homogeneous Helmholtz equation

$$(\nabla^2 + k^2) U^{(i)}(\mathbf{r}) = 0. \tag{2.7}$$

Using these two facts, Equation 2.4 simplifies to

$$(\nabla^2 + k^2) U^{(s)}(\mathbf{r}) = -4\pi F(\mathbf{r}) U(\mathbf{r}). \tag{2.8}$$

As is often the case when studying partial differential equations, reformulating them into integral forms can yield a means to solve the equations. We consider that approach and let $G(\mathbf{r} - \mathbf{r}')$ be the Green's function of the Helmholtz operator, satisfying the equation

$$(\nabla^2 + k^2) G(\mathbf{r} - \mathbf{r}') = -4\pi \delta^{(3)}(\mathbf{r} - \mathbf{r}') \tag{2.9}$$

where $\delta^{(3)}(\mathbf{r}-\mathbf{r}')$ is the three dimensional Dirac delta function. We choose the Green's function to be of the form

$$G(\mathbf{r}-\mathbf{r}') = \frac{\exp(ik|\mathbf{r}-\mathbf{r}'|)}{|\mathbf{r}-\mathbf{r}'|^2} \tag{2.10}$$

because it is radially symmetric and $G(\mathbf{r}-\mathbf{r}') \to 0$ as $\mathbf{r} \to \infty$. The scattered wave can then be written as

$$U^{(s)}(\mathbf{r}) = \int_\Gamma F(\mathbf{r}')U(\mathbf{r}')\frac{\exp(ik|\mathbf{r}-\mathbf{r}'|)}{|\mathbf{r}-\mathbf{r}'|}\,dV'. \tag{2.11}$$

The integral for the scattered field can be further simplified depending on the situation that is intended for study. In practical CXI experiments, the detectors are placed far away from the scattering medium relative to its size. In studying such a scenario, let Q be a point in the scattering medium Γ and P be a point far away from it as depicted in Figure 2.2.

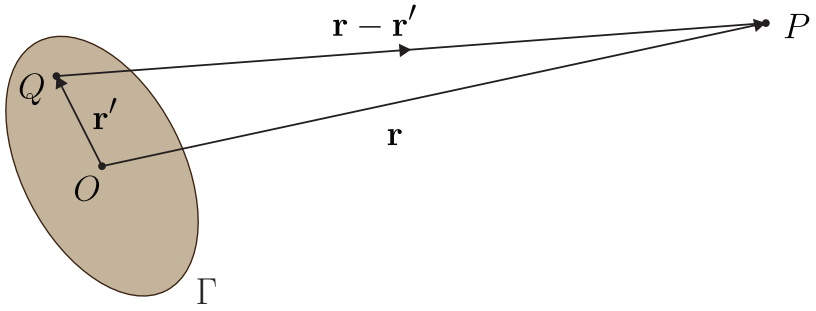


Figure 2.2: Note that $\mathbf{r}$ is a vector pointing from the origin O to a point P, $\mathbf{r}'$ is a vector pointing to Q in Γ, and P is a point far away from the scattering medium. Given the "far" away assumption, the approximations in Equations 2.12 and 2.13 can be made.

Furthermore, let $\mathbf{r}'$ be the position vector of point Q and $\mathbf{r} = r\mathbf{s}$ be the vector

of point P where s is a unit vector pointing in the direction of P. Then,

$$|\mathbf{r} - \mathbf{r}'| \sim r - \mathbf{s} \cdot \mathbf{r}' \tag{2.12}$$

and

$$\frac{\exp(ik|\mathbf{r} - \mathbf{r}'|)}{|\mathbf{r} - \mathbf{r}'|} \sim \frac{e^{ikr}}{r} \exp(-ik\mathbf{s} \cdot \mathbf{r}'). \tag{2.13}$$

If we subsitute the appropriate factors in Equation 2.11 with 2.12 and 2.13, we see that

$$U^{(s)}(r\mathbf{s}) = \frac{e^{ikr}}{r} \int_{\Gamma} F(\mathbf{r}')U(\mathbf{r}')e^{-ik\mathbf{s}\cdot\mathbf{r}'}\, dV' \tag{2.14}$$

$$= \frac{e^{ikr}}{r} f(\mathbf{s}) \tag{2.15}$$

and define $f(\mathbf{s})$ to be the *scattering amplitude* while the e^{ikr}/r factor is the *outgoing spherical wave*.

2.2.2 Born approximation

We finally note that any interaction between the XFEL pulse and the scattering medium will scatter a tiny fraction of the photons, and most will in fact traverse through the medium unaffected. In such cases, we can assume that

$$U(\mathbf{r}) \approx U^{(i)}(\mathbf{r}) \tag{2.16}$$

$$= e^{ik(\mathbf{s}_0\cdot\mathbf{r})} \tag{2.17}$$

where $\mathbf{s_0}$ is a unit vector pointing in the direction of the incident field's propagation. The assumption made in Equation 2.16 is known as the *first-order Born approximation*, and it is particularly useful in our study since it approximates the scattering amplitude

$$f_{Born}(\mathbf{s}, \mathbf{s}_0) = \int_\Gamma F(\mathbf{r}')e^{-ik(\mathbf{s}-\mathbf{s}_0)\cdot\mathbf{r}'}\, dV' \tag{2.18}$$

$$= \mathcal{F}[F(\mathbf{r}')](k(\mathbf{s}-\mathbf{s}_0)) \tag{2.19}$$

to be the Fourier transform of the scattering potential F, with the reciprocal variable quantifying the deviation of the scattered wave from the unscattered wave. As a matter of notation, we define the vector

$$\mathbf{q} = k(\mathbf{s}-\mathbf{s}_0) \tag{2.20}$$

to be the scattering vector. Equation 2.19 can then be rewritten as

$$f_{Born}(\mathbf{q}) = \mathcal{F}[F(\mathbf{r}')](\mathbf{q}). \tag{2.21}$$

Lastly, for r sufficiently large and fixed, the e^{ikr}/r factor can be taken to be a constant, so Equation 2.19 is enough to describe the scattered wave.

The consequences of Equation 2.19 are profound, as modeling scattering phenomena via Fourier transforms allows for the various properties of the transform to be exploited in studying the scattering objects, often in very creative ways. The most significant consequence is the determination of the scattering potential simply by taking the inverse three-dimensional Fourier transform of

the scattered wave

$$F(\mathbf{r}) = \mathcal{F}_{3D}^{-1}[f_{Born}(\mathbf{q}')](\mathbf{r}), \tag{2.22}$$

which would then yield the structure of the scattering object in question. This step is complicated by the fact, however, that when measurements are taken to record the scattered wave, it is usually not the wave itself, but rather the *intensity* of the wave, defined as

$$I(\mathbf{q}) = |f_{Born}(\mathbf{q})|^2 \tag{2.23}$$

which gets measured. In other words, the recorded diffraction pattern contains only the *modulus* of the complex scattered wave, but not the *phase*. This gives rise to the *phase problem*, which is discussed in greater detail in Section 2.3.

2.2.3 Diffraction geometry and Ewald sphere

Given the Fourier transform formulation of the scattered wave and intensity, it helps to think about the quantities as densities in a three-dimensional $\mathbf{q}$-space, much like how the density of the scattering object is defined in real space. The intensity recorded on a two-dimensional detector, which is the sort of data we study in the next chapter, can be thought of as a two-dimensional slice of the three-dimensional intensity described by Equation 2.23, as shown in Figure 2.3. Via the Fourier slice theorem, the two-dimensional slice is related to a two-dimensional "flattened" projection of the scattering object, which we call the object's *contrast*.

To help relate the recorded intensity to the scattering object, we define a sphere of radius k centered at the location of the scattering object as shown

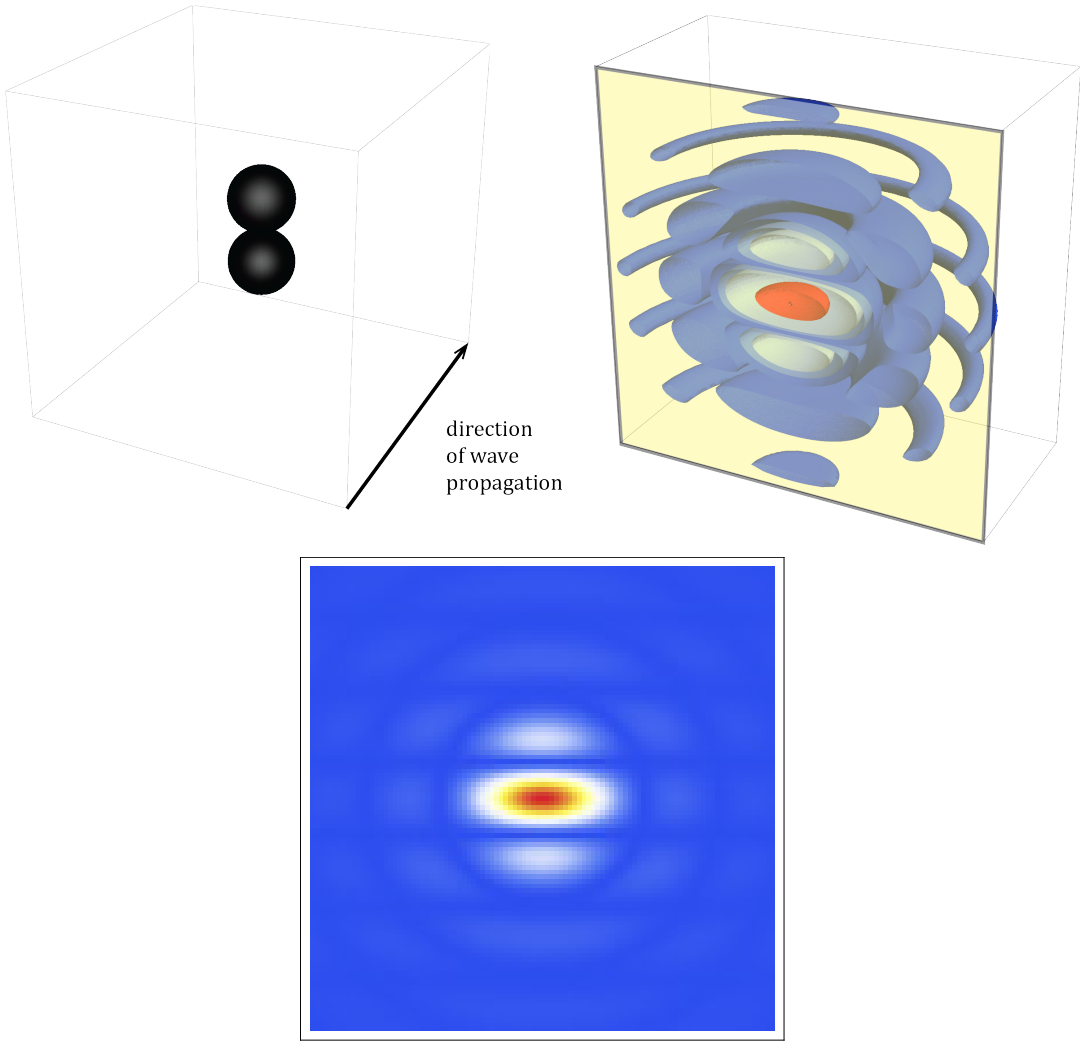


Figure 2.3: A pair of nano-spheres, top left, and the square of the three-dimensional Fourier transform of the spheres, top right. The diffraction pattern, resulting from a pulse passing through the nano-spheres in the direction pointed by the arrow in the top left subfigure, recorded by a detector, bottom center, can be thought of as a two-dimensional slice of the three-dimensional Fourier intensity, as depicted by the yellow screen on the top right subfigure.

in Figure 2.4. This sphere is called the *Ewald sphere*, and its surface captures all possible vectors ks [17]. This construction assumed that waves will largely scatter elastically, so k does not change.

Getting back to the first point made in this subsection, the intensity distribution can be thought to reside in q-space. This space has its origin in O', located

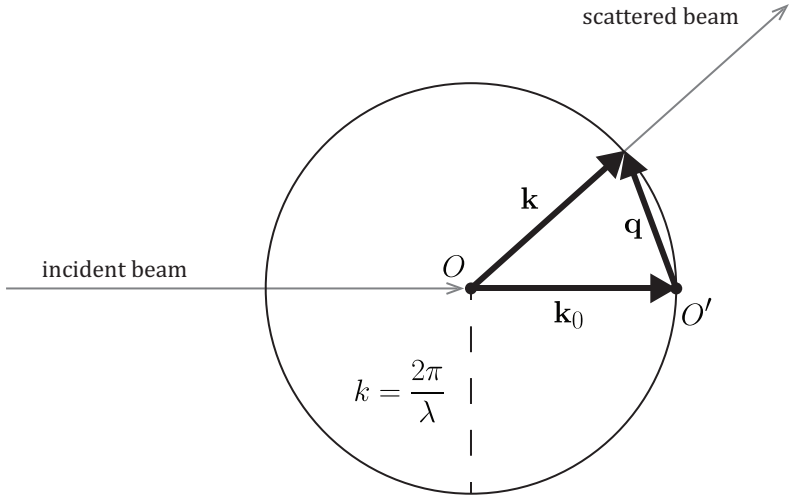


Figure 2.4: The Ewald sphere, with a radius k and its center located at O. The three-dimensional Fourier intensity is centered at O', and the detector can be thought of sampling the three-dimensional Fourier intensity space which intersects with a portion of the surface of the sphere.

$2\pi/\lambda$ away from O, along the direction of the incident wave. For a vector $\mathbf{k} = k\mathbf{s}$, there is a corresponding $\mathbf{q}$ vector. The mapping from $k\mathbf{s}$ to $\mathbf{q}$ provides a way to take recorded intensities and map them onto $\mathbf{q}$-space.

Given a flat panel detector located a distance L away from the scattering object, let the vector $\mathbf{x} = (x, y)$ be the coordinate of a particular point on the detector, with the origin defined by where the incident pulse is expected to intersect the detector, as shown in Figure 2.5. Note then that the mapping can be obtained via the geometric relation

$$\mathbf{q} = \mathbf{k} - \mathbf{k_0} = \frac{2\pi}{\lambda}(\mathbf{s} - \mathbf{s_0}) \tag{2.24}$$

$$= \frac{2\pi}{\lambda}\left(\frac{(x, y, L)}{\sqrt{x^2 + y^2 + L^2}} - (0, 0, 1)\right). \tag{2.25}$$

Using Equation 2.24, it becomes possible to map the recorded intensities into

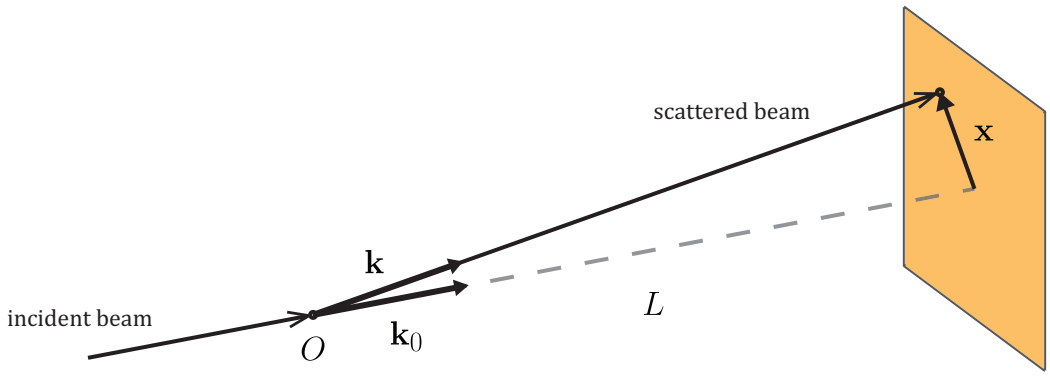


Figure 2.5: The geometric relation between between **q**-space and the "detector" -space. This can be described by Equation 2.24.

q-space. It turns out that the intensities will map onto the surface of an Ewald sphere, so the two-dimensional slice of the three-dimensional intensity we previously alluded to, especially in Figure 2.3, is not planar, but actually a curved surface. This complicates efforts to recover the scattering object's geometry, but if the curvature of the Ewald sphere is insignificant over the range of the recorded intensities, it can effectively be ignored and the slice can be approximated to be planar. The precise conditions under which this assumption can be made are discussed in Chapter 3.

2.3 Image reconstruction via phase retrieval

As we observed in the previous section, a recorded diffraction pattern $I(\mathbf{q})$ is related to a two-dimensional, flattened version of the scattering potential of the object, which we called the contrast, $\Psi(\mathbf{r})$, via the two-dimensional inverse Fourier transform relation,

$$\Psi(\mathbf{r}) = \mathcal{F}_{2D}^{-1}[\sqrt{I(\mathbf{q})}\exp(i\phi(\mathbf{q}))], \tag{2.26}$$

where $\phi(\mathbf{q})$ is the associated phase of the complex scattering amplitude. If the nano-spheres configuration shown in Figure 2.6 is scattered by a high energy X-ray laser, its two-dimensional modulus would be recorded, but its phase would not. However, in order to make use of Equation 2.26 to recover the contrast, the phase has to be recovered somehow. This scenario is known as the phase problem, a problem well known to X-ray crystallographers.

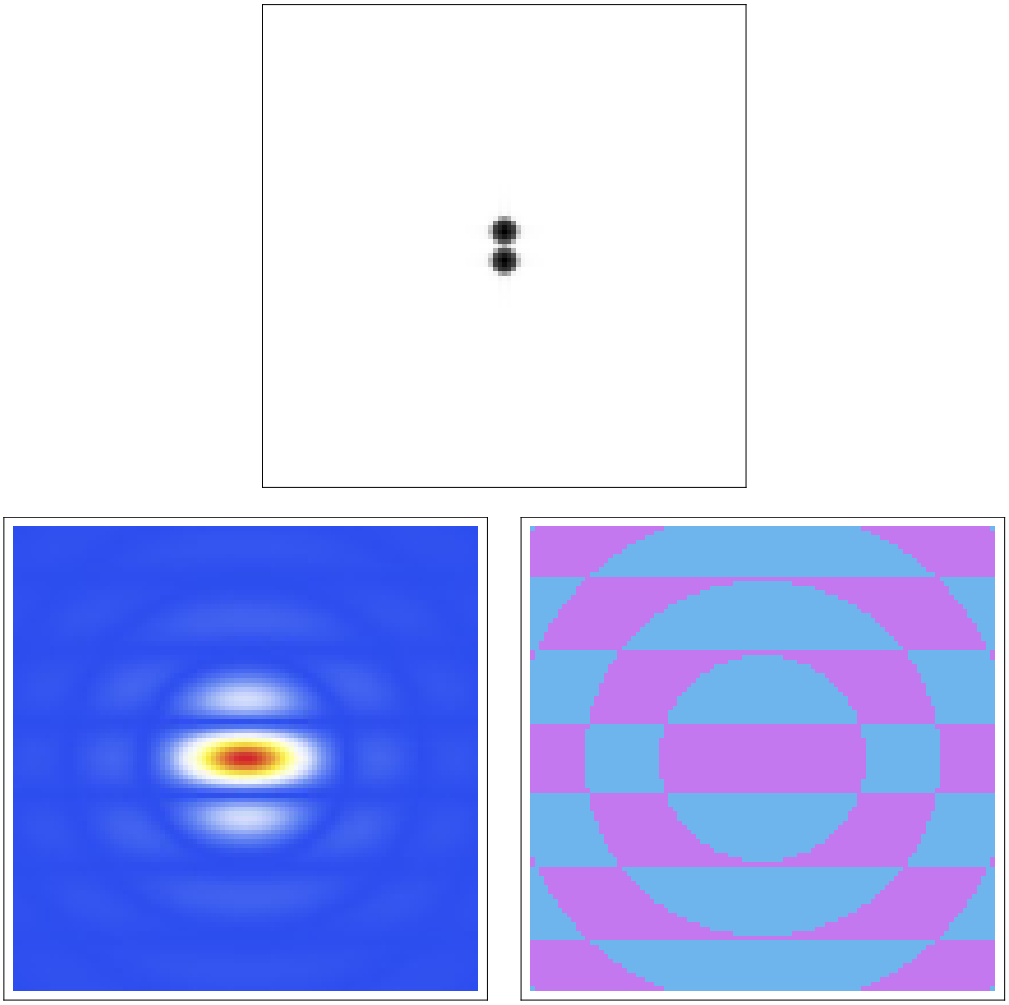


Figure 2.6: A two-dimensional projection of two nano-spheres, top center. Its two-dimensional Fourier modulus, bottom left, and corresponding phase, bottom right. In a CXI experiment, the Fourier modulus would be observed, while the phase would not be.

2.3.1 The phase problem

It was in the 1970s, beginning with Gerchberg and Saxton's alternating projection algorithm [18], that phase retrieval was addressed in an iterative manner, followed later by Fienup's seminal paper on the hybrid-input output (HIO) algorithm [19]. The general idea behind iterative techniques is that there are constraints on what the phase can be due to prior knowledge of the scattering object. In the case of CXI experiments, a crude approximation of the imaged object, specifically the space it occupies, can often be a powerful constraint in drastically restricting the possible phases for a diffraction pattern obtained from a real object.

In describing the phase retrieval problem in a more mathematical manner, we are interested in finding a $\phi(\mathbf{q})$ such that, given $I(\mathbf{q})$, it satisfies Equation 2.26 as well as an additional constraint based on the shape S of the particle. However, the caveat must be added that, although we are looking to recover $\phi(\mathbf{r})$, ultimately, it is $\Psi(\mathbf{r})$ that is of interest as that is the object's contrast. Shifting the focus to $\Psi(\mathbf{r})$, referred henceforth as Ψ, also allows for a more intuitive interpretation of the constraint satisfaction problem. From this different perspective, Ψ is a candidate for an object's contrast if the following two constraints are satisfied:

1. The magnitude of Ψ's Fourier transform must match the observed diffraction pattern.
2. Ψ can be nonzero within some region S but must be zero outside.

2.3.2 Iterative methods

The way that the two constraints are imposed is via two projection operators, P_F and P_S, known as the Fourier and support projections. They take as inputs any Ψ, and output a version of Ψ with just one of the constraints having been imposed. The Fourier projection is defined to be

$$P_F[\Psi] = \mathcal{F}^{-1} \circ M_F \circ \mathcal{F}[\Psi] \tag{2.27}$$

where

$$M_F[\hat{\Psi}] = \begin{cases} \sqrt{I(\mathbf{q})}\dfrac{\hat{\Psi}(\mathbf{q})}{|\hat{\Psi}(\mathbf{q})|} & \text{if } I(\mathbf{q}) \text{ is known and } |\hat{\Psi}(\mathbf{q})| \neq 0 \\ \hat{\Psi}(\mathbf{q}) & \text{otherwise.} \end{cases} \tag{2.28}$$

As shown in Figure 2.7, P_F rescales the Fourier magnitude of the input $\hat{\Psi}$ to match that of the square root of the measured intensity. The support projection is defined to be

$$P_S[\Psi] = \begin{cases} \Psi(\mathbf{r}) & \mathbf{r} \in S \text{ and } \Psi(\mathbf{r}) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.29}$$

and it returns a modified input where any regions of Ψ outside of S and negative values of $\Psi(\mathbf{r})$ are set to zero, as shown in Figure 2.8.

A solution to the phase problem, Ψ_s, satisfies the relation

$$\Psi_s = P_S[\Psi_s] = P_F[\Psi_s], \tag{2.30}$$

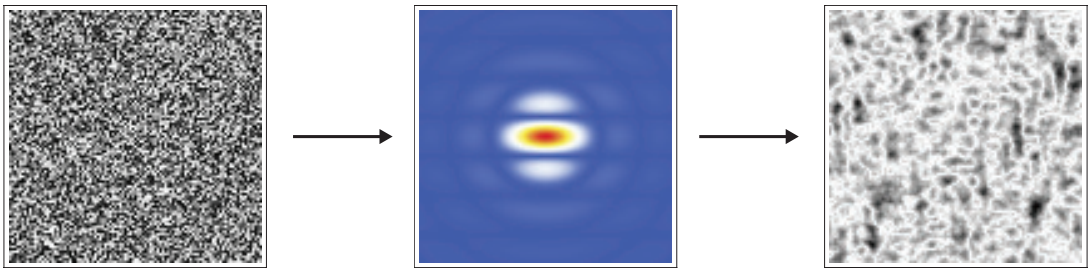


Figure 2.7: The Fourier projection, $P_F[\Psi]$, takes an input, shown left, and replaces its Fourier modulus with the diffraction pattern, shown center. It then returns the inverse Fourier transform of the modified input, shown right.

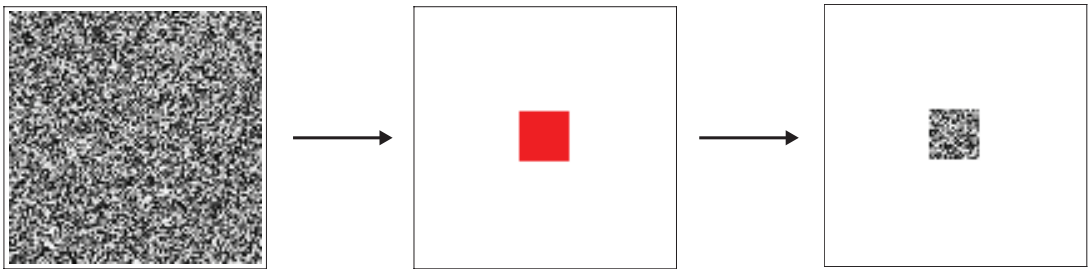


Figure 2.8: The support projection, $P_S[\Psi]$, takes an input, shown left, and zeroes any part of the input that lies outside the support, shown center, as well as any negative values present in the input. It then returns the modified input, shown right.

where the projections of both constraints on an input return the same input. Such a fixed point is known to lie at the intersection of the two constraints sets, and this, or a close approximation of this, is what we wish to find when solving the phase problem.

Beginning with a random initial contrast Ψ_0, iterative phase retrieval methods make use of the projections in various ways to "search" for solutions satisfying both constraints. There are a number of methods which have been proposed over the years, all possessing different properties and quirks suitable for particular occasions. Some of the more widely used algorithms are tabulated in Table 2.1.

Gerchberg and Saxton's alternating projections is one of the simplest ways

of searching for a solution, where a contrast is updated via a straightforward composition of both projections,

$$\Psi_{n+1} = (P_S \circ P_F)[\Psi_n] = P_S[P_F[\Psi_n]] \tag{2.31}$$

$$= \begin{cases} P_F[\Psi_n] & \mathbf{r} \in S \\ 0 & \mathbf{r} \notin S. \end{cases} \tag{2.32}$$

The algorithm has been known to find "non-solutions" which do not satisfy Equation 2.30 before effectively stopping its searching [20]. This phenomenon, coined *stagnation*, prevents the method from exploring alternative solutions and is one of the main reasons why the method has been superseded by more sophisticated ones.

Table 2.1: Some of the widely used phase retrieval algorithms. Note that $R_o[\Psi] = 2P_o[\Psi] - \Psi$ is defined as the *reflection operator* and β is a free parameter, usually between 0 and 1.

Method	$\Psi_{n+1} =$
Alternating projections [18]	$(P_S \circ P_F)[\Psi_n]$
Hybrid input-output (HIO) [19]	$\begin{cases} P_F[\Psi_n] & \mathbf{r} \in S \\ \Psi_n - \beta P_F[\Psi_n] & \mathbf{r} \notin S \end{cases}$
Difference map (DM) [21]	$\Psi_n + \beta((P_S \circ f_F)[\Psi_n] - (P_F \circ f_S)[\Psi_n])$ $f_F[\Psi_n] = P_F[\Psi_n] + (P_F[\Psi_n] - \Psi_n)/\beta$ $f_S[\Psi_n] = P_S[\Psi_n] - (P_S[\Psi_n] - \Psi_n)/\beta$
Relaxed averaged alternating reflections (RAAR) [22]	$\Psi_n + \frac{1}{2}\beta((R_S \circ R_F)[\Psi_n] - R_F[\Psi_n])$

Fienup's HIO algorithm gets around the stagnation problem by updating the

regions outside the support S with a term inspired from control theory,

$$\Psi_{n+1} = \begin{cases} P_F[\Psi_n] & \mathbf{r} \in S \\ \Psi_n - \beta P_F[\Psi_n] & \mathbf{r} \notin S \end{cases} \tag{2.33}$$

where β is a free parameter between 0 to 1. Rather than zeroing the regions outside S, the HIO algorithm introduces a negative feedback effect which encourages the iterate to get closer to zero by updating the region with the difference between the previous iterate and the Fourier projection of the iterate.

A common way to check whether an algorithm has found a solution is by keeping track of the change between iterates. The error metric quantifies this change, and is defined as

$$||\epsilon[\Psi_n]|| = ||\Psi_n - \Psi_{n-1}|| \tag{2.34}$$

where $||\cdot||$ is the Euclidean norm. In successful phase retrieval methods, the error metric will decrease with increasing iterations, as shown in Figure 2.9. When the metric falls below some small, fixed tolerance τ and consistently stays close to τ over many iterations, this is an indication that the iterate does not change much after each update. This generally happens when an iterate approximately satisfies Equation 2.30, so iterations are usually stopped then.

There are no known convergence guarantees for these iterative methods because non-convex sets, like the constraint sets of the Fourier projection, have not extensively been studied for their convergence properties. Furthermore, recorded diffraction patterns are noisy due to the statistical nature of photon hits and detector anomalies. Therefore, the appropriate number of iterations

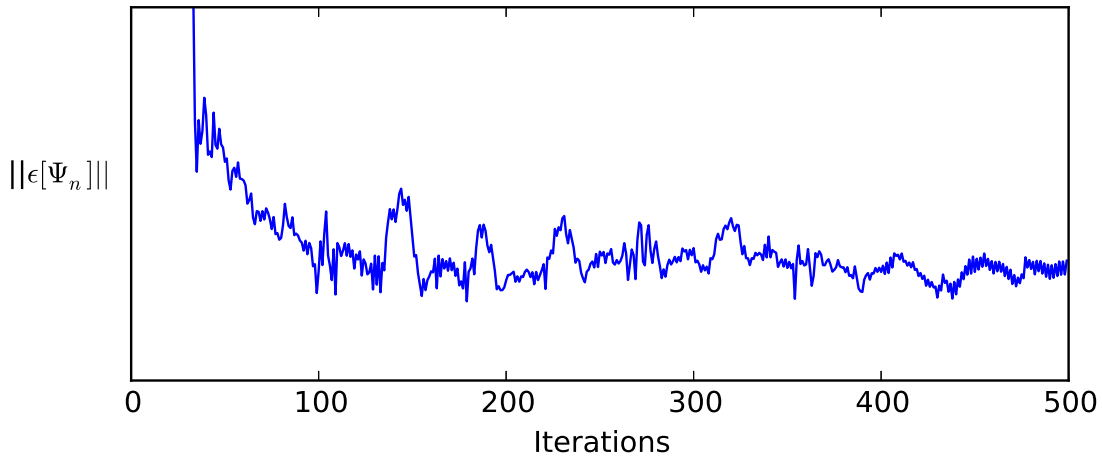


Figure 2.9: The error metric time series, $||\epsilon[\Psi_n]||$, as a function of the number of iterations n. In most successful phase retrieval methods, the error metric generally decreases with increasing iterations while exhibiting noticeable fluctuations. It usually does not converge to zero when both constraints cannot completely be satisfied.

n_{iter} until the error metric falls below τ is obtained heuristically.

With the most recent iterate at hand, Ψ_f, which satisfies

$$||\epsilon[\Psi_f]|| < \tau \tag{2.35}$$

one way a solution Ψ_{s_f} can be found is by applying either projections on it. To obtain a final solution Ψ_s, multiple solutions $\Psi_{s_{f_1}}, \Psi_{s_{f_2}}, \ldots, \Psi_{s_{f_m}}$ are obtained and averaged. These solutions could either be obtained by prolonging the iterations after Equation 2.35 has been reached and sampling m different iterates which still satisfy that condition, or starting with m initial random iterates and attempting m different reconstructions until Equation 2.35 is satisfied for each attempt. Either step is necessary to obtain a reliable solution. As Figure 2.10 shows, the various $\Psi_{s_{f_i}}$s all possess small and random differences among the collection of solutions. Through averaging, these effects are minimized, as Fig-

ure 2.11 shows.

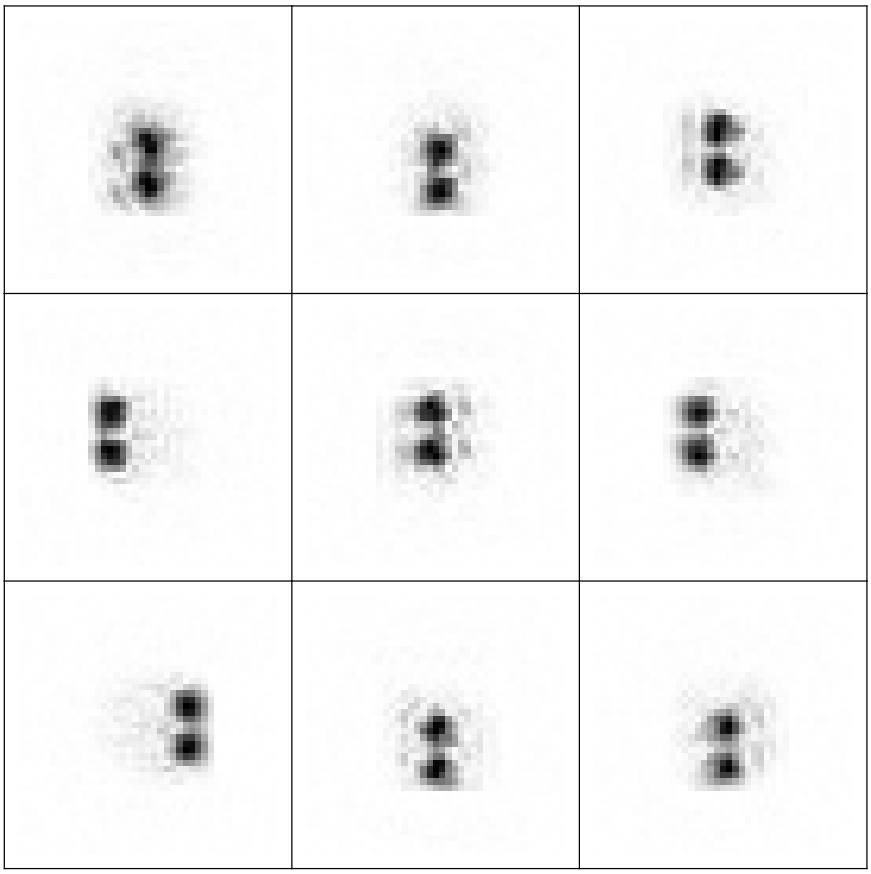


Figure 2.10: Different attempts at reconstructions using the diffraction pattern from Figure 2.3. For each of the nine reconstructions, a different, random initial iterate was subjected to the difference map. Although all nine attempts have successfully reconstructed the image of the nano-spheres, there are significant variations between each of the reconstructions.

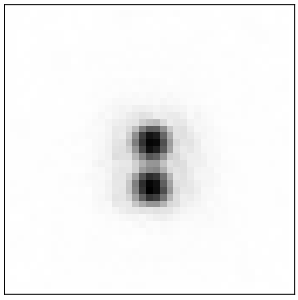


Figure 2.11: Final image reconstruction, obtained by averaging the nine different reconstructions from Figure 2.10. Note that many of the variations have been averaged away and the final image is smoother than the nine reconstructions that went into making it.

2.3.3 Difference map

The difference map is an iterative method developed by Elser [21]. As stated in Table 2.1, the difference map is of the form

$$\Psi_{n+1} = \Psi_n + \beta_D((P_S \circ f_F)[\Psi_n] - (P_F \circ f_S)[\Psi_n]) \tag{2.36}$$

where

$$f_F[\Psi_n] = P_F[\Psi_n] + (P_F[\Psi_n] - \Psi_n)/\beta_D \tag{2.37}$$

$$f_S[\Psi_n] = P_S[\Psi_n] - (P_S[\Psi_n] - \Psi_n)/\beta_D \tag{2.38}$$

and β_D is a free parameter. When $\beta_D = 1$, the difference map reduces to a case of the HIO where $\beta = 1$.

Much like the HIO algorithm, one of the main draws of the difference map is its ability to avoid stagnation. There are potential regions in a solution space where Equation 2.30 loosely holds, which are called *near-intersections*, and they can often be places where algorithms like alternating projections stagnate. The difference map will tend to send iterates away from near-intersections towards other regions in the solution space where better near-intersections, or fixed points, may exist. This is how stagnation is overcome, but this can often hinders the map from settling near a solution.

In Chapter 3, we make use of the difference map, but in a modified fashion that helps tame its eagerness to explore other near-intersections.

2.3.4 Phase uniqueness

To have a reasonable chance at determining the correct phases and recovering an accurate image, it helps to know as much about the object's support. One can imagine that it would be much more difficult to obtain the phases of the nano-sphere contrast in Figure 2.6 if one started with a generous support in real space whose area equals the area of the recorded diffraction pattern as opposed to a support that tightly contains the two spheres.

It was shown by Elser *et al.* [23] that a sufficient criterion for ensuring unique phase retrieval in most cases can be expressed via the constraint ratio

$$\Omega = \frac{A_{auto}}{2A_{object}} \tag{2.39}$$

where A_{auto} is the area of the support of the autocorrelation of the object and A_{object} is the area of the object's support. Note that the autocorrelation of the object can be obtained via the relation

$$\Psi * \Psi = \mathcal{F}[I(\mathbf{q})] \tag{2.40}$$

and A_{auto} can be obtained from thresholding Equation 2.40. For $\Omega > 1$, finding a unique solution to the phase problem is in most cases tractable since Ω is the ratio of the number of independent measurements to the number of contrast variables to be determined. For the other case when $\Omega < 1$,there is no uniqueness without additional knowledge. Equation 2.39 suggests that a smaller A_{object} will better ensure uniqueness over a larger A_{object}.

2.3.5 Dynamic support update (Shrinkwrap)

As discussed in the previous subsection, key to the phase retrieval process is a tight support S. Smaller supports ensure that reconstruction algorithms will stand a better chance at finding a unique solution. However, without prior knowledge of the object's shape, finding a tight support seems implausible. This circular impasse seems to make the reconstruction problem intractable. Fortunately, methods were developed to address this issue. One commonly used technique is known as Shrinkwrap [25]. The method allows for S to be updated during the reconstruction process via the outlined steps:

1. Define an initial support S_0. In *Marchesini et al.* [25], the autocorrelation of the object's contrast is used to define S_0. Mathematically, this is done as follows:

$$S_0 = T_\alpha[\mathcal{F}[I(\mathbf{q})]] \tag{2.41}$$

   where T is the indicator function that sets any regions of the input less than α to 0 and the rest to 1.

   In practice, any support that differs in size from the true support by a factor as large as two or as small as a half in area has been shown to be a viable S_0.

2. Use S_0 as the support and apply the phase reconstruction algorithm on an initial guess for a fixed number of iterations.

3. After n number of iterations, take the reconstructed image Ψ, convolve it with a Gaussian of narrow width to blur the image, and use that to

determine a new support S_u. Mathematically, that step translates to

$$S_u = T_\beta[\Psi(\mathbf{r}) * H_\gamma(\mathbf{r})] \tag{2.42}$$

where $H_\gamma(\mathbf{r})$ is a Gaussian centered at the origin with a variance of γ^2 and β is some small percentage of the maximum value of $\Psi(\mathbf{r})$.

4. Repeat step 2 using the updated support S_u, then repeat step 3 while occasionally adjusting the parameters n, β and γ.

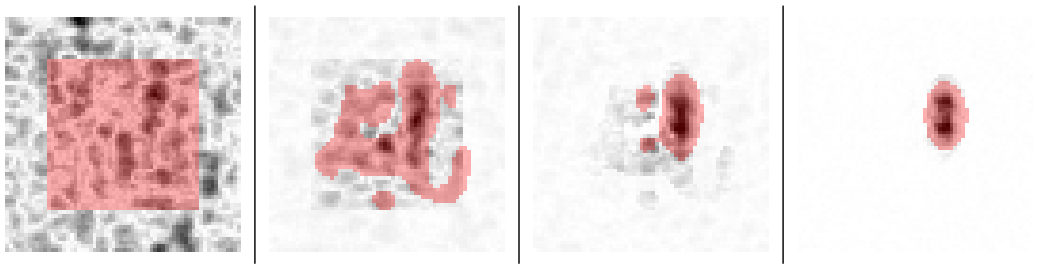


Figure 2.12: Beginning with a square support, shown left in red, Shrinkwrap dynamically updates the support between iterations by redefining the support around regions of high contrast. In practice, given the right set of parameters, Shrinkwrap will downsize an initially large support until it tightly defines a region where the object's contrast is expected to be, as the figure on the right suggests.

These steps ensure that the support gets updated based on high contrast regions of the iterate. In practice, as the iterates get updated, Shrinkwrap will contract the support until it tightly contains the contrast, much like the packing method from which it gets its name. Figure 2.12 shows how a square support eventually contracts to an oval which contains the nano-spheres contrast.

2.3.6 Reconstruction quality and PRTF

As was mentioned in Subsection 2.3.2, multiple reconstructed contrasts are obtained and averaged to arrive at a final, reconstructed contrast. One popular measure to gauge the quality of an averaged reconstruction is through the phase retrieval transfer function (PRTF) [26], defined as

$$\begin{aligned} \mathrm{PRTF}[\Psi_{ave}](\mathbf{q}) &= \frac{1}{m}\left|\sum_{j=1}^{m}\exp(i\phi_j(\mathbf{q}))\right| && (2.43) \\ &= \frac{1}{m}\left|\sum_{j=1}^{m}\frac{\mathcal{F}[\Psi_j](\mathbf{q})}{|\mathcal{F}[\Psi_j](\mathbf{q})|}\right| && (2.44) \end{aligned}$$

where $\phi_j(\mathbf{q})$ is the phase at $\mathbf{q}$ of the reconstructed contrast Ψ_j. For a value $\mathbf{q}$, the PRTF measures how well the phases agree over a set of reconstructed contrasts. If there is near-perfect agreement between all the phases, the PRTF will tend to 1, whereas disagreement will result in values closer to 0. This then gives a good sense of regions of $\mathbf{q}$ where the reconstructed phases can be trusted more so than others.

The PRTF, in general, decreases monotonically as $|\mathbf{q}|$ gets larger, as Figure 2.13 demonstrates. Smaller $|\mathbf{q}|$ values correspond to features on larger length scales, which mostly agree in an ensemble of similar reconstructions. However, as finer features are compared over the ensemble, there is more likelihood for disagreement. Regions of large $|\mathbf{q}|$ values describe these finer features, so we expect the PRTF to be lower there.

Angularly averaging the PRTF results from a two-dimensional plot has commonly been used to find the value $|\mathbf{q}|$ which serves as a cutoff c for where the phases can no longer be considered reliable, as demonstrated in Figure 2.14.

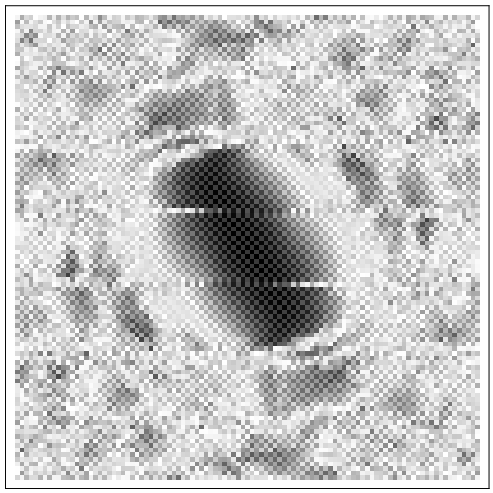


Figure 2.13: The PRTF of the averaged reconstruction shown in Figure 2.11, in grayscale. Note that there is better agreement near the center, where $|\mathbf{q}|$ is smaller. As diffraction patterns become noisier and fainter at regions of higher $|\mathbf{q}|$, the PRTF there will tend closer to zero.

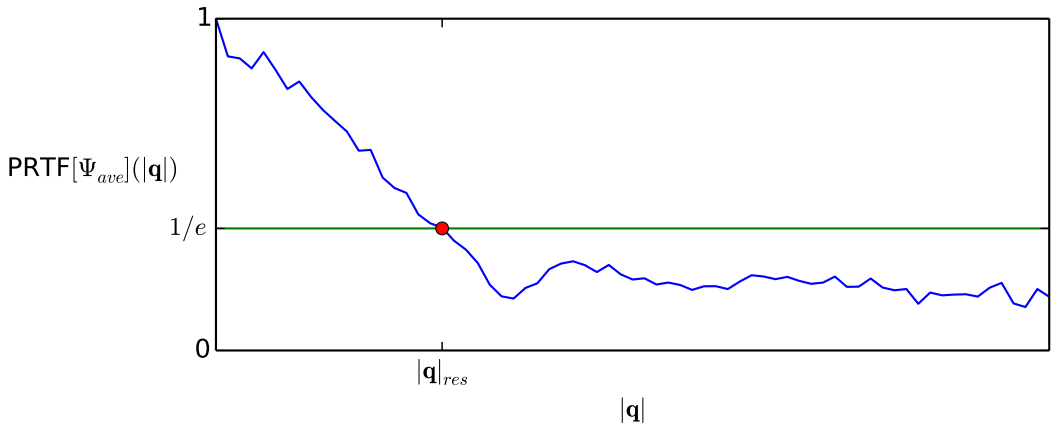


Figure 2.14: The rotationally averaged version of Figure 2.13, shown in blue. Conventionally, the lowest $|\mathbf{q}|$ where the PRTF reaches $c = 1/e$, depicted as the green line, is used to determine an effective resolution. The red dot shows where the rotationally averaged PRTF and the cutoff c meet.

The value for which $|\mathbf{q}|$ first reaches c (commonly, $c = 1/e$ is used) is then used to determine an effective resolution for the averaged reconstruction, which is given by the formula

$$|\mathbf{r}|_{res} = \frac{\pi}{|\mathbf{q}|_{res}}. \tag{2.45}$$

CHAPTER 3

UNSUPERVISED IMAGE RECONSTRUCTION

The contents of this chapter are based on work published in Optics Express with a multitude of coauthors [27].

3.1 Introduction

Single-shot diffraction imaging via X-ray free-electron lasers [28] has emerged as a potentially significant tool for studying particles in the nanometer regime, from biological samples [3, 4] to nanocrystals [29]. As particles of interest are propelled into the path of short X-ray pulses of high fluence such as those generated at the Linac Coherent Light Source (LCLS), their interaction diffracts a small fraction of the photons off the particle before the onset of significant radiation damage. The resulting far-field diffraction patterns, recorded on X-ray detectors, can be used to reconstruct real space contrasts of the diffracting particles via iterative phase retrieval methods [30]. Since the pulse-particle interactions occur mid-flight, imaging individual particulate matter such as soot *in situ* at nanometer resolution is made possible, allowing for morphological studies of in-flight particles [31, 5] that have in the past relied on other imaging techniques such as transmission electron microscopy [32], where substrate deposition could potentially alter particles' morphologies.

A typical imaging experiment could generate hundreds of thousands of usable diffraction patterns in a single day, compelling the need for an unsupervised contrast reconstruction process requiring minimal user guidance. Due to experimental realities such as variable pulse profiles and detector noise, how-

ever, there are significant differences in the quality of the data from pattern to pattern. Simply automating existing tools for reconstruction would be problematic, and in some cases insufficient, especially when these tools rely heavily on human supervision. Any scalable protocol ought to replace user guidance and visual inspection with efficient unsupervised algorithms.

In this chapter, we present a series of measures that aim to facilitate the unsupervised contrast reconstruction of a large collection of single shot diffraction patterns. We identified steps during the reconstruction process that require user guidance and replaced them with reasonable algorithms. Through these measures, we were able to successfully reconstruct hundreds of contrasts with minimal guidance.

3.2 Experiment and data set

In this study we worked with highly variable diffraction patterns of soot particles of multiple length scales. Data was collected at the Atomic, Molecular and Optical Science beam line at the LCLS. Two different kinds of soot particles were considered for imaging: particles created by a Palas GFG100 spark source generator [33] and NIST 2975 diesel soot particles [34]. In separate runs, the Palas and NIST soot were propelled into the path of X-ray pulses by a differentially pumped aerodynamic focusing inlet [35]. Some of the particles, when they reached the interaction region with a velocity of $100-200$ m/s, were intercepted by a single X-ray pulse focused to an area of about $10\mu\mathrm{m}^2$ with an average fluence of 4×10^{12} photons, each with 1.24 keV of energy, per pulse, assuming a transmission efficiency of 20%. The scattered photons were recorded on a pair of

pn-junction charge coupled device (pnCCD) panels installed in the CFEL ASG Multi-Purpose (CAMP) instrument [36]. Each panel contained 512×1024 pixels, each of area $75 \times 75 \mu\mathrm{m}^2$. A gap of $1.6\mathrm{mm}$ between the panels and semicircular cutouts of 1.2 mm diameter allowed for the passage of the pulse into a beam dump. Further details can be found in Loh *et al.* [5].

Pulse generation and the detector readout rate coincided at 60 Hz, allowing for a theoretically maximum data collection rate of 2.2×10^5 patterns per hour. In practice, however, the sample hit rate was lower than the pulse generation rate because of random particle injection. In the soot experiments, the hit rate was observed, on average, to be 0.09 Hz. A total of 953 successful hits were identified and considered for reconstruction.

3.3 Practical considerations

Any collection of diffraction patterns, even when sorted and classified [10], is bound to show differences in quality from pattern to pattern. The pulse-particle interaction contributes to much of this variability, as the profile of individual pulses can differ from each other. Wavefront aberrations in pulses can result in randomly shifted diffraction patterns which need to be corrected [37]. Also, as each pulse's transverse profile cannot be assumed to be a planar wave of constant intensity, the position of each randomly injected particle during the pulse-particle interaction, relative to the focus, has a noticeable effect on the signal-to-noise ratio of the resulting pattern, a value that is already affected by the variability in the pulse fluence.

Given the incomplete nature of phasing algorithms, successful contrast re-

construction is not guaranteed within a set number of iterations, even for ideal, noiseless patterns. As a result, reconstructions from the same pattern can exhibit significant differences. Noise can further frustrate the phasing process and encourage significant variability between reconstructions. A reliable check for the confidence in a reconstruction is to see how often the algorithm will arrive at the same, or similar solution, beginning from different initial conditions. This step, often done visually, can be a major bottleneck in the reconstruction process.

In this section, we introduce techniques to address these various issues. In the first part, we discuss ways of exploiting centrosymmetry to correctly center each pattern. In the next part, we highlight a noise robust phasing algorithm that can handle patterns with low signal-to-noise ratios. Then, we propose a technique for assessing the reliability of a reconstruction algorithmically. Lastly, we suggest a strategy to check the degree to which the missing data region could affect the final reconstruction.

3.3.1 Centrosymmetry of diffraction patterns

While the detailed form of each X-ray pulse is lost as soon as it is absorbed into the beam dump, the pulse variability is often noticeable in diffraction patterns. Random phase tilts in the pulses is one such detail, and they were observed to translate diffraction patterns of polystyrene nano-spheres by as much as six pixels [37]. These translations, when uncorrected, could potentially decrease the overall resolution of the reconstructed contrasts, especially when the speckle features are roughly on the same scale as the translations. Thus, correctly centering diffraction patterns before reconstruction is attempted is crucial.

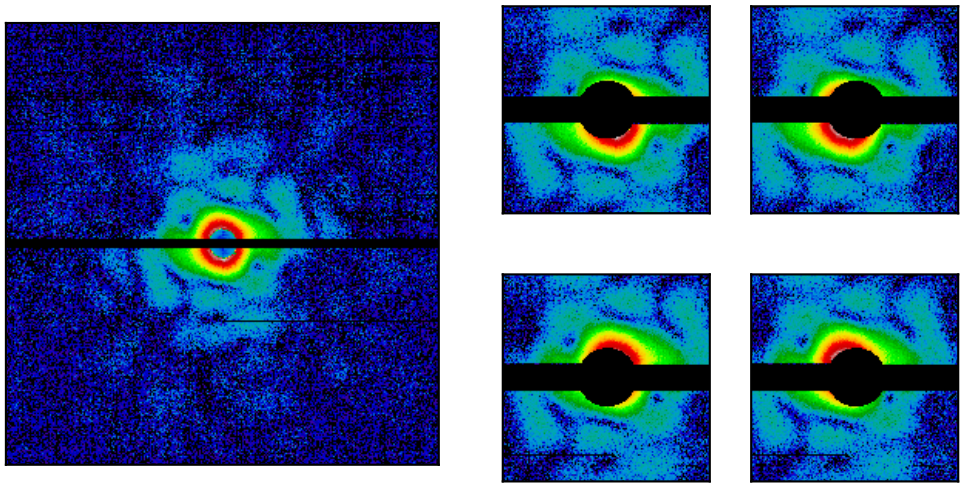


Figure 3.1: To find the center of a diffraction pattern (left), square regions, translated by a set of candidate shifts (exaggerated on the right), are tested for centrosymmetry. An identical mask is applied to each of these square regions to mask out the missing central intensities. The shifted square region that is most centrosymmetric (see text for details) is presumed to be properly centered. This diffraction pattern was found to be shifted to left by two pixels.

The diffraction pattern of a real (*i.e.*, not complex-valued) object can be approximated as centrosymmetric when the extent of the Ewald sphere's curvature is less than half a speckle diameter, $d/2$, at the edge of the detector,

$$k(1 - \cos\theta_{max}) \ll d/2, \tag{3.1}$$

where $k = 2\pi/\lambda$ is the magnitude of the wave vector and θ_{max} is the maximum scattering angle. Equation 3.1 can be simplified to

$$\theta_{max} \ll 1/N, \tag{3.2}$$

where $N = k\theta_{max}/d$ is approximately the number of speckles that can be counted on a ray from the origin to the edge of the detector.

When unshifted, the Fourier transform of the intensity $I(\mathbf{q})$ is the autocorrelation of the particle contrast Ψ. Mathematically, $\mathcal{F}[I(\mathbf{q})]$ should be real and centrosymmetric if absorption effects are negligible. When shifted by an unknown amount, $\mathbf{q}_{unknown}$,

$$\mathcal{F}[I(\mathbf{q}-\mathbf{q}_{unknown})] = \mathcal{F}[I(\mathbf{q})]\exp(-i\mathbf{q}_{unknown}\cdot\mathbf{x}) \tag{3.3}$$

the Fourier transform of the shifted intensity is equal to the Fourier transfer of the unshifted intensity multiplied by a linear phase ramp. To identify the shift that best approximates $\mathbf{q}_{unknown}$, the intensity $I(\mathbf{q}-\mathbf{q}_{unknown})$ is shifted by a known amount $\mathbf{q}_C$. Once the Fourier transform is then computed, the sum of the absolute values of its imaginary components

$$\sum_x |\mathrm{Im}[\mathcal{F}[I(\mathbf{q})]\exp(-i\mathbf{q}_{unknown}\cdot\mathbf{x})\exp(-i\mathbf{q}_C\cdot\mathbf{x})]| \tag{3.4}$$

will equal zero if

$$\mathbf{q}_C = -\mathbf{q}_{unknown} \tag{3.5}$$

and the shifts are restricted to small values. Because actual recorded intensities are not perfectly centrosymmetric due to noise and missing data regions, Equation 3.4 will most often equal a nontrivial value even when the proper shift is found. Still, the sum should be smaller for Equation 3.5 than for other shifts.

To implement the shift locator, square regions of the pattern centered at different offsets q_o are cut out as shown in Figure 3.1. A mask, fixed with respect to the square cutout, is applied to cover up the CCD gap and the central, circu-

lar region of unreliable photon count. The mask is made thicker so that it can be applied uniformly on any square cutout and still cover up the appropriate regions. The Fourier transform of these square cutouts is computed, and the offset with the lowest sum of the absolute value of the imaginary components is identified as the correct shift.

3.3.2 Noise robust difference map

In far-field diffraction theory, the contrast Ψ is the scattered wavefront immediately past the scattering particle. It can be characterized by I via the inverse Fourier transform relation,

$$\Psi = \mathcal{F}^{-1}[\sqrt{I}\exp(i\phi)], \tag{3.6}$$

where ϕ is the associated phase of the complex exit wave. To recover Ψ, both the magnitude and the phase of the observed wavefront must be known. However, since the phase is not recorded, it is recovered using other information, such as the size and shape of the scattering particle.

Phase retrieval methods in use today employ iterative schemes to find a ϕ that best reflects all available information regarding the scattering particle. This is done by finding a Ψ that satisfies the measured I as well as an additional constraint based on the shape S of the particle. Two projection operators, P_F and P_S, known as the Fourier and support projections, respectively, are defined as follows,

$$P_F[\Psi] = \mathcal{F}^{-1} \circ M_F \circ \mathcal{F}[\Psi] \tag{3.7}$$

where

$$M_F[\hat{\Psi}] = \begin{cases} \sqrt{I(\mathbf{q})}\dfrac{\hat{\Psi}(\mathbf{q})}{|\hat{\Psi}(\mathbf{q})|} & \text{if } I(\mathbf{q}) \text{ is known and } |\hat{\Psi}(\mathbf{q})| \neq 0 \\ \hat{\Psi}(\mathbf{q}) & \text{otherwise} \end{cases} \tag{3.8}$$

rescales the Fourier magnitude of the input to match that of the square root of the measured intensity and

$$P_S[\Psi] = \begin{cases} \Psi(\mathbf{r}) & \mathbf{r} \in S \text{ and } \Psi(\mathbf{r}) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

sets to zero any region that lies outside of S and imposes positivity. Beginning with a random initial contrast Ψ_0, phase retrieval methods search for solutions by projecting iterates onto the constraint sets through a combination of the projection operations and follow the form

$$\Psi_{n+1} = \Psi_n + \epsilon[\Psi_n], \tag{3.10}$$

where $\epsilon[\Psi_n]$ is an additive update to the iterate which depends on the choice of the phasing method. When the error metric $||\epsilon[\Psi_n]||$ falls below some fixed tolerance, the iterations are stopped and the reconstruction is defined by the estimate $\Psi \approx P_F[\Psi_{n+1}]$.

We use the $\beta = 1$ form of the difference map [21],

$$\Psi_{n+1} = \Psi_n + P_S\left[2P_F[\Psi_n] - \Psi_n\right] - P_F[\Psi_n] = \Psi_n + \epsilon_D[\Psi_n], \tag{3.11}$$

which is equivalent to the $\beta = 1$ form of Fienup's hybrid input-output rule [19]. The difference map is best suited for finding a true common element in the constraint sets, on the assumption one exists. When dealing with measured diffraction patterns, however, the presence of noise could shift the Fourier constraint set such that it does not exactly intersect the support constraint set. As a result, the difference map's propensity for guiding iterates away from near-intersections to avoid stagnation works to its disadvantage as iterates are sent elsewhere to look for solutions. Practically speaking, this results in a frustrated search where reconstructions associated with each iterate fluctuate significantly in shape and size.

In the face of variable signal-to-noise ratios, the phase retrieval process should be robust so that the search does not so easily stray from near-intersections [38, 39, 22]. Loh *et al*. [38] propose an intermediate step where Ψ_n is updated by the formula

$$\Psi'_n = \alpha\Psi_n + (1-\alpha)P_F[\Psi_n], \tag{3.12}$$

where $0 \leq \alpha \leq 1$ is a "leash" parameter that reins in the iterate such that it is brought closer to the Fourier constraint set before it is run through the difference map:

$$\Psi_{n+1} = \Psi'_n + \epsilon_D[\Psi'_n]. \tag{3.13}$$

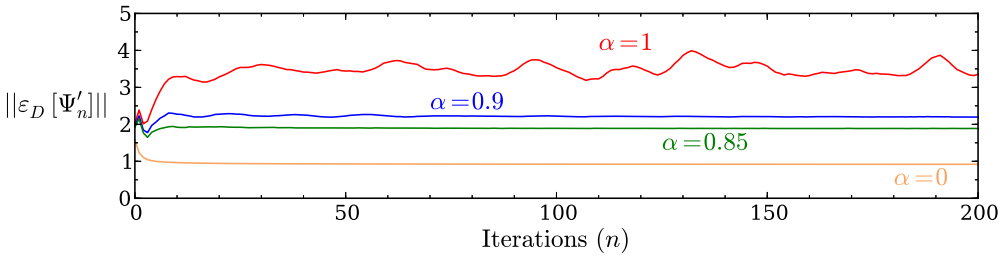


Figure 3.2: The stability of the modified difference map for various α's around a solution can be measured by the error metric, $||\epsilon_D[\Psi'_n]||$. Starting with a final reconstruction (*i.e.* solution) as the initial contrast and using a fixed support previously generated with Shrinkwrap [25], the modified difference map continues on its search in the neighborhood of the solution. An α slightly decreased from unity will significantly tighten the scope of the search and improve the stability of the difference map around a solution.

In our trials, $\alpha = 0.85$ was used. The choice for α reflects a desire to balance out the need for preventing the search from deviating from a near-intersection too much while also preventing it from settling too easily near a point which may not necessarily best reflect the near-intersection. There is some latitude in the choice of α as even a slight decrease from unity will significantly tighten up the search neighborhood around a near-intersection, as Figure 3.2 suggests.

3.3.3 Reconstruction assessment

In iterative phase retrieval methods, convergence to a unique solution within a set number of steps is not necessarily guaranteed, so the iterations are stopped when the difference between iterates is small enough or a large number of iterations, t_{max}, is reached. Consequently, an individual reconstruction will, at times, seem like it has not converged or perhaps even converged to a seemingly different point when compared to a different reconstruction. When presented with a collection of dissimilar reconstructions, visual inspection usually aids in as-

sessing which reconstructions are successful, but this would be time consuming when processing thousands of diffraction patterns.

Since the particle's contrast is not known beforehand, assessing the success of reconstructions presents another challenge as there are no training examples to aid in assessing. In practice, given a set of m reconstructions, the largest subset that contain similar looking reconstructions is usually deemed to be a successful collection. This practice, however, relies on the assumption that the phasing algorithm can guide iterates to the correct near-intersection most of the time.

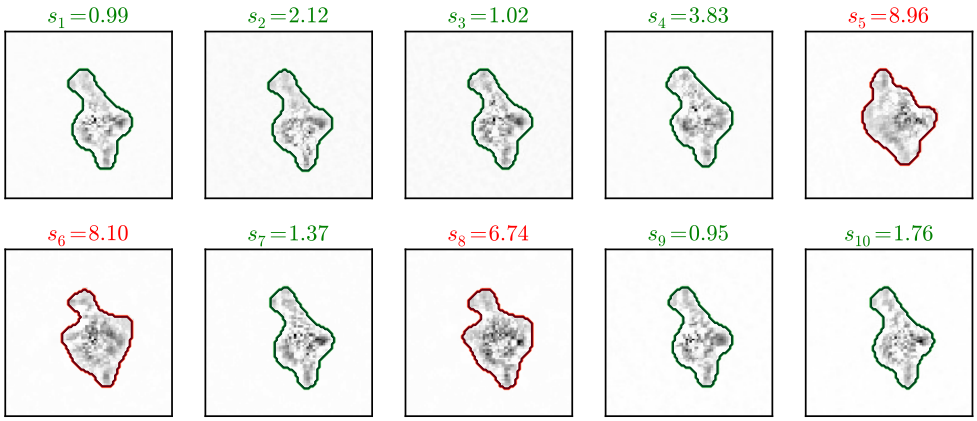


Figure 3.3: Ten individual reconstructed contrasts with overlaid outlines of their supports, as found by Shrinkwrap, and their corresponding s_i values. The reconstructions whose s_i's exceed the threshold $s_{max} = 5\%$ are marked in red and were deemed failures.

The steps taken in visually assessing and rejecting reconstructions are used to devise an algorithm that can perform the same task. Beginning with a set of m individual reconstructions, their corresponding supports, as found by Shrinkwrap [25], are compared. The supports are preferred over the reconstructions as that will emphasize during comparison the overall low-resolution shapes of the reconstructed contrasts as opposed to their subtle high-resolution features which

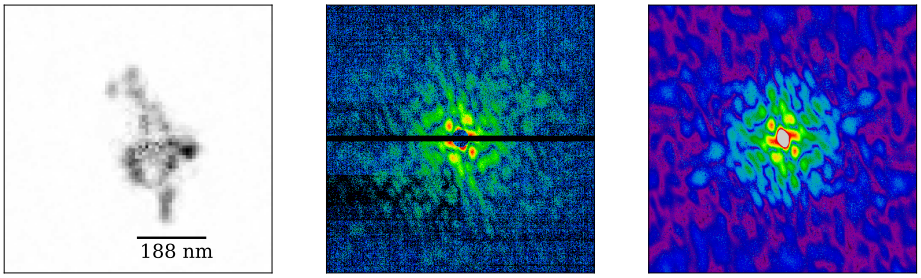


Figure 3.4: A final reconstruction Ψ (on the left) obtained from averaging ten acceptable individual reconstructions. The measured diffraction pattern I (in the middle) and reconstructed intensity $|\hat{\Psi}|^2$ (on the right) demonstrate similar speckle structures in the low scattering angle regions, but differ considerably in the higher scattering angle regions.

could vary greatly. Let y_i be an $n \times n$ pixel array of binary values representing an individual support appropriately translated and inverted and $\bar{y} = \sum_i y_i/m$ be the mean of the supports. Translations and inversions can be identified relative to a reference support, which can be any of the m supports, by maximizing the cross-correlation of the individual and reference supports. The % deviation from the mean,

$$s_i = \frac{||y_i - \bar{y}||}{||\bar{y}||} \times 100, \tag{3.14}$$

where $||\cdot|| = \sum_{j,k} |(\cdot)_{jk}|$ is summed over pixels, also known as the L_1 norm, is obtained for each reconstruction and ordered from least to greatest. In the event all m supports are similar, all s_i's will generally be small. When some supports differ greatly from the majority, however, the % deviations will increase across all i's due to the inclusion of those dissimilar supports in computing the mean. To mitigate the effects of these inflated % deviations, a new mean $\bar{y}_{new}$ is computed based on the $m/2$ individual reconstructions with the lowest s_i's, which we consider to be the similar reconstructions. New % deviation values s_i's are then computed for each reconstruction. An absolute rejection criterion,

$s_{max} = 5\%$, is set such that all reconstructions with $s_i > s_{max}$ are rejected, as shown in Figure 3.3.

As subtle differences in shape and density can exist between similar individual reconstructions, these subtleties can be averaged away by adding the reconstructions. The approach taken in this paper averages $m = 10$ reconstructions from different phasing runs with random initial contrasts and different initial circular supports of varying radii. For dissimilar reconstructions in the collection, new reconstructions are attempted using the same initial circular support with which the reconstruction began but with a different initial contrast. After the new reconstructions are obtained, the s_i's are computed again for the set of ten individual reconstructions consisting of the new as well as the previously unrejected reconstructions, and those failing to meet the rejection criterion are again discarded. This method is repeated until all the s_i's fall below s_{max}. Once ten acceptable reconstructions are obtained, they are then averaged to obtain a final reconstruction, as shown in Figure 3.4.

3.3.4 Missing data

Diffraction patterns will often have significant regions of missing data mainly due to the gap between the CCDs and pixel saturation. When using information about the particle's support in phase retrieval, these missing data regions can be problematic as they could give rise to unconstrained modes, which are spurious features with enough power to exist in the support in real space and the missing data region in Fourier space [40]. Depending on the size of the missing data regions relative to the speckles, unconstrained modes could become problematic

as they superimpose themselves over the true particle contrasts and result in inaccurate reconstructions.

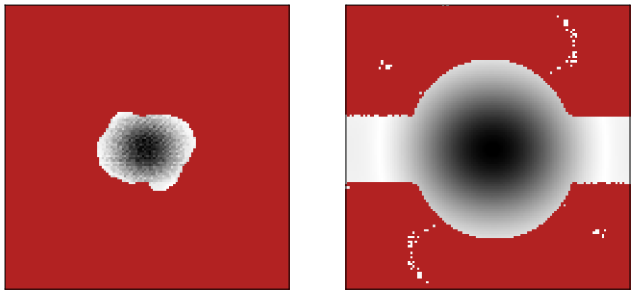


Figure 3.5: A weakly constrained feature f in real space, shown in greyscale, with most of its power contained within the support, regions not colored in red (left). In Fourier space, the same feature, again shown in grayscale, has most of its power contained within the missing data region, again regions not colored in red (right).

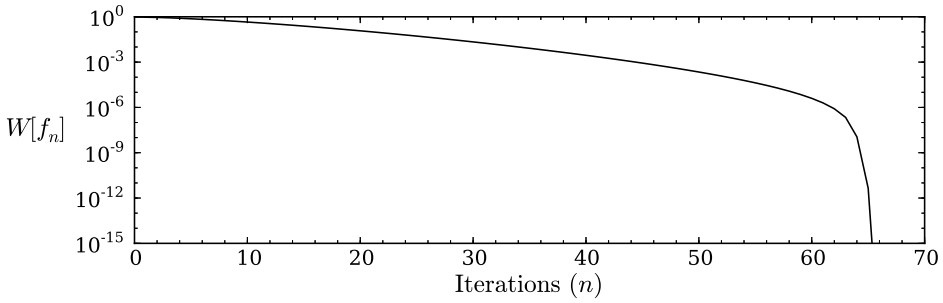


Figure 3.6: The power of an unconstrained feature as it is iteratively updated via the variation of the modified difference map with S and M from Figure 3.5. The feature's power decreases by six decades in ~ 60 iterations before it abruptly falls effectively to zero. This suggests any unconstrained features that arise during the reconstruction process will effectively be suppressed if the time scales of their decay are much less than the time scales of the overall reconstruction process.

The degree to which modes may be unconstrained can be measured by the rate at which they lose power during the phase retrieval process. Given some unconstrained feature in real space f we define its unconstrained power to be

$$W[f] = \frac{1}{2}\left(\int_S |f|^2\, dr + \int_M |\hat{f}|^2\, dq\right) \tag{3.15}$$

where S is the (previously defined) particle's support in real space and M is the missing data region in Fourier space.

Measuring the degree to which these features are constrained can be done by separately running a variation of the phase retrieval process. We define the missing data projection,

$$P_M[\Psi] = \mathcal{F}^{-1} \circ S_M \circ \mathcal{F}[\Psi] \tag{3.16}$$

where

$$S_M[\hat{\Psi}] = \begin{cases} \mathrm{Re}[\hat{\Psi}(q)] & q \in M \\ 0 & q \notin M \end{cases} \tag{3.17}$$

and substitute the Fourier projection P_F with P_M in the modified difference map, while keeping everything else, such as the α parameter, the same. Beginning with a random initial contrast f_0, the modified difference map with the missing data projection will search for features whose power are not constrained within the S and M in real and Fourier space, respectively. When there are no modes with significant unconstrained power, we expect any initial contrast to decay quickly when the above scheme is iterated.

The rate of power loss gives a sense of how constrained the features are during the phase retrieval process and of how severely they could distort the final reconstruction. For power loss as shown in Figure 3.6, the decrease of six decades in about sixty iterations followed by the abrupt drop to zero in total power suggests that the support and missing data regions are too restrictive in allowing any significant unconstrained features to persist. For phase retrieval runs consisting of thousands of iterations, it can be expected that features as

those described in Figure 3.5 will not contribute significantly to the final reconstruction's total power.

3.4 Results

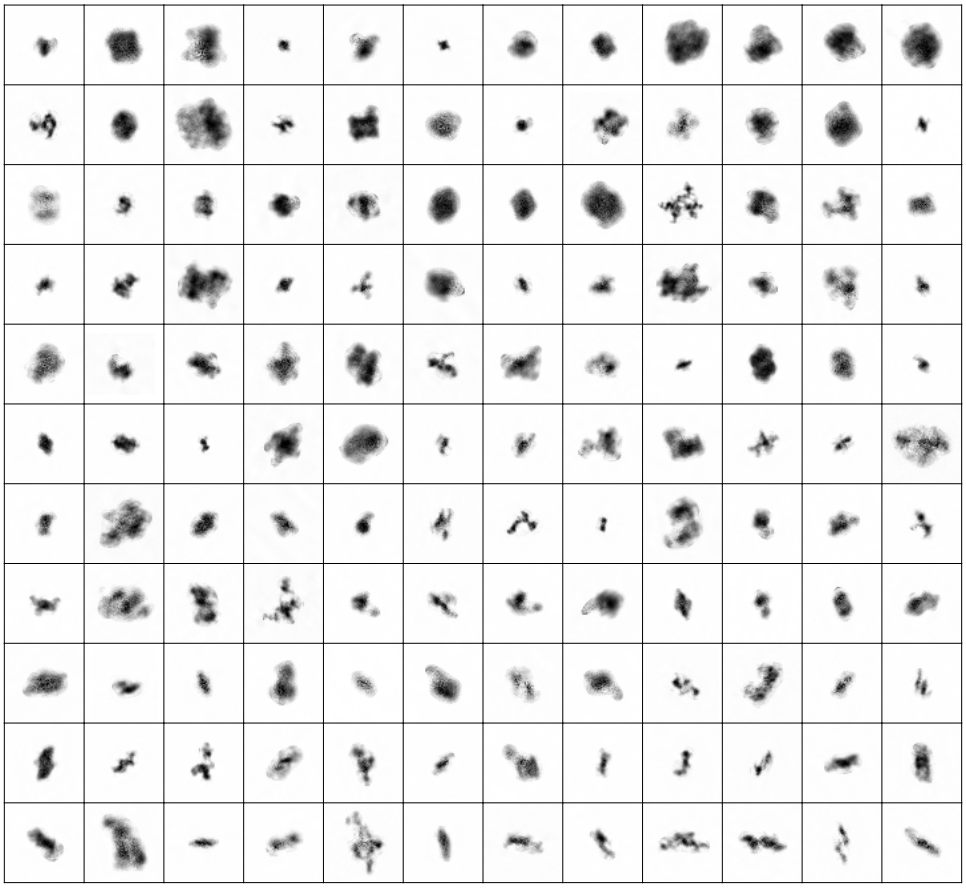


Figure 3.7: A selection of reconstructed soot contrasts, arranged by increasing shape eccentricity. The length of each square box is 573 nm.

The reconstruction process consists of the following steps: centering the diffraction patterns, generating ten acceptable individual reconstructions, and checking whether unconstrained modes and features could exist in the reconstructions. All computations were performed on a standard desktop computer equipped with a quad-core Intel i7-2600 with a clock cycle of 3.4 GHz and 8 GB

RAM. Each individual reconstruction run with $t_{max} = 2000$ iterations took approximately 15 minutes on a single thread. By taking advantage of multithreading, up to five threads ran simultaneous individual reconstructions, shortening the computation of ten individual reconstructions to a minimum of 30 minutes. A maximum of four attempts were made with each initial support. In the event an individual reconstruction attempt was rejected a fourth time, the whole reconstruction process was deemed a failure.

The diffraction images recorded at the LCLS underwent preprocessing where the running background was subtracted. They were then subjected to an intensity-based thresholding routine to identify those that contained sufficient photon signal likely to result from particle-pulse diffraction events and did not exhibit pixel saturation effects. A collection of 953 patterns was generated, and 309 of those patterns were chosen for phasing through visual inspection based on the size of the speckles and good signal-to-noise ratio. An investigation on unbiased pattern selection is underway.

The top and bottom halves of the patterns were added after centering to increase the signal-to-noise ratio and to constrain the contrasts to be real, assuming the patterns largely obeyed centrosymmetry. Given the maximum scattering angle, $\theta_{max} = 0.075$ rads, the condition for centrosymmetry as described in Equation 3.2 to hold requires that the distance from the center of the detector to the edge not exceed 13 speckles. A number of patterns did exceed that count by a couple of speckles, but in most of those cases, noise made it difficult to clearly discern any speckles close to the edge, making the effective maximal scattering angle less than what the detector allows for.

Of those chosen, 36 patterns failed to produce 10 similar individual recon-

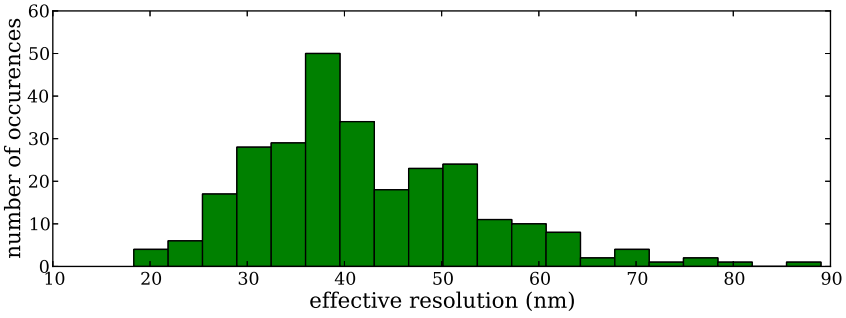


Figure 3.8: Histogram of the effective resolution of the 273 reconstructions, quantified by where the phase retrieval transfer function dips below $1/e$. The smallest effective resolution was determined to be 18 nm, and the largest was 89 nm.

structions. In 30 out of those 36 instances, at least 8 individual reconstructions were deemed similar. A total of 273 patterns yielded averaged reconstructions, and some these reconstructions are shown in Figure 3.7. The quality of the reconstructions was assessed by computing the phase retrieval transfer function (PRTF) and an effective resolution was characterized by where the PRTF drops to $1/e$ [26]. There was great variability in the quality of the reconstruction, as shown in Figure 3.8, with a resolution range of 18 nm to 89 nm.

Many of the diffraction patterns were shifted by various amounts as shown in Figure 3.9. A considerable number of them demonstrated shifts as much as 4 pixels and only 19 patterns were unshifted by the pulse. None of the averaged reconstructions had significant missing data problems as unconstrained features all experienced power decay to zero when run through the procedure outlined in Subsection 3.2.4 using averaged supports. Only in 3 cases did the decay take over 100 iterations. Even then, the longest time it took for complete power loss was 258 iterations. Since the overwhelming majority of patterns chosen had speckles larger than the missing data region, which was indirectly a consequence of Equation 3.1, this was to be expected.

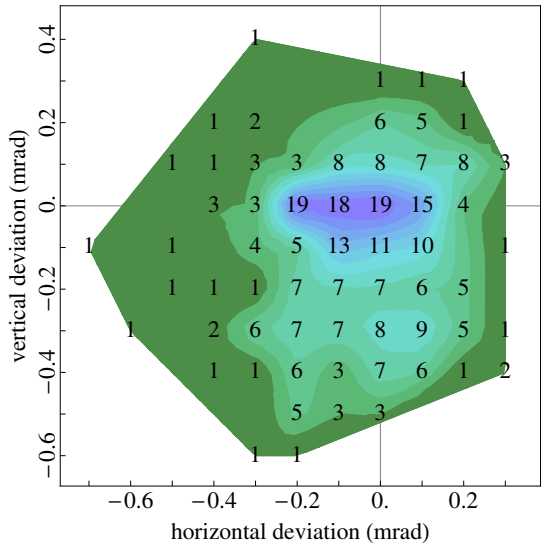


Figure 3.9: 2D histogram of the offsets, measured as outlined in Section 3.1, in the 309 patterns due to random phase tilts in the X-ray wavefront. The distribution of offsets displays a strong spread in horizontal deviations, particularly those with no vertical deviations.

3.5 Conclusion

Data collection rate at facilities such as the LCLS makes it infeasible to carry out a user guided reconstruction for each diffraction pattern. The ability to analyze and extract meaningful results from single shot diffraction imaging experiments will invariably require a speedy and reliable contrast reconstruction process, ideally with no supervision. We presented measures aimed at facilitating data processing and the contrast reconstruction steps, and they have shown that high throughput, unsupervised reconstructions are possible. A desktop implementation of our methods quickly reaches its computational limits, however, and orders of magnitude speedup is necessary for the data collection and processing rates to reach parity. Graphical processing units (GPUs) [41] and other “multi-core" computing solutions show promise in providing the necessary speedup.

The ability to generate a large collection of images via single shot diffraction

imaging enables the possibility for morphological studies analogous to those performed on collections of images obtained through other imaging techniques such as transmission electron microscopy. Diffraction imaging has an advantage over those imaging techniques as it allows for observation *in situ* of airborne particles such as soot. A whole host of other aerosols, such as medicinal nanoparticles to cloud seeds, could benefit from study via single shot diffraction imaging as their airborne structures could yield new insight into their function.

3.6 Acknowledgments

Experiments were carried out at the LCLS, a national user facility operated by Stanford University on behalf of the U.S. Department of Energy (DOE), Office of Basic Energy Sciences. We acknowledge support by the following: DOE grant DE-FG02-11ER16210 (H. J. Park, V. Elser); Human Frontier Science Program (N. D. Loh, M. J. Bogan); AMOS program within the Chemical Sciences, Geosciences, and Biosciences Division of the Office of Basic Energy Sciences, Office of Science, U.S. DOE (N. D. Loh, R. G. Sierra, C. Y. Hampton, D. Starodub, and M. J. Bogan); the Max Planck Society for funding the development and operation of the CAMP instrument within the ASG at CFEL; the Hamburg Ministry of Science and Research and Joachim Herz Stiftung as part of the Hamburg Initiative for Excellence in Research (LEXI); the Hamburg School for Structure and Dynamics in Infection; the Swedish Research Council, the European Research Council, and Knut och Alice Wallenbergs Stiftelse. Part of this work was performed under the auspices of the US Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Support for M. Frank, G. Farquar, W.H. Benner, S. Hau-Riege was provided by the UCOP Lab

Fee Program (award no. 118036). The Max Planck Advanced Study Group at CFEL acknowledges technical support by R. Andritschke, K. Gärtner, O. Hälker, S. Herrmann, A. Hömke, Ch. Kaiser, K.-U. Kühnel, W. Leitenberger, D. Miessner, D. Pietschner, M. Porro, R. Richter, G. Schaller, C. Schmidt, F. Schopper, C.-D. Schröter, Ch. Thamm, A. Walenta, A. Ziegler, and H. Gorke. We thank the staff of the LCLS for their support in the experiments that provided the data for this study.

Part II

Protein structure prediction

CHAPTER 4

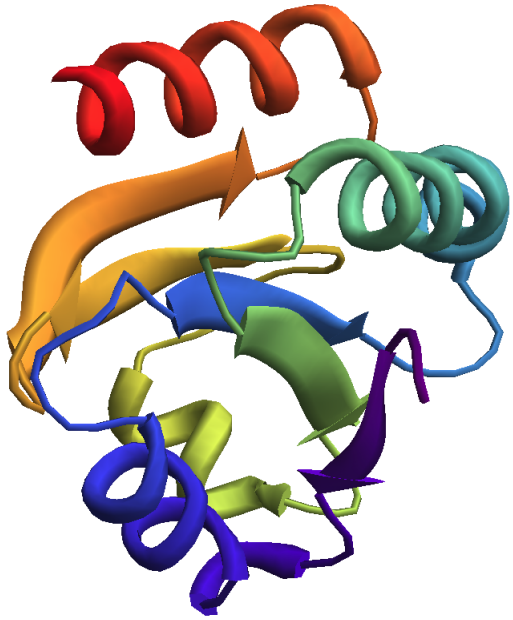
ITERATIVE METHOD FOR PROTEIN STRUCTURE PREDICTION

The contents of this chapter are based in part on unpublished work by Veit Elser [42].

4.1 Introduction

As proteins are some of the most studied molecules due to their wide range of importance in biology, chemistry, medicine, having the ability to better understand their structures has been one of the main motivations for progress in imaging on the nanoscale. Parallel to these direct imaging efforts, there has been a significant push in the past few decades within the computational biology community to study and determine protein structures not through direct observations, but through predicting structures based on sequences of amino acid *residues* as shown in Figure 4.1. These efforts largely fall under what is called the protein structure prediction problem, and to this day it remains a very active field of research.

Broadly speaking, there are two classes of prediction problems [43]. The first class of problems is of cases where sequences of unknown proteins bear similarities to those of known proteins. The known structures serve as templates used to fashion the structures of the unknown cousins. This type of prediction problem is known as comparative modeling. A number of widely used algorithms have been developed over the years, and many of them have shown reliable results for predicting structures of sequences similar to sequences of known structures by as little as 30% [44, 45].



```
VGVKP VGSDP DFQPE LSGAG SRLAV VKFTM RGCGP CLRIA PAFSS MSNKY
PQAVF LEVDV HQCQG TAATN NISAT PTFQF FRNKV RIDQY QGADA VGLEE
KIKQH LE
```

Figure 4.1: A ribbon diagram and the amino acid residue sequence of the protein 1GH2 [46], which is one of many proteins expressed during human fetal brain development. In the structure prediction problem, one aims to determine the three-dimensional structure of the protein (top) using its amino acid residue sequence (bottom).

The other class of problems deals with cases where the unknown sequences do not share similarities with known proteins, so templates cannot be used to predict their structures. This type of problem is known as *ab initio* or *de novo* protein modeling. Because these problems deal with unknown sequences with no known *homologs*, methods for predicting their structures rely on approaches such as "gluing" together smaller structures or letting an initial guess of the structure interact with its environment via physical forces and minimizing energy functions [47, 48]. In general, *de novo* methods require greater computational resources than comparative modeling methods due to the need to explore greater numbers of conformations.

In this chapter, we detail a prediction algorithm of the *de novo* variety which makes use of constraint satisfaction via iterating projections as described in Chapter 2. Given a sequence of residues for a protein of unknown structure, the algorithm identifies subsequences of fixed lengths and draws upon collections of known structures to best predict the structures of the subsequences. The collections are generated by parsing known protein structures found in a widely used, online repository called the Protein Data Bank (PDB) [49]. Work towards building a comprehensive collection is discussed in Chapter 5.

4.2 A primer on proteins

Proteins are important biological molecules whose functions are numerous and essential for living organisms. Structurally, they are mainly composed of chains of residues from twenty different amino acids listed in Table 4.1. Chains are formed through peptide bonds between consecutive amino acids, and the chains wind and assemble themselves into elaborate structures, like in Figure 4.1.

Table 4.1: A list of the twenty amino acids that form the building blocks of proteins. Their one-letter codes are used to conveniently describe the sequences that make up protein chains.

A	Alanine	G	Glycine	M	Methionine	S	Serine
C	Cysteine	H	Histine	N	Asparagine	T	Threonine
D	Aspartic acid	I	Isoleucine	Q	Glutamine	V	Valine
E	Glutamic acid	K	Lysine	P	Proline	W	Tryptophan
F	Phenylalanine	L	Leucine	R	Arginine	Y	Tyrosine

Amino acids have a general structure that follow the form shown in Figure 4.2. They are composed of three parts: the amino group, the carboxyl group, and a side chain, all of which are bonded to a central carbon atom called C_α. Amino acids bond with other amino acids via condensation, where the amino

group of one amino acid and the carboxyl group of the other amino acid come together to form a peptide bond between the amino acid residues, as shown in Figure 4.3. These bonds serve as the basis for the creation of chains.

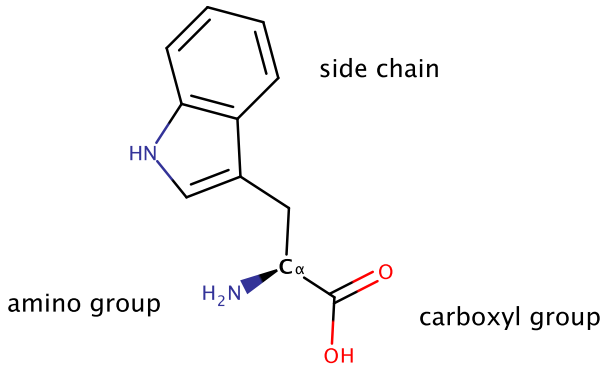


Figure 4.2: A chemical diagram of tryptophan, one of the twenty amino acids that serve as building blocks for proteins. Amino acids generally have the same chemical structure, where a carboxyl group, an amino group, and a side chain are all joined together via the C_α atom. The figure was generated using MarvinSketch version 15.16.8 (2015) [50].

The peptide bonds impose both a periodicity and a directionality on the *backbone* of a chain. The backbone is made up of repeating units of NH - C_αH - CO, so all proteins conventionally "begin" at the N atom in the unbonded amino group (the N-terminus), and "end" at the C atom in the unbonded carboxyl group (the C-terminus). A convenient, visual way to represent the backbone is through the planar configuration shown in Figure 4.3. The C_α atoms on consecutive residues, as well as the remaining oxygen atom from the carboxyl group and the hydrogen atom from amino group form a quadrilateral which contains the peptide bond. On longer chains, the backbone can be represented via a series of connected quadrilaterals.

While getting a handle on how the global structure of a chain (also called *tertiary structure*) arises is challenging, understanding how local structures arise can be done mainly through studying the interatomic forces between relevant

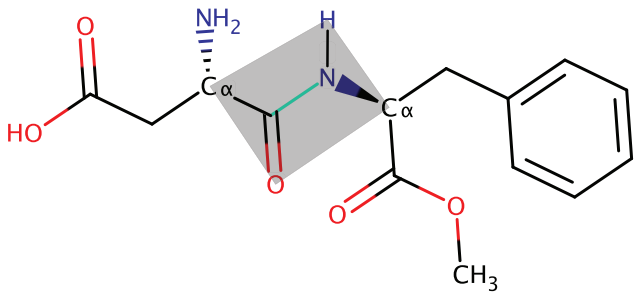


Figure 4.3: The popular artificial sweetener aspartame is a dipeptide composed of two amino acids, aspartic acid (left) and phenylalanine (right). The two amino acid residues are joined via a peptide bond (shown in teal). The grey region, a quadrilateral defined by the four corner atoms, encloses the peptide bond.

atoms. One of the more important forces in local structure determination is the *hydrogen bond*, which is technically not a bond in the electron "sharing" sense, but rather an electrostatic attraction between a hydrogen atom with an electronegative atom such as oxygen. Hydrogen bonds are responsible for the assembly of structures like the coils and the long, tight, winding strands shown in Figure 4.4. These *secondary structures*, called α-helices and β-sheets, are very common structural motifs found in many proteins, and play an important role in our discussions in Chapter 5.

Proteins need not be composed of a single chain, but could be made up of multiple chains, giving rise to *quaternary structures*. Also, they could incorporate other molecules such ligands or nucleic acids into their structures to serve elaborate functions. These additional components only give way to more complexity and further challenge our ability to understand how such structures can arise.

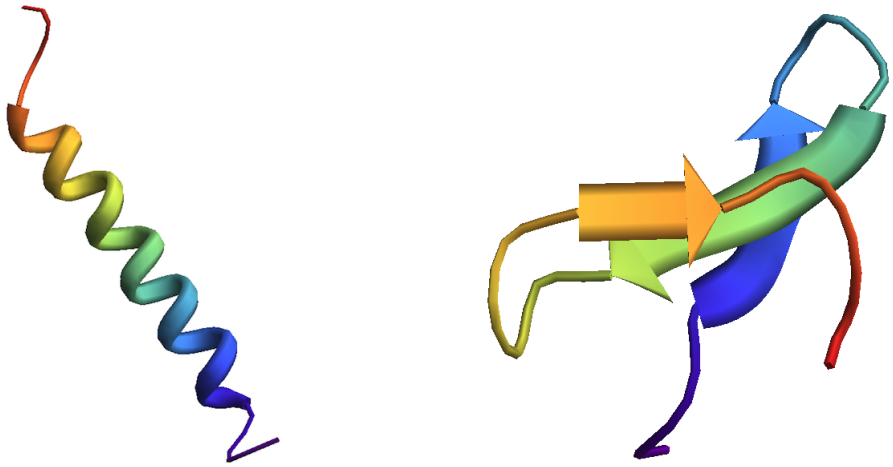


Figure 4.4: Two proteins, 2MVJ [51] (left) and 2MWD [52] (right), composed mainly of secondary structures. 2MVJ can be characterized as a single, long, α-helix, and 2MWD as an anti-parallel β-sheet, shown as a sequence of arrows in close proximity to each other.

4.3 Prediction problem

As previously stated, protein structure prediction is a process where a protein's three dimensional structure is determined from its residue sequence. Conventionally, determining the three-dimensional coordinates of the C_α atoms is sufficient in describing a proteins' structure. The prediction algorithm to be outlined in this chapter has been applied on single chains, and has initially shown success in predicting structures of chains with roughly sixty residues. At the heart of the algorithm is the "divide and concur" formalism [53]. It is a technique which defines two projections that do the following: independently satisfy an arbitrary number of constraints over multiple copies of solution attempts (the divide projection) and reconcile the disagreements between different attempts at satisfying all the constraints (the concur projection). It has been applied to a number of problems, ranging from disk packing to 3SAT, and has performed on par with other algorithms adept at solving these problems. Before discussing

how this method can be applied to the protein prediction problem, we first discuss the method in generality.

4.3.1 Divide and concur

Divide and concur is a constraint satisfaction technique which overcomes the two-constraint limit in the methods outlined in Chapter 2 and allows for problems with N constraints to be solved via iterated projections. The way this is done is by reformulating problems in a way that requires the use of two particular projections, called the divide and concur projections.

The first of the projections involves satisfying each of the N original constraints independently, without regard for how satisfying any one will affect satisfaction of the others. To accomplish this, a satisfaction problem, whose solution usually resides in some space K, is reformulated by constructing elements in a space consisting of N "copies" of K. The space where the "meta-solution" resides in is the N-fold Cartesian product space of K's. An element of this product space can be described as

$$\mathbf{y} = \mathbf{x}^{(1)} \times \mathbf{x}^{(2)} \times \ldots \times \mathbf{x}^{(N)}. \tag{4.1}$$

where each $\mathbf{x}^{(i)}$, an element of K, addresses a single constraint. This construction allows for each of the N constraints to be satisfied on each of the elements in the Cartesian product without care for how the other constraints are satisfied.

To perform the independent satisfactions, we define the divide projection, P_D, which takes elements of the Cartesian product space as inputs and outputs

modified elements residing in the same space. Each of the constraints are individually and independently satisfied via their own projections, P_i, on each of the N elements of $\mathbf{y}$, through the following operation

$$P_D[\mathbf{y}] = P_1[\mathbf{x}^{(1)}] \times P_2[\mathbf{x}^{(2)}] \times \ldots \times P_N[\mathbf{x}^{(N)}]. \tag{4.2}$$

The second projection is meant to complement the first and impose consensus among all the different attempts at individual constraint satisfaction. We define the concur projection, P_C, which takes N elements of $\mathbf{y}$ and brings all of them to agreement without care for whether or not doing so compromises the individual constraints. Mathematically, this is accomplished by first computing the weighted average

$$\bar{\mathbf{x}} = \frac{\sum_{i=1}^{n} \lambda_i \mathbf{x}^{(i)}}{\sum_{i=1}^{n} \lambda_i} \tag{4.3}$$

and returning the N-fold Cartesian product with each element consisting of Equation 4.3

$$P_C[\mathbf{y}] = \bar{\mathbf{x}} \times \bar{\mathbf{x}} \times \ldots \times \bar{\mathbf{x}}. \tag{4.4}$$

With these two projections defined, a solution which satisfies all N constraints can be thought to lie at the intersection of the divide and concur constraint sets. As in the case of phase retrieval, the meta-solution $\mathbf{y}_s$ which lies at the intersection of the two constraint sets satisfies the equation

$$\mathbf{y}_s = P_D[\mathbf{y}_s] = P_C[\mathbf{y}_s]. \tag{4.5}$$

Finding an intersection of the two sets can be done via iteratively applying a

composition of these projections, much like the approach discussed in Chapter 2. This will be discussed in detail in Subsection 4.3.5.

4.3.2 Structure prediction via divide and concur

To apply divide and concur to the protein structure problem, the prediction algorithm divides up a sequence of m amino acid residues into smaller, overlapping subsequences of length p. It also begins with a random guess for what the three-dimensional structure of the sequence looks like. Each subsequence has a library of diverse three dimensional structures at its disposal, from which it finds a particular structure that the current three-dimensional structure characterizing the subsequence most "agrees" with, and then replaces the current structure with the most agreeable one. In this step, the subsequence also considers interactions between itself and residues as well as water in choosing its preferred three-dimensional conformation.

Finding the "ideal" three-dimensional structure for a subsequence can be thought of as a single constraint which needs satisfying, the details of which will be described in the next subsections. This is done via defining a projection for that particular subsequence which does what was described in the previous paragraph. The divide projection will take the projections on all the subsequences and apply them independently in each application of P_D, as suggested in Equation 4.2. Complementing this effort is the concur projection, which reconciles the different copies via averaging.

4.3.3 Subsequences and "litemotifs"

Consider a single sequence of length m. We define subsequences to be of length p and note that there are $m-p+1$ possible subsequences to consider on a single sequence if we allow for overlaps. If we again consider the single chain protein 1GH2 from Figure 4.1, we see it has a sequence consisting of $m = 107$ residues. If we set $p = 4$, then the sequence has a total of 104 potential subsequences of interest. Each subsequence in 1GH2 has a particular residue *code*. For instance, the third subsequence of length 4 has the code 'VKPV'. To predict the three-dimensional structure for this particular subsequence, a collection of known three-dimensional structures from subsequences of length 4 with the same code is needed.

When considering the three-dimensional structures of subsequences of a particular code, there are bound to be numerous different structures due to configurations resulting from interactions between the residues in the subsequence and residues elsewhere on the chain or water molecules. It is important to keep track of these additional residues or water molecules. This motivates the notion of *litemotifs*, which serve as "building blocks" for the prediction algorithm. A litemotif is defined to be a three dimensional configuration of p C_α atoms from a subsequence of interest plus a set of q other atoms which the residues of the C_α atoms are in contact with.

Generating collections of litemotifs requires finding them from known protein structures. The PDB is an online repository for known protein structures with a collection of 109,457 molecular structures as of mid-2015 [49]. These structures are obtained via numerous experimental techniques, from X-ray crystallography to nuclear magnetic resonance (NMR). For our purposes, it is one

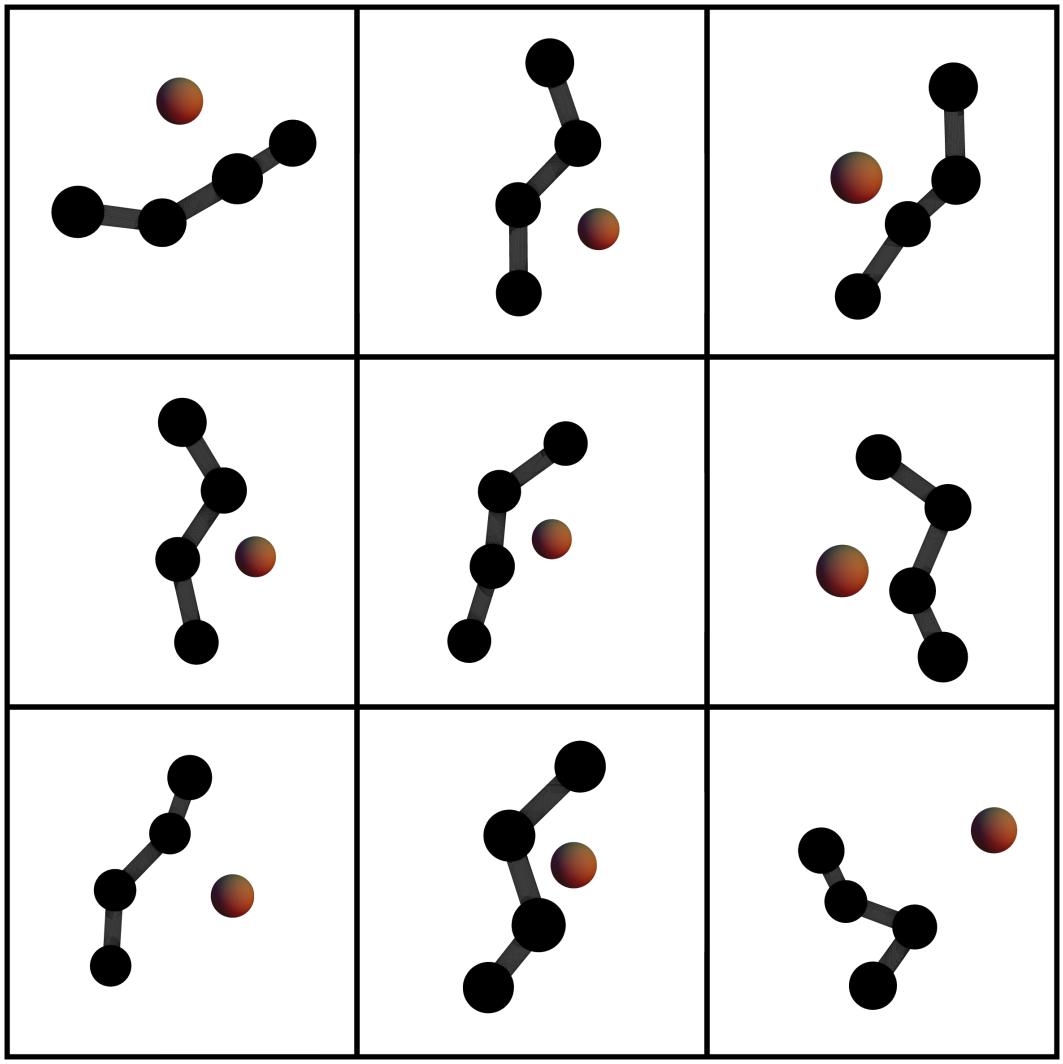


Figure 4.5: Some examples of litemotifs. They consist of $p = 4$ C_α atoms, depicted in black, and $q = 1$ contact atoms, depicted in orange, which could either be another C_α atom or an oxygen atom from a water molecule. These were extracted from protein structures deposited in the PDB.

of the most comprehensive collections of protein structures from which we can extract litemotifs. Figure 4.5 shows a small selection of litemotifs extracted from a number of different proteins from the PDB.

4.3.4 Divide and concur projections

Each of the subsequences in a chain, as previously stated, is subject to a single constraint: it must match some litemotif. Let us suppose that there are $N = m - p + 1$ subsequences of interest. There can be cases when certain subsequences can be left out of consideration, but we shall assume that is not the case. Furthermore, for simplicity's sake, let us consider cases where subsequences are of length $p = 4$ and a subsequence is in contact with $q = 1$ atoms, implying $N = m - 3$. Lastly, we leave out water molecules in our discussions, but they could be incorporated as additional atoms whose coordinates need to be determined.

We first consider the space in which the solution and "meta-solution" reside in. As in Equation 4.1, we define N copies of the set of atoms characterizing the protein structure via the Cartesian product

$$\mathbf{y} = \mathbf{x}^{(1)} \times \mathbf{x}^{(2)} \times \ldots \times \mathbf{x}^{(N)} \tag{4.6}$$

where $\mathbf{x}^{(i)} = \{x_{i,1}, \ldots, x_{i,m}\}$ is a set of m three-dimensional coordinates for all the m C_α atoms, as shown in the simplified representation in Figure 4.6. The (i) superscript is meant to denote the ith copy on which the ith subsequence is modified in the divide step.

The projection $P_j[\mathbf{x}^{(j)}]$ replaces the jth subsequence and contact atom with an element from some litemotif collection L_j. Suppose $L_j = \{l_1, \ldots, l_\mu\}$ has μ different litemotifs, where each l_i is composed of $p = 4$ three-dimensional coordinates of C_α atoms and $q = 1$ additional coordinates to describe the contact

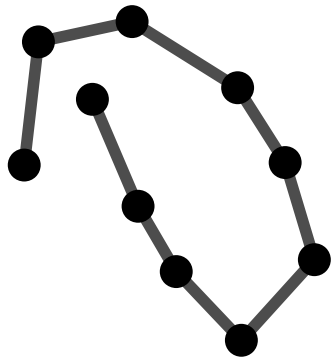


Figure 4.6: A two-dimensional diagram depicting an initial guess, $\mathbf{x}^{(i)}$, at a solution for the C_α chain. A candidate "meta-solution" can be fashioned from this guess by constructing an N-fold Cartesian product by making N copies of $\mathbf{x}^{(i)}$.

atoms. More specifically, each l_i can be expressed as a set of coordinates:

$$l_i = \{z_{i,1}, \ldots, z_{i,4}, z_{i,5}\} \tag{4.7}$$

where the first four $z_{i,k}$'s are sets of coordinates from a subsequence of a known structure and $z_{i,5}$ is the coordinate of the contact atom.

The projection goes through L_j and finds the litemotif, $l_c = \{z_{c,1}, \ldots, z_{c,4}, z_{c,5}\}$ that best aligns with the coordinates $x_{j,j}, \ldots, x_{j,j+3}, x_{j,j'}$ in $\mathbf{x}^{(j)}$, where j' is the index of the contact atom. Since j' is not known, the projection loops through all indices except for those in the subsequence and chooses the index that best aligns a fixed litemotif with the five positions in $\mathbf{x}^{(j)}$, as shown in Figure 4.7. It then does this for all litemotifs until l_c and the best contact position for l_c, j'_c, are found. Alignment is measured using the root mean squared deivation after aligning the two sets of atoms' coordinates via least-squares minimization. The details of the RMSD and alignment calculations can be found in Appendix A. With the litemotif l_c and the best contact index j'_c found, the projection modifies

$\mathbf{x}^{(j)}$ by replacing the coordinates $x_{j,j}, \ldots, x_{j,j+3}, x_{j,j'_c}$ with $z_{c,1}, \ldots, z_{c,4}, z_{c,5}$.

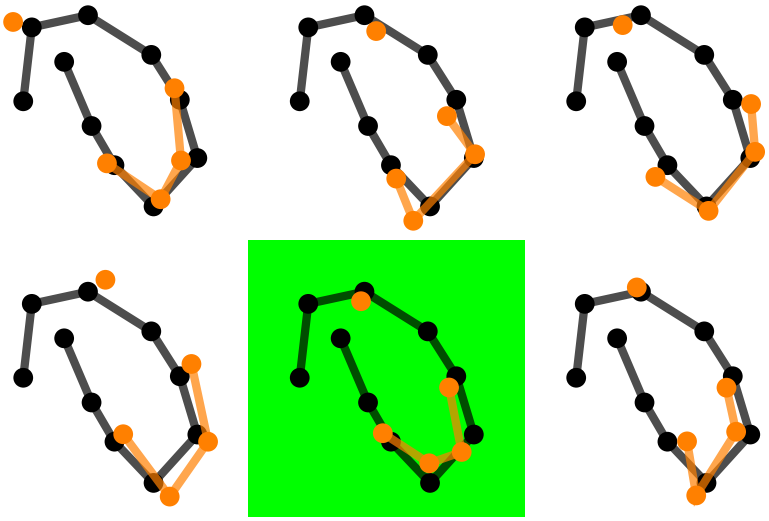


Figure 4.7: An individual projection, $P_j[\mathbf{x}^{(j)}]$, will compare the j'th subsequence against all the litemotifs, depicted in orange, found in the collection L_j. It then finds l_c, the litemotif most similar to $x_{j,j}, \ldots, x_{j,j+3}, x_{j,j'_c}$, and replaces the latter with the former. In this case, the litemotif in the green square is deemed most similar to the subsequence and contact atom.

With the individual projections P_j's defined, the divide projection $P_D[\mathbf{y}]$ applies them on each element of the Cartesian product as indicated in Equation 4.2. Each element will only have one subsequence and contact atoms replaced by a litemotif, while leaving the other coordinates untouched, as depicted in Figure 4.8.

The concur projection, $P_C[\mathbf{y}]$ is a straightforward application of Equation 4.4. This step involves taking all N copies of the three-dimensional coordinates and averaging them to produce one set of coordinates

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)} = \left\{\frac{1}{N}\sum_{i=1}^{N}x_{i,1}, \ldots, \frac{1}{N}\sum_{i=1}^{N}x_{i,m}\right\}. \tag{4.8}$$

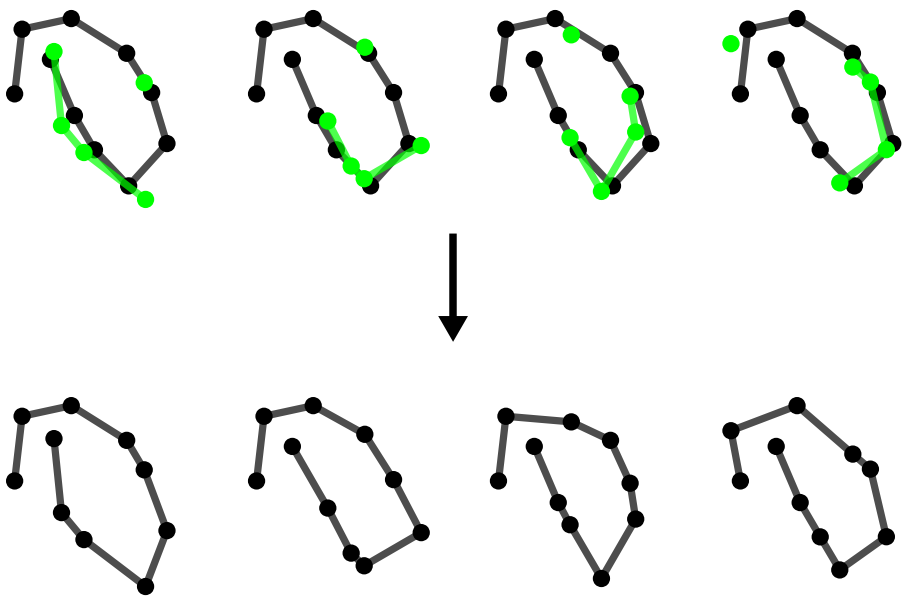


Figure 4.8: The divide projection, $P_D[\mathbf{y}]$, applies the individual projections $P_j[\mathbf{x}^{(j)}]$ over all j subsequences and returns a Cartesian product of the N modified sequences. On the top row, the green circles represent the most favored litemotifs from each of the individual projections. We only demonstrate the divide projection applied on four subsequences in this figure.

The projection then returns a Cartesian product of the averaged coordinates

$$P_C[\mathbf{y}] = \bar{\mathbf{x}} \times \bar{\mathbf{x}} \times \ldots \times \bar{\mathbf{x}}. \tag{4.9}$$

as shown in Figure 4.9.

4.3.5 Iterated projections and ADMM

Like the phase problem, using iterated projections to predict protein structures follows the same general set of procedures, where a random initial iterate $\mathbf{y}_0$ is updated through a composition of the divide and concur projections. Any of the algorithms described in Table 2.1 could conceivably be used, and the iterations

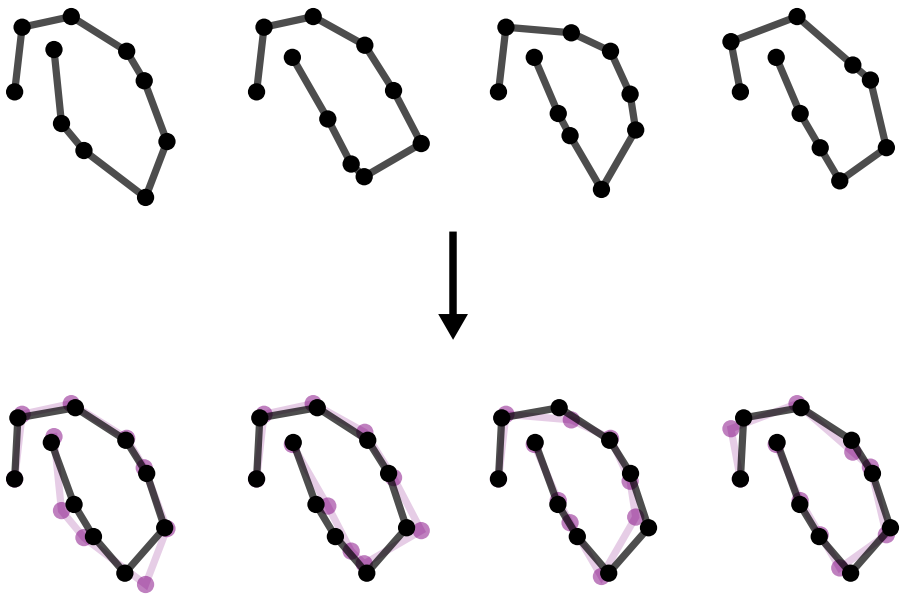


Figure 4.9: The concur projection, $P_C[\mathbf{y}]$, computes the average of the N different subsequences and returns an N-fold Cartesian product consisting of the average. The purple shadow demonstrates how much each of the original sequences differs from the averaged sequence.

would stop when the error metric (which in this case is equivalent to the RMSD between iterates) $\epsilon[\mathbf{y}_n]$ satisfies the inequality

$$\epsilon[\mathbf{y}_n] < \tau \tag{4.10}$$

for some tolerance τ. As was the case with the phase problem, there is no convergence guarantee for the prediction algorithm because the divide projection is non-convex. Hence, the number of iterations required until Equation 4.10 is satisfied is determined through trial and error.

Recent efforts have used a different algorithm than the ones listed in Table 2.1. The algorithm of choice has been the Alternating Direction Method of Multipliers (ADMM), a method developed by Boyd *et al.* [54] for solving convex minimization problems. In the context of iterated projections, the method can

be described by the following sequence of steps

$$\mathbf{d}_n = P_D[\mathbf{c}_n + \alpha \mathbf{f}_n] \tag{4.11}$$

$$\mathbf{c}_{n+1} = P_C[\mathbf{d}_n - \alpha \mathbf{f}_n] \tag{4.12}$$

$$\mathbf{f}_{n+1} = \mathbf{f}_n + \mathbf{c}_{n+1} - \mathbf{d}_n \tag{4.13}$$

where α is a free parameter and the algorithm is started with initial seeds $\mathbf{c}_0$ and $\mathbf{f}_0 = 0$.

In one round of ADMM, the divide and concur projections are applied sequentially on iterates modified by the *discrepancy* $\mathbf{f}_n$ scaled by α. Depending on the choice of α, ADMM is equivalent to some of the other methods listed in Table 2.1. When $\alpha = 0$, ADMM reduces to the alternating projections method, with the discrepancy not playing any role in how the iterates $\mathbf{d}_n$ and $\mathbf{c}_n$ get updated. For $\alpha > 0$, the discrepancy is "fed" into the next round of alternating projections as a way to prevent stagnation, similar in spirit to the HIO method. The difference between consecutive $\mathbf{f}_n$'s can be used to compute the error metric, and when Equation 4.10 is satisfied, the concur iterate $\mathbf{c}_n$ can be used to obtain the solution.

4.4 Preliminary results

The first attempts at structure prediction used "4+1" litemotifs, which are composed of four consecutive C_α atoms and an additional contact atom, which is either another C_α atom or an oxygen atom from a water molecule. Contact is defined via some distance cutoff d_{co}, and is between side chains and other side

chains or water molecules. To ensure the geometry of the 4 C_α atoms is heavily influenced by the contact atom, at least two of the side chains of the corresponding C_α atoms have to meet the distance cutoff with the contact atom. The litemotifs shown in Figure 4.5 were constructed using these rules.

The choice of $p = 4$ was made so that the litemotifs are small enough that many samples could be found in the PDB, but also large enough that they capture the geometrical diversity. Smaller litemotifs of length $p = 3$ would mainly capture how the C_α atoms span a plane, but fail to capture the coils and kinks of the backbones of the proteins they come from.

A total of 20,148 proteins were processed from the PDB to generate 23,593,246 different litemotifs with the help of the ProDy package [55]. For each litemotif, the four-letter code was recorded, so that "subcollections" of litemotifs consisting only of those with the same codes can be constructed. There are 11,505 different codes in the collection, representing a fraction of the 20^4 different possibilities. The subcollection sizes are not uniform, and range from some with hundreds of thousands of litemotifs to others with just one litemotif. The median subcollection size is 831. Table 4.2 lists the codes with some of the smallest and largest subcollections.

Initial prediction efforts centered on single chains of less than a hundred residues in length. Sequences of known proteins, whose structures can be found in the PDB, were tested using litemotif collections which excluded those extracted from the tested protein. Additional constraints were imposed on water molecules, restricting them to the exterior of a compact region occupied by the protein.

Table 4.2: A listing of the codes with the largest and smallest subcollections. Note that there are far many more codes with subcollections of size 1, but only five are listed here.

Largest		**Smallest**	
Code	Size	Code	Size
MRYF	548,854	RITR	1
TQSP	512,101	EQLN	1
NLQG	398,175	IWEV	1
VKFG	184,460	LALA	1
NMKR	157,786	KLVF	1

One of the seemingly first successes came in predicting the structure of 2P5K [56], a single chain protein of sixty-three residues. After roughly 150,000 iterations, the algorithm was able to hone in on a structure which largely did not change, minus rigid rotations, over the next 50,000 iterations. This is evident in the time series of the RMSD between consecutive iterates, also known as the error metric, as shown in Figure 4.10, where the RMSD value fluctuates around a smaller value after 150,000 iterations. The final predicted structure, shown in Figure 4.11, differed from the true structure as reported in the PDB by an RMSD of 2.78Å.

When attempting to predict the structure of the human CD59 glycoprotein, 2J8B [57], another single chain protein but with seventy-eight residues, a problem arose. Much like 2P5K, the algorithm found a structure that it largely settled on after a few hundred thousand iterations, as shown in Figure 4.12. However, after auditing the subsequences to determine from which known proteins the litemotifs responsible for the final structure came from, a large number of them were associated with 1CDQ, 2OFS, 2UX2, and 4BIK. These four proteins are

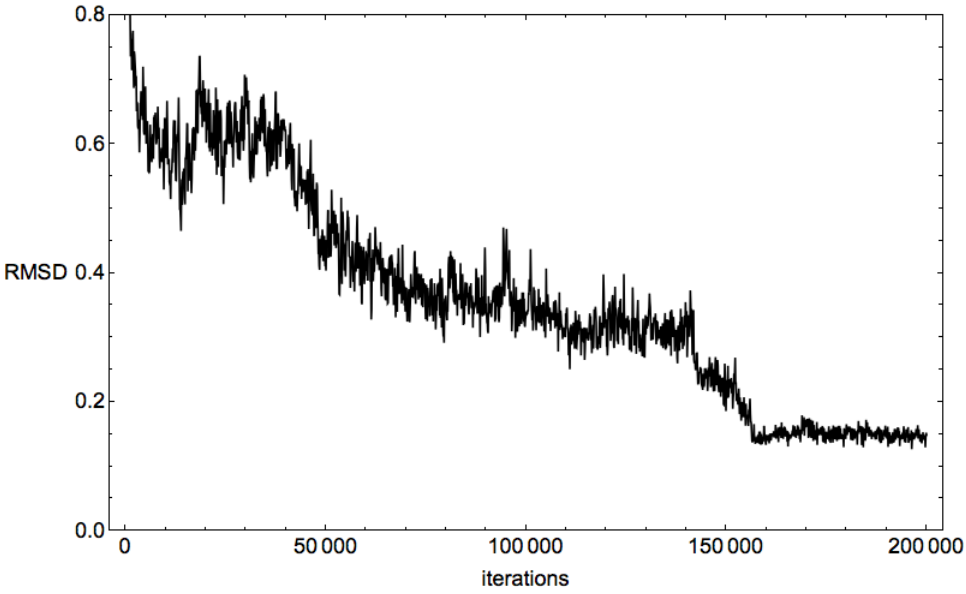


Figure 4.10: The RMSD between consecutive iterates from predicting 2P5K. Note that the value generally decreases until after about 150,000 iterations, after which it largely settles around an RMSD of 0.15Å. Reproduced with permission from the creator [42].

either the same CD59 glycoprotein studied at different resolutions or with different methods, or contain the glycoprotein as part of their larger structures (in the case of 4BIK).

After further pruning the collection of litemotifs to exclude those that come from near-identical proteins, prediction was attempted on 1ULR [58]. In this case, the prediction algorithm failed to find a structure that closely resembles the actual structure. Many of the secondary structures present in the actual version were missing, giving way to erratic coils as shown in Figure 4.13. Note that large swaths of the β-sheets in the actual version are simply not present in the predicted version.

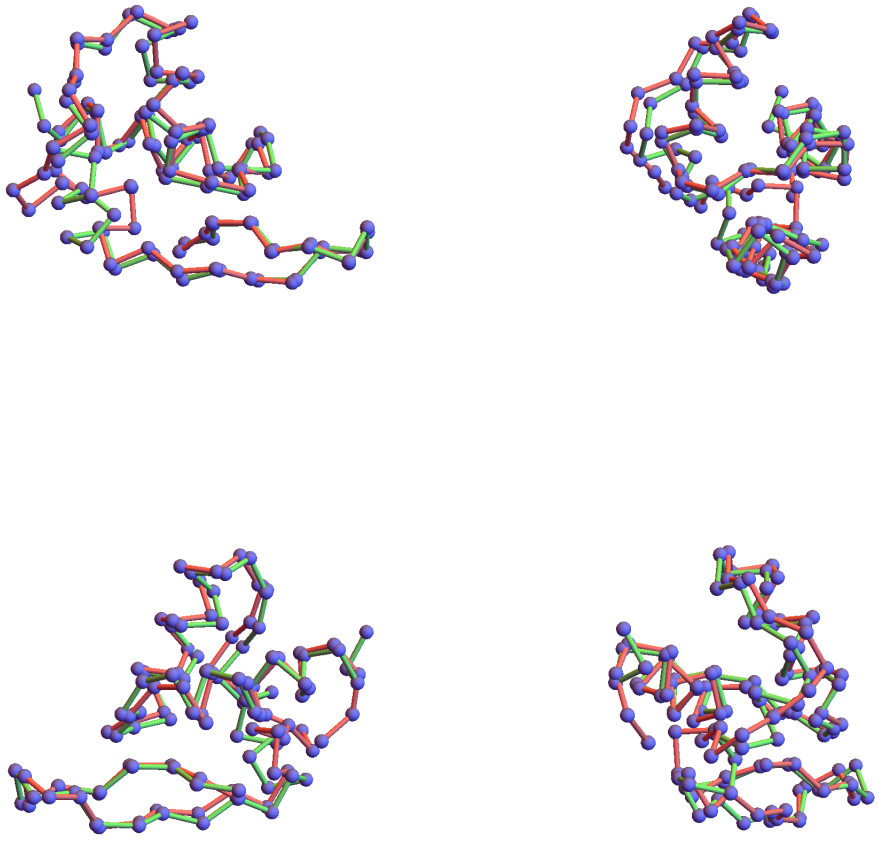


Figure 4.11: Snapshots of the predicted structure of 2P5K, shown in purple with red connections, overlaid on the actual structure, shown in purple with green connections, shown from different angles. The two structures differ by an RMSD of 2.78Å. Reproduced with permission from the creator [42].

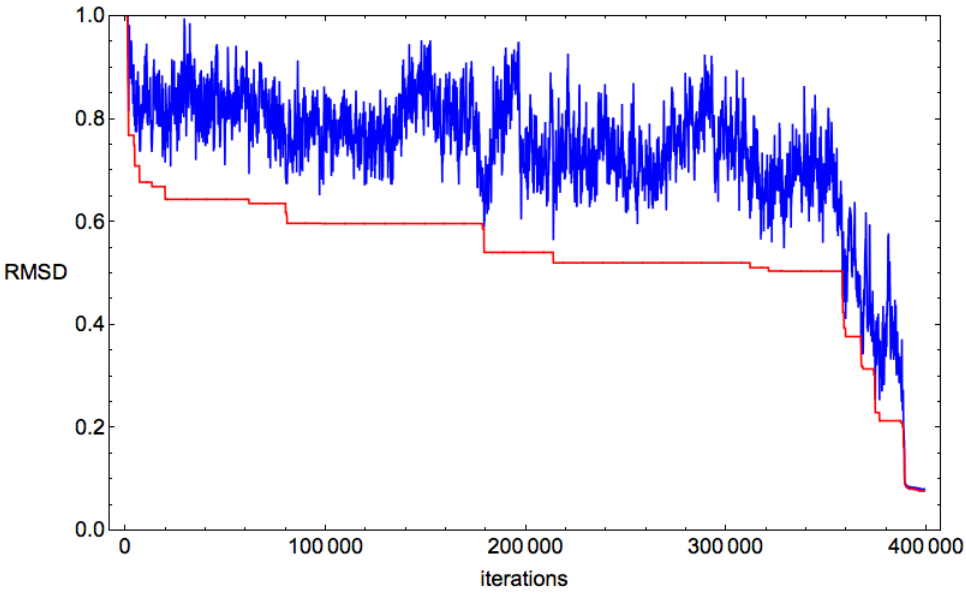


Figure 4.12: The RMSD between consecutive iterates from predicting 2J8B. Much like in Figure 4.10, the value gradually decreases over the course of hundreds of thousands of iterations. The abrupt drop starting from about 340,000 iterations signals that the algorithm is close to finding a solution that satisfies both the divide and concur constraints. Reproduced with permission from the creator [42].

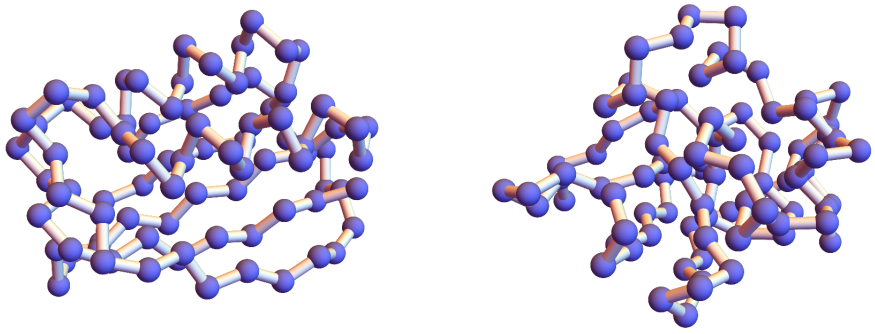


Figure 4.13: The actual structure of 1ULR, shown left, versus the predicted structure shown right. Note that many of the α-helices and β-sheets found in the actual structure are not found in the predicted structure. Reproduced with permission from the creator [42].

4.5 Future work

While the latest attempts exposed certain shortcomings of the prediction strategy using a particular set of litemotifs, it is worth noting that through the various trials, the algorithm has a tendency to find the "right" litemotifs to represent subsequences, despite looking through a large number of them. When given the chance, the algorithm found most of the litemotifs which were associated with near-identical versions of the predicted protein, which is promising if it continues to hold, but ideally on more properly curated collections.

Going forward, the two issues that need addressing are building litemotif collections for testing that exclude those coming from near-identical proteins, and also considering different litemotif constructions that better capture local geometries. Although the "4+1" litemotifs fare reasonably in capturing the backbone shapes, they do not effectively capture the elaborate structures borne from secondary structures, as Figure 4.13 makes clear. Some of these issues are addressed in the next chapter, and further results from these changes will be forthcoming in the near future.

CHAPTER 5

PROTEIN LITEMOTIF GENERATION

To complement the structure prediction method outlined in Chapter 4, protein structures from the PDB were used to extract litemotifs. Depending on the rules of construction, collections with up to millions of litemotifs were generated. These collections represent a "complete" sampling of the possible litemotifs given all the protein structures we know. While it is not entirely unreasonable to take these complete collections and use them without further analysis in the prediction studies, there are two issues that makes this approach problematic. One is that many litemotifs could similar to each other. For such cases, it may be appropriate to only keep one similar litemotif and remove the rest from the collection. The other issue is that there is no way to know if a complete collection contains a diverse enough number of litemotifs.

In this chapter, we discuss efforts to address these two issues. We first define the litemotifs of interest and extract them from the deposited proteins in the PDB. We then quantify the pairwise closeness between the litemotifs and use those values to build graphs where each litemotif is represented by a node. The graph construction allows us to "compress" large collections through finding *dominating sets*, which can loosely be thought to represent non-redundant subcollections. These dominating sets are further analyzed using tools from information theory and dimensionality reduction methods to better understand the "expansiveness" of the collections they represent.

The methods discussed in this chapter are applied to different sets of litemotifs, described in the next section, based on the latest attempts at protein folding via divide and concur. They could easily be applied and studied to conceivably

any collection of litemotifs.

5.1 Litemotif definitions

After initial attempts at structure prediction with the "4+1" litemotifs, there was a desire to build new collections based more firmly on configurations resulting from hydrogen bonds as well as less stringent side-chain to side-chain contacts. It is suspected that litemotifs based on these interactions would better capture local secondary structures as well as residue-water contacts.

In this section, we define four new litemotifs. Unlike the "4+1" motifs from before, keeping track of the code of each of these litemotifs is not required. The use of different litemotifs in tandem in structure prediction through "divide and concur" requires only a slight modification of the divide projection, where the number of individual constraints increases by however many sets of litemotifs are necessary. This would only increase the Cartesian product space by roughly a factor of the number of different sets of litemotifs. The concur projection would remain the same.

As was briefly discussed in the last chapter, litemotifs are extracted from known protein structures deposited in the PDB. The files contain the coordinates for most of the important atoms, save for the hydrogen atoms on the carboxyl groups of residues. The locations of those atoms have to be inferred from their neighbors.

- **"3+3" via hydrogen bonding:**

  The first litemotif consists of a pair of three consecutive C_α atoms as shown in Figure 5.1. The residues corresponding to the middle C_α atoms in each

pair are in contact with each other, while the other four C_α are not constrained. Contact is determined via hydrogen bond, where the hydrogen atom on the amino group on the middle residue is within some distance cutoff d_{hb} of an oxygen atom on the carboxyl group of the other middle residue.

- **"3+3" via side-chain to side-chain contact:**

  Much like the "3+3" hydrogen-bonded litemotifs, the side-chain to side-chain litemotifs also consist of a pair of three consecutive C_α atoms as shown in Figure 5.2. The difference is that the side chains attached to the two middle C_α atoms are in contact. Contact in this case is defined by at least three atom-to-atom pairings, where all three pairings have distances which fall under some cutoff d_{sc}. A single atom on one side chain can be paired with atoms on the other side chain as long as the pairings meet the distance cutoff.

- **"3+1" side chain - solvent contact**:

  There are really two different kinds of litemotifs that can be constructed from three consecutive C_α and a water molecule, as shown in Figures 5.3 and 5.4. The two share the same, general construction in that the residue of the middle C_α atom is hydrogen-bonded to the water molecule. The difference stems from whether the residue is a hydrogen donor or acceptor.

  When the residue is a hydrogen donor, the hydrogen atom on the amino group is bonded to an oxygen atom on a water molecule. This can be determined via the same d_{hb} distance cutoff criterion between the relevant atoms, as discussed in the "3+3" via hydrogen bonding litemotifs.

  When the residue is a hydrogen acceptor, the oxygen atom on the carboxyl

group is bonded with a hydrogen atom on a water molecule. While contact is defined in the same way as the previous case, the means of determining it is trickier. The protein structure files contained in the PDB have coordinates for the oxygen atoms from water molecules, but not their hydrogen atoms. So, the hydrogen bond has to be inferred via proximity between the oxygen atoms on the carboxyl group and on the water molecule, via the distance cutoff d_{oo}.

As an added constraint, when considering both hydrogen bonding scenarios, it is important for the hydrogen acceptor or donor on the residue, as well as the atom it is covalently bonded to, to align with the oxygen atom on the water molecule so that they all fall on a straight line.

A total of 12,308 single chain proteins with sequence similarities of less than 50% were considered for litemotif extraction. The total numbers obtained for each kinds are listed in Table 5.1. Given that the "3+3" side-chain to side-chain litemotifs are the most populous, subsequent sections will use their collection to motivate discussions, and will be abbreviated as "3+3" SC-SC's. A summary of the important findings for the other kinds of litemotifs will be discussed in Section 5.5.

Table 5.1: Number of litemotifs extracted for each type. A total of 12,308 proteins provided the source. The most numerous are the "3+3" via side-chain to side-chain variety.

Litemotif type	Count
"3+3" side-chain contact	2,528,607
"3+3" hydrogen bonded	425,415
"3+1" O on residue	39,562
"3+1" H on residue	150,731

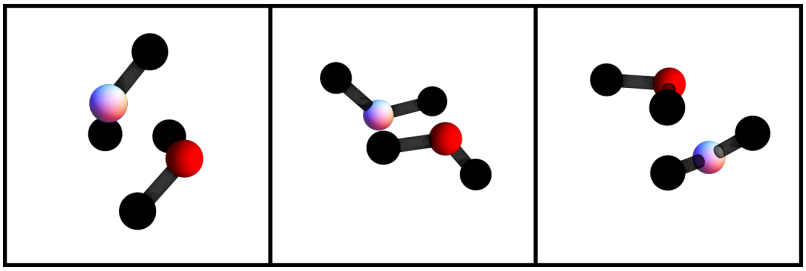


Figure 5.1: Examples of "3+3" via hydrogen bonding. A pair of three consecutive C_α atoms are brought together via hydrogen bonding, with the hydrogen atom on one residue, indicated by the C_α shown in white, attracted to the oxygen atom on another residue, indicated by the C_α shown in red.

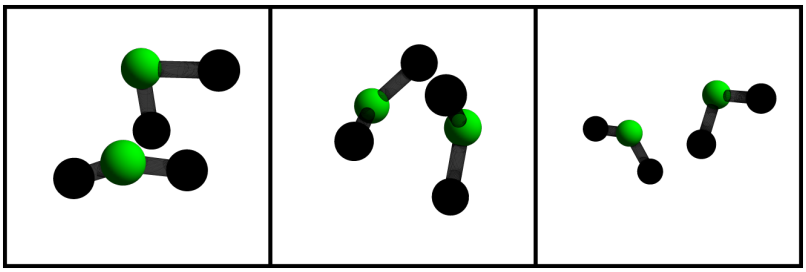


Figure 5.2: Examples of "3+3" via side-chain to side-chain contact. A pair of three consecutive C_α atoms are in contact through their middle C_α's, depicted in green. To ensure the side chains of the middle residues are strongly in contact, at least three atom-to-atom pairings need to fall under the cutoff d_{sc}.

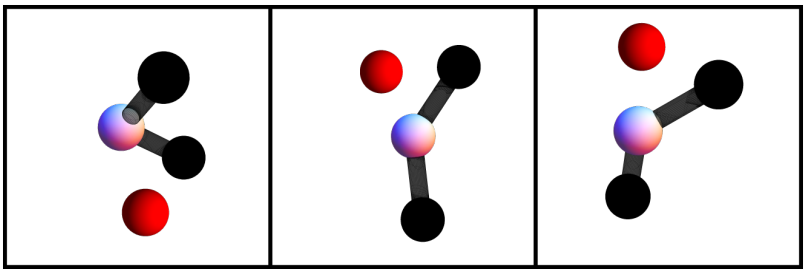


Figure 5.3: Examples of "3+1" litemotifs with three consecutive C_α atoms and an oxygen atom from a water molecule. The residue of the middle C_α atom acts as a hydrogen donor, shown in white, while the oxygen atom acts as a hydrogen acceptor, shown in red.

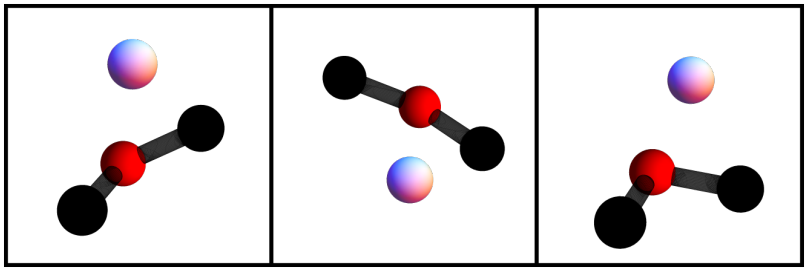


Figure 5.4: Examples of "3+1" litemotifs with three consecutive C_α atoms and an oxygen atom from a water molecule. The residue of the middle C_α atom, shown in red, acts as a hydrogen acceptor, while the water molecule represented by the oxygen atom acts as a hydrogen donor, shown in white.

5.2 Litemotif graphs and dominating sets

Using the litemotif definitions from the previous section, we noted in Table 5.1 that extracting litemotifs from known protein structures in the PDB have yielded counts on the order of tens of thousands to millions. Whether or not these numbers are sufficient and provide enough of a diverse sampling of litemotifs are questions that need to be addressed. Also important is getting a sense of how many non-redundant litemotifs there are in a collection. For instance, litemotifs extracted from α helices could plausibly be similar to each other. In such groupings, it would be preferable to only keep one litemotif.

These issues can only be addressed when we consider how similar litemotifs are to one another. A natural measure of similarity that was briefly discussed last chapter is the RMSD. By computing the pairwise RMSDs for all possible pairs of litemotifs, l_i and l_j, one gets a clearer picture of the ensemble of litemotifs. Figure 5.5 shows a histogram of the RMSD values computed from a random collection of 10,000 "3+3" SC-SC litemotifs.

While the RMSD captures how similar all the litemotifs are to one another, it can quickly become cumbersome to manage as the number of litemotifs in-

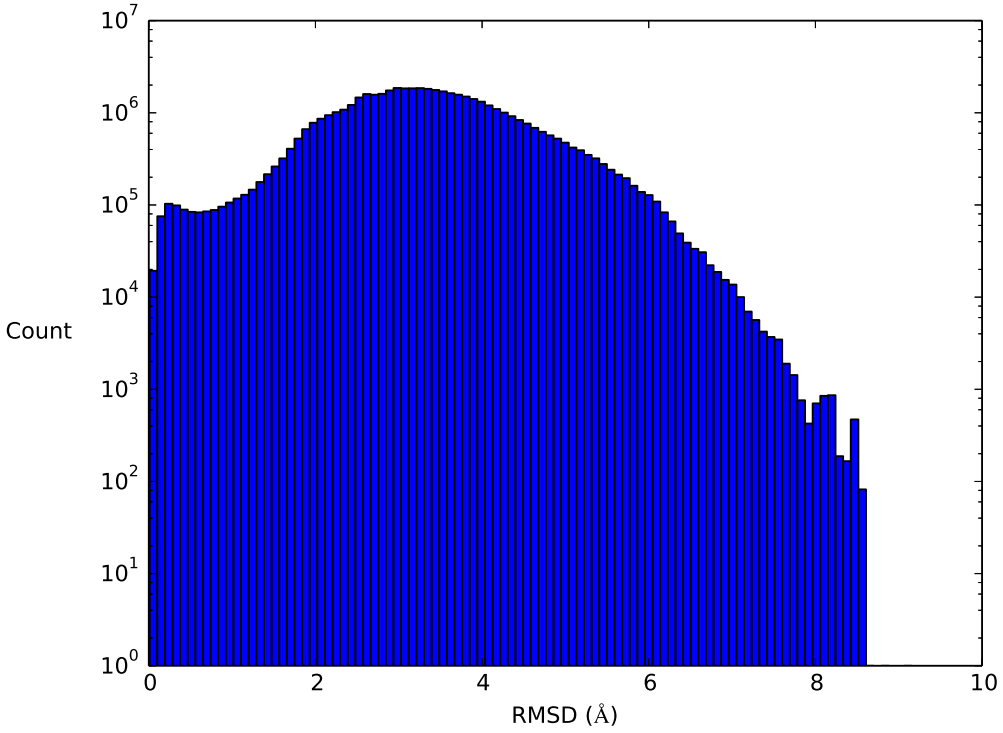


Figure 5.5: Histogram of all the pairwise RMSDs of a random collection of 10,000 "3+3" SC-SC litemotifs. Each pair is optimally aligned, using the procedure outlined in Appendix A, before their RMSDs are computed. The mean RMSD is 3.29Å.

creases. A way to simplify the representation of the collection is to construct litemotifs graphs based on the RMSD calculations. Let each litemotif be characterized by a node, and the "nearness" of two litemotifs by an edge. Unweighted edges are constructed if the RMSD between two litemotifs falls under some threshold c. The RMSD values could conceivably be used to compute weights on each of the edges. For the sake of simplicity, we are interested in using the edges to denote similarity and dissimilarity without regards to the degree of similarity. Care must be taken to determine a threshold c that is not arbitrary, and this is highly dependent on the structure prediction algorithm.

The unweighted litemotif graph, G, provides a simplified picture of how

"close" litemotifs are to each other. It also allows us to use tools from graph theory to answer some of the questions that were posed earlier in the section. One of them, the problem of redundancy, involves finding a smaller collection of litemotifs which are sufficiently dissimilar from each other, yet continue to represent the collection as a whole. This problem can be addressed through finding a *dominating set* of G.

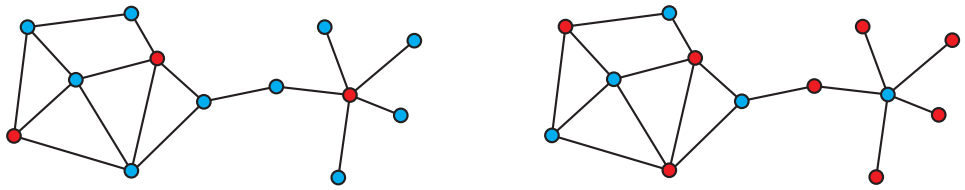


Figure 5.6: A dominating set of a graph G is the set of nodes that are adjacent to all other nodes in G. A graph can have multiple dominating sets, as shown in the two figures above, where red nodes belong to dominating sets.

A dominating set D is a subset of nodes in G which are adjacent to all nodes in G [59]. In our studies, the dominating set represents a smaller collection of litemotifs to which all litemotifs are similar to. As shown in Figure 5.6, multiple dominating sets can exist for a graph. Finding the smallest dominating set would be preferable, but that problem is NP-complete for general graphs. Just finding a dominating set, one whose node count is significantly smaller than that of the entire graph can be done in polynomial time. Figure 5.7 shows how the dominating set of a large graph has significantly fewer nodes than the graph. The algorithm used to find dominating sets is described in detail in Appendix A.

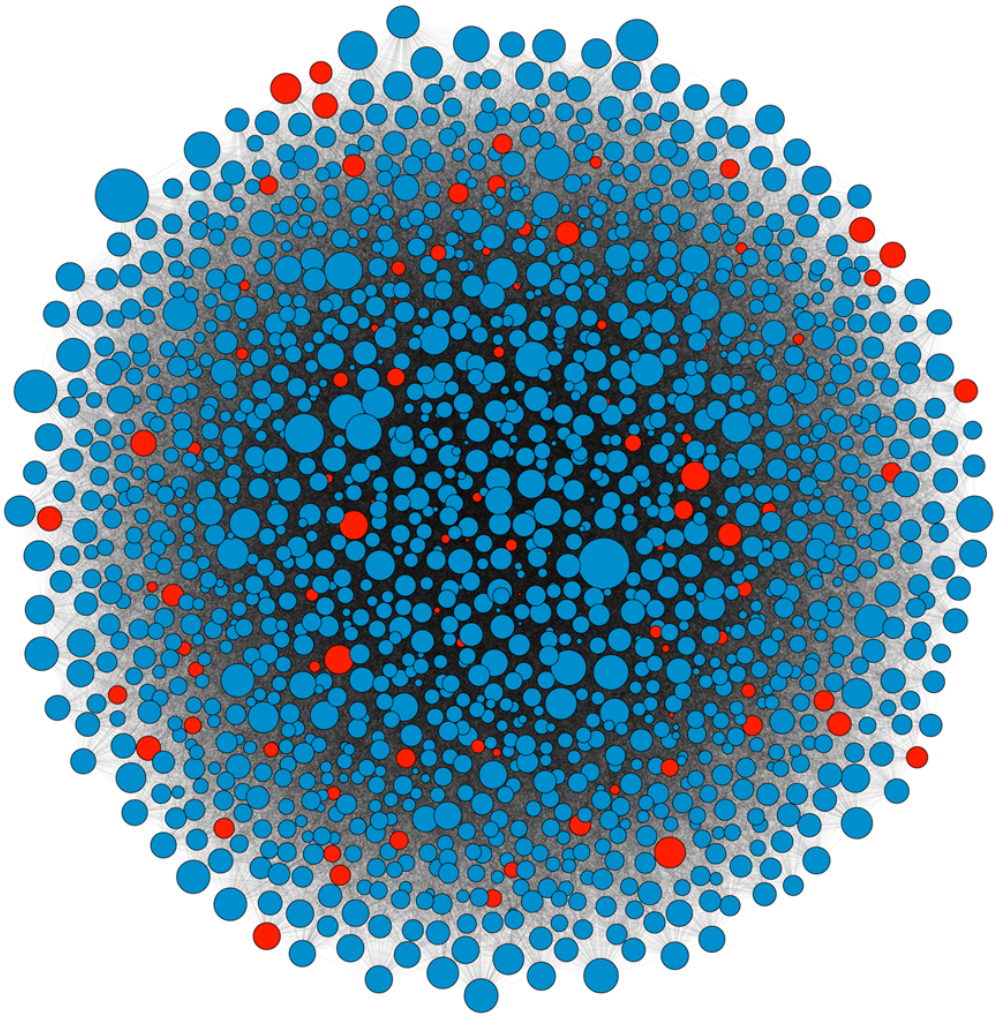


Figure 5.7: A graph of 1500 "3+3" SC-SC litemotifs, with edges, shown in faint grey, placed between litemotifs with RMSDs of less than $c = 1.71$Å. Litemotifs belonging to the dominating set are colored in red, while the rest of the litemotifs are in teal. The size of a node is proportional to its degree. The figure was generated using Gephi [60].

5.3 Saturation of the litemotif collection

The first attempt at understanding the expansiveness of the litemotif collection involved studying how a dominating set's size grows as a function of the size of the graph it represents. It seems plausible that as a litemotif graph grows

larger, the rate at which a corresponding dominating set grows should slow and hopefully converge to a *saturation value*.

Figure 5.8 shows how the dominating sets from the "3+3" SC-SC litemotif graphs grow as the graphs themselve grow. The RMSD cutoffs for edge placement were determined using the RMSD histogram from Figure 5.5, and are RMSD values in the 5th, 10th, 25th, and 50th percentiles. Graphs constructed using generous cutoffs, such as $c = 3.29$Å and 2.64Å, have dominating sets which seem to have converged, but those constructed using stricter cutoffs, $c = 2.06$Å and 1.71Å, have dominating sets that show no signs of convergence.

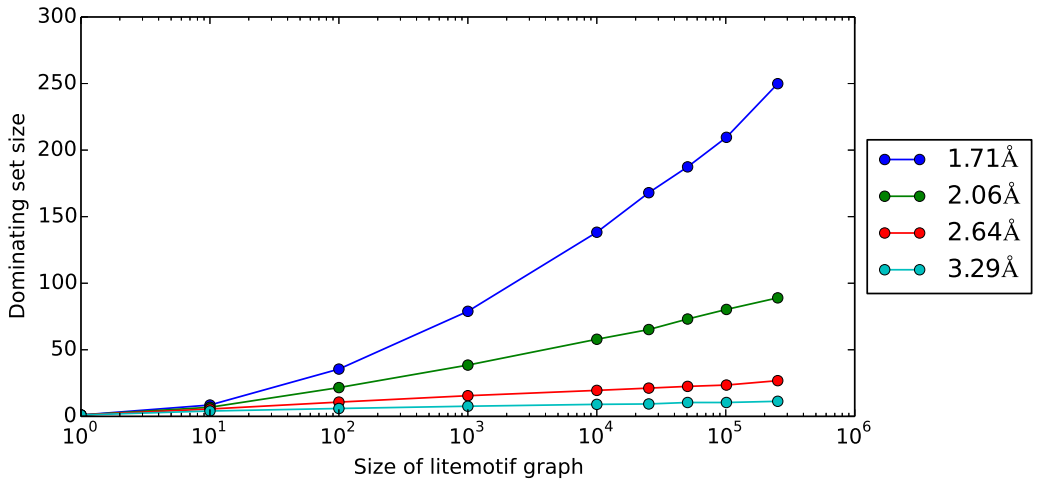


Figure 5.8: "3+3" SC-SC litemotif graphs are generated from random subcollections of fixed sizes. For fixed size and RMSD cutoffs, 10 random graphs were generated, and the average of their dominating set sizes was used to generate the plot. For generous RMSD cutoffs, the dominating set size seems to have converged, but for stricter cutoffs, that seems not to be the case.

For more stringent RMSD cutoff values, there is an issue with determining the saturation value in that convergence is difficult to observe when a "complete" collection contains many rare litemotifs. From a graph perspective, these litemotifs correspond to isolated nodes, which are expected to be numerous. When a dominating set is found from a graph of a random subcollection, the

chance that it contains some of the rare litemotifs can be small. Also, even when considering all the litemotifs that can be extracted from all known proteins, it is unclear whether all possible configurations have been accounted for. It is entirely plausible that yet to be discovered proteins with unusual structures could possess unique litemotifs. This makes it difficult to definitively quantify the saturation value.

5.3.1 Shannon entropy

Instead of hoping to determine the saturation value, quantifying the number of common litemotifs might stand a more reasonable chance at success. To do this, we recall that in the graph G of a random subcollection of size m, each of the m nodes "associates" with a dominating set node d_j. For each node d_j in D, there is a corresponding count $C(d_j)$ denoting the number of nodes in G that associate with it. The counts can be used to generate a probability distribution $P(d_j)$, of a random node n_{random}, selected from G, associating itself with d_j.

An indirect way to quantify the number of common litemotifs is through studying how the *Shannon entropy* of the probability distribution on the dominating set D grows. The Shannon entropy is, much like its thermodynamic counterpart, a measure of the uncertainty (or choice) contained within an information source [61]. In information theory, it characterizes how much uncertainty is contained in an event when the outcomes are inherently probabilistic in nature. In the discrete case, the quantity is defined by the formula

$$H = -\sum_j p_j \log(p_j). \tag{5.1}$$

where p_j is the probability of an event j occurring.

A popular toy case that provides some intuition for the entropy is the coin flip scenario. We characterize the coin flip via the Bernoulli distribution

$$P(X = x) = p^x(1 - p)^{(1-x)} \tag{5.2}$$

where p is the probability of getting a head and X is a random variable denoting whether a head ($X = 1$) or a tail ($X = 0$) is flipped. If we compute the Shannon entropy for the coin flip, we find that

$$H(X) = -p\log(p) - (1 - p)\log(1 - p) \tag{5.3}$$

which is shown in Figure 5.9. We can see that for $p = 0$ and $p = 1$, $H = 0$. Considering there is no uncertainty (nor choice) in coin tosses with such probabilities, such an entropy value makes sense. In the case when $p = \frac{1}{2}$, H attains its maximum, which indicates that uncertainty is greatest when the coins are fair.

For studying the dominating sets of litemotifs, we note from discussions earlier in the section that a probability distribution on the nodes of a litemotif graph belonging to D can be constructed. The dominating set D is finite in size with s nodes, so D can be thought of as a discrete random variable taking on the realized values $\{d_1, \ldots, d_s\}$. The entropy of the event of a node in G associating with a dominating set node can be defined as

$$H(D) = -\sum_j P(d_j)\log(P(d_j)). \tag{5.4}$$

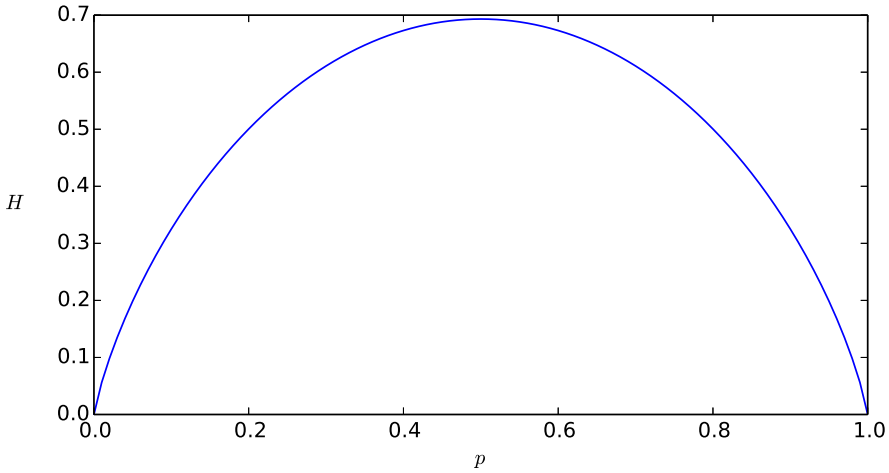


Figure 5.9: The entropy of a coin flip as a function of p. Note that when $p = 0$ and $p = 1$, $H = 0$, while H is maximized when $p = 1/2$. The entropy can be thought of as quantifying uncertainty, so in the event that a coin flip will always turn up heads or tails, it will have zero entropy.

If we study how this value behaves as the size of G increases and see that it converges to some value, we have reason to suspect that the probability distributions for D do not differ. Figure 5.10 shows the entropy calculations for the same graphs studied in Figure 5.8. Regardless of the cutoff, the entropy values level off.

5.3.2 Diversity index

Another way to interpret the entropy is through the diversity index,

$$W = e^H \tag{5.5}$$

which is a value of importance in ecology [62]. In that context, the diversity index is a measure that balances both the "evenness" and the numbers of species

present in an ecosystem and represents an effective species count [63].

The entropy H used in ecology can be computed in the same fashion as the information theoretic version. However, its interpretation is different. Instead of representing probability distributions, each p_j represents the proportion of a particular species within an ecosystem. So, instead of quantifying uncertainty in random events, the entropy can be thought of as a measure of "diversity" within an ecosystem. If we reconsider the coin flip example from the ecologists' perspective, there is zero diversity (i.e. entropy) when $p = 1$ or $p = 0$, since all the organisms within that ecosystem will belong exclusively to one species. In the case when the probability distribution is uniform, $p_j = 1/q$, $W = q$ represents the actual number of different species. For cases of non-uniform probability distributions, W suppresses the proportion of the rarer species.

For our studies, calculating W gives us an approximate sense of the "effective" number of litemotifs in the dominating set. It addresses the main issue we faced earlier in quantifying the saturation value because of the rare litemotifs. Those litemotifs contribute little to the diversity index, as we can see in Figure 5.11. Much like the entropy values we saw in Figure 5.10, the diversity indices quickly converge to a value regardless of the RMSD cutoffs.

5.4 Dimensionality reduction of dominating sets samples

Another way to study how litemotifs are related to each other is by finding representations of them in lower dimensions. While litemotifs naturally reside in $\mathbb{R}^{3n}$, where n is the number of atoms, their geometries most likely constrain them in ways that require fewer degrees of freedom to fully characterize them.

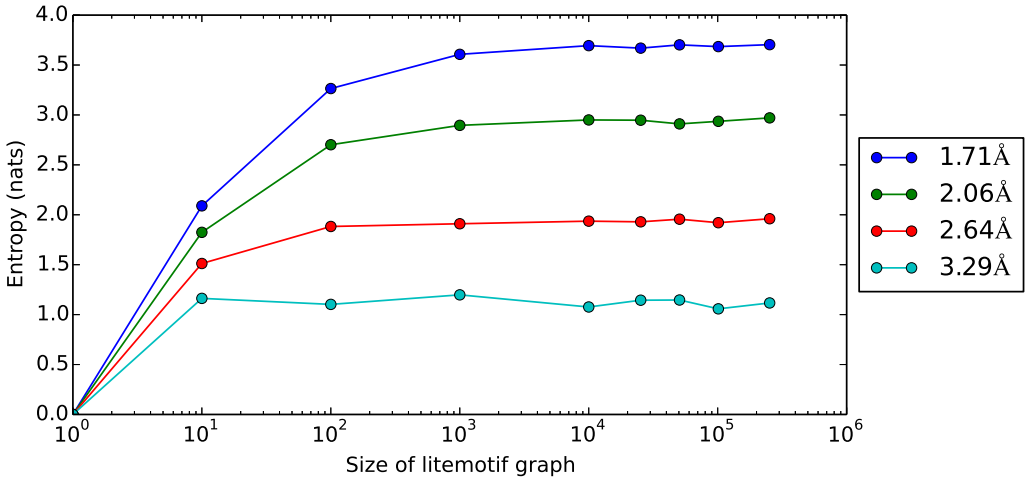


Figure 5.10: Using the "3+3" SC-SC litemotif graphs studied in Figure 5.8, the entropies of the dominating sets were computed. Regardless of the RMSD cut-off values, the entropy values quickly converge and stay more or less constant. Entropy has units of nats when the log function is of base e.

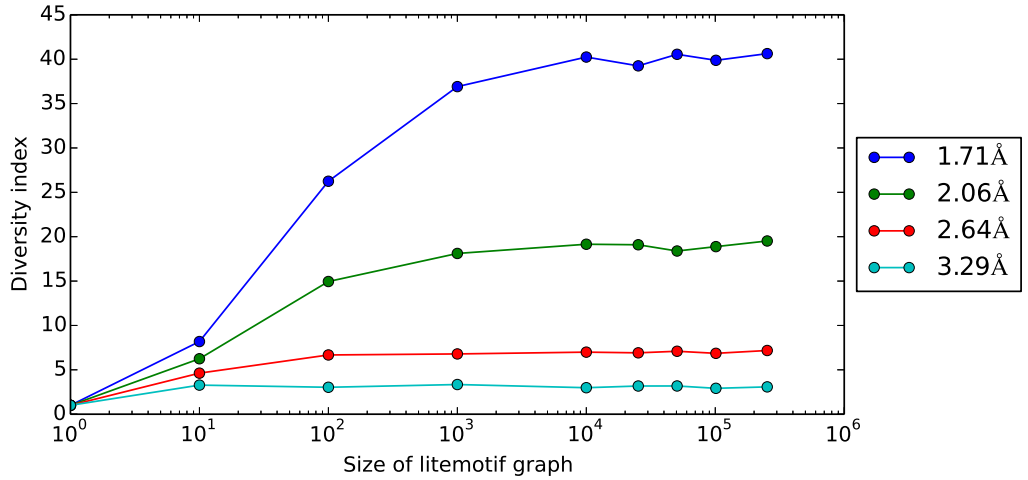


Figure 5.11: The diversity indicies for the dominating sets of the "3+3" SC-SC litemotif graphs studied in Figure 5.8 and Figure 5.10. The entropy values computed for the latter figure were used to compute the diversity indices. These values represent the "effective" number of litemotifs needed to represent this structure type at the specified resolution.

Also, finding $m \leq 3$ dimensional representations allows us to better visualize and appreciate the degree of similarity and closeness between litemotifs which are not possible with the graph representations we discussed in previous sections.

In this section, we first describe a low-dimensional embedding method that is based on the constraint satisfaction methods we outlined in Chapters 2 and 4. The method could be applied to entire collections of litemotifs, but in practice the pairwise RMSD computations would prove to be expensive. For large collections, it is better to first reduce the sample size by using dominating sets.

5.4.1 Lower dimensional embedding via constraint satisfaction

Let $l_1, \ldots, l_k$ be a collection of k litemotifs belonging to the dominating set, and each $l_i \in \mathbb{R}^{3n}$ We wish to find a collection $\mathbf{x}_1, \ldots, \mathbf{x}_k$, where $\mathbf{x}_i \in \mathbb{R}^m$ for $m < 3n$ is a low dimensional representation of the ith litemotif. We can represent the collection as column vectors of the matrix $\mathbf{X}$

$$\mathbf{X} = \begin{bmatrix} \uparrow & & \uparrow \\ \mathbf{x}_1 & \cdots & \mathbf{x}_k \\ \downarrow & & \downarrow \end{bmatrix}, \tag{5.6}$$

and $\mathbf{C} = \mathbf{X}^T\mathbf{X}$ is the $k \times k$ Gram matrix, where each entry $C_{ij} = \mathbf{x}_i^T\mathbf{x}_j$ is the dot product of the low-dimensional representations of l_i and l_j. Instead of searching in $\mathbb{R}^{m \times k}$ where $\mathbf{X}$ resides, we search for $\mathbf{C}$ in $\mathbb{R}^{k \times k}$ that satisfies two constraints.

The first constraint is that the elements of $\mathbf{C}$ satisfy the equations

$$C_{ii} + C_{jj} - 2C_{ij} = \Delta_{ij} \tag{5.7}$$

for i, j and Δ_{ij} is the square RMSD between litemotifs l_i and l_j. The second constraint is that $\mathbf{C}$ should be of rank m. To satisfy these two constraints, we define the projections $P_{dist}[\mathbf{C}]$ and $P_{\text{rank}=m}[\mathbf{C}]$. If $\mathbf{C}_{sol}$ satisfies the relation

$$\mathbf{C}_{sol} = P_{dist}[\mathbf{C}_{sol}] = P_{\text{rank}=m}[\mathbf{C}_{sol}] \tag{5.8}$$

then $\mathbf{C}_{sol}$ is the Gram matrix of k points in $\mathbb{R}^m$ that have the same distances as the original litemotifs in $\mathbb{R}^{3n}$.

To construct the first projection, we note that for any input $\mathbf{C}$, we wish to find $\mathbf{C}' = P_{dist}[\mathbf{C}]$ which is closest to $\mathbf{C}$ and all the entries of $\mathbf{C}'$ satisfy Equation 5.7. This can be done by defining a cost function

$$f(\mathbf{C}, \mathbf{C}') = ||\mathbf{C}' - \mathbf{C}||_F^2 = \sum_{i=1}^{k} \sum_{j=1}^{k} (C'_{ij} - C_{ij})^2 \tag{5.9}$$

where $||\cdot||_F$ is the Frobenius norm and f is minimized subject to the constraints

$$C'_{ii} + C'_{jj} - 2C'_{ij} = \Delta_{ij} \tag{5.10}$$

for all i, j. To minimize Equation 5.9, we use the method of Lagrange multipliers

to minimize the augmented function

$$g(\mathbf{C}, \mathbf{C}', \mathbf{\Lambda}) = f(\mathbf{C}, \mathbf{C}') + \sum_{i=1}^{k} \sum_{j=1}^{k} \lambda_{ij}(C'_{ii} + C'_{jj} - 2C'_{ij} - \Delta_{ij}) \tag{5.11}$$

where $\mathbf{\Lambda}$ is the matrix of λ_{ij}'s. Note that taking the partial derivatives of g yields the following equations,

$$\frac{\partial g}{\partial C'_{ij}} = 2(C'_{ij} - C_{ij}) - 2\lambda_{ij} = 0 \tag{5.12}$$

$$\frac{\partial g}{\partial C'_{ii}} = 2(C'_{ii} - C_{ii}) - \sum_{\substack{j=1 \\ j \neq i}}^{k} (\lambda_{ij} + \lambda_{ji}) = 0 \tag{5.13}$$

$$\frac{\partial g}{\partial \lambda_{ij}} = C'_{ii} + C'_{jj} - 2C'_{ij} - \Delta_{ij} = 0 \tag{5.14}$$

from which we solve for C'_{ij}s. After much algebraic manipulation, it can be shown that

$$C'_{ii} = \frac{\sum_{j=1}^{k} \Delta_{ij} - \alpha + 2 \sum_{\substack{j=1 \\ j \neq i}}^{k} C_{ij}}{k - 2} \tag{5.15}$$

where

$$\alpha = \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} \Delta_{ij} + 2 \sum_{i=1}^{k} \sum_{\substack{j=1 \\ j \neq i}}^{k} C_{ij}}{2(k-1)} \tag{5.16}$$

and

$$C'_{ij} = \frac{1}{2}(C'_{ii} + C'_{jj} - \Delta_{ij}) \tag{5.17}$$

where C'_{ii} and C'_{jj} come from Equation 5.15.

The second projection, $P_{rank=m}[\mathbf{C}]$, computes the rank m projection of $\mathbf{C}$ via

spectral decomposition,

$$P_{\text{rank}=m}[\mathbf{C}] = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \ldots + \lambda_m \mathbf{v}_m \mathbf{v}_m^T \tag{5.18}$$

where λ_i is the ith largest eigenvalue of $\mathbf{C}$ and $\mathbf{v}_i$ is its corresponding eigenvector. To see why this is the case, we wish to find a low-rank approximation that minimizes the cost function

$$h(\mathbf{C}, \mathbf{C}'') = ||\mathbf{C}'' - \mathbf{C}||_F \tag{5.19}$$

where $\mathbf{C}''$ is the rank m approximation. We note that $\mathbf{C}$ is a symmetric matrix, and we expect $\mathbf{C}''$ to also be symmetric. Both $\mathbf{C}$ and $\mathbf{C}''$ can be characterized via the spectral decompositions,

$$\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \tag{5.20}$$

$$\mathbf{C}'' = \mathbf{U}\mathbf{\Gamma}\mathbf{U}^T \tag{5.21}$$

with which we can rewrite Equation 5.19 as

$$h(\mathbf{C}, \mathbf{C}'') = \text{tr}((\mathbf{C}'' - \mathbf{C})^T(\mathbf{C}'' - \mathbf{C})) \tag{5.22}$$

$$= \sum_{k=1}^{n} \lambda_k^2 + \sum_{i=1}^{m} \gamma_i^2 - 2\text{tr}(\mathbf{C}''^T\mathbf{C}) \tag{5.23}$$

$$= \sum_{k=1}^{n} \lambda_k^2 + \sum_{i=1}^{m} \gamma_i^2 - 2\text{tr}(\mathbf{\Gamma}\mathbf{W}\mathbf{\Lambda}\mathbf{W}^T) \tag{5.24}$$

where $\text{tr}(\mathbf{A})$ is the trace operator, γ_i is the ith largest eigenvalue of $\mathbf{C}''$ and $\mathbf{W} = \mathbf{U}^T\mathbf{V}$ is an orthogonal matrix. To minimize h, we need to find $\mathbf{C}''$ such that the

last two terms of Equation 5.24 are minimized. We see that

$$\sum_{i=1}^{m} \gamma_i^2 - 2\mathrm{tr}(\mathbf{\Gamma W \Lambda W}^T) = \sum_{i=1}^{m} \gamma_i^2 - 2\sum_{i=1}^{m}\sum_{j=1}^{n} \gamma_i w_{ij}^2 \lambda_j \tag{5.25}$$

$$= \sum_{i=1}^{m} \left(\gamma_i^2 - 2\gamma_i \sum_{j=1}^{n} w_{ij}^2 \lambda_j \right) \tag{5.26}$$

where w_{ij} is an element of $\mathbf{W}$. Note that $\sum_{j=1}^{n} w_{ij}^2 = 1$ for all i is a constraint, so again, we resort to the method of Lagrange multipliers and consider the augmented function

$$z(\mathbf{\Gamma}, \mathbf{W}, \mathbf{\Lambda}, \mathbf{\Theta}) = \sum_{i=1}^{m} \left[\left(\gamma_i^2 - 2\gamma_i \sum_{j=1}^{n} w_{ij}^2 \lambda_j \right) + \theta_i \left(\sum_{j=1}^{n} w_{ij}^2 - 1 \right) \right]. \tag{5.27}$$

We take the partial derivatives of z to get

$$\frac{\partial z}{\partial \gamma_i} = 2\gamma_i - 2\sum_{j=1}^{n} w_{ij}^2 \lambda_j = 0 \tag{5.28}$$

$$\frac{\partial z}{\partial w_{ij}} = -4\gamma_i w_{ij} \lambda_j + 2\theta_i w_{ij} = 0 \tag{5.29}$$

$$\frac{\partial z}{\partial \theta_i} = \sum_{j=1}^{n} w_{ij}^2 - 1 = 0. \tag{5.30}$$

If $w_{ij} \neq 0$, then Equation 5.29 simplifies to

$$2\gamma_i \lambda_j = \theta_i \tag{5.31}$$

for all j. There is no guarantee that $\lambda_s = \lambda_t$ for all s, t, so the only way for Equation 5.29 to hold while satisfying Equation 5.30 is if $w_{ij} = 0$ for $i \neq j$ and $w_{ii} = \pm 1$. From this, it follows that the ith columns of $\mathbf{U}$ and $\mathbf{V}$ must be parallel

or antiparallel, meaning that $\mathbf{U}$'s column vectors are effectively equal to $\mathbf{V}$'s. We finally note that based on the structure of $\mathbf{W}$, we get from Equation 5.28 that $\gamma_i = \lambda_i$ for $i = 1 \dots m$. So, we see that the sum of the first m spectral components of $\mathbf{C}$ is a suitable, distance-minimizing projection operation.

The projections can be applied in an iterative fashion, like in Chapters 2 and 4, to find a $\mathbf{C}_f$ that approximately satisfies Equation 5.8. Once $\mathbf{C}_f$ is found, the lower dimensional representation can be determined by noting

$$\mathbf{C}_f \approx \begin{bmatrix} \uparrow & & \uparrow \\ \mathbf{v}_1 & \cdots & \mathbf{v}_m \\ \downarrow & & \downarrow \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix} \left(\begin{bmatrix} \uparrow & & \uparrow \\ \mathbf{v}_1 & \cdots & \mathbf{v}_m \\ \downarrow & & \downarrow \end{bmatrix} \right)^T \tag{5.32}$$

from which we can compute the lower dimensional coordinates

$$\mathbf{X} = \begin{bmatrix} \sqrt{\lambda_1} & & \\ & \ddots & \\ & & \sqrt{\lambda_m} \end{bmatrix} \begin{bmatrix} \leftarrow & \mathbf{v}_1 & \rightarrow \\ & \dots & \\ \leftarrow & \mathbf{v}_m & \rightarrow \end{bmatrix}. \tag{5.33}$$

5.4.2 Embedding the dominating set litemotifs

From a random collection of 252,860 "3+3" SC-SC litemotifs, with edge cutoff at $c = 1.71$Å, 244 litemotifs were found to form a dominating set. Their squared RMSDs were used to find $m = 3$ dimensional representations. The method of alternating projections found $\mathbf{C}_f$ with a low error metric within 200 iterations, and was fastest compared to other methods like ADMM.

As shown in Figure 5.12, the eigenvalues of $P_{dist}[\mathbf{C}_f]$ decay rapidly and are

small, relative to the largest eigenvalue, starting from the 6th largest. The ratio

$$\frac{\lambda_1 + \lambda_2 + \lambda_3}{\sum_{i=1}^{k=244} \lambda_i} = 0.618 \tag{5.34}$$

however, suggests that there are still more significant directions with which to describe the 244 litemotifs, particularly in the directions corresponding to the 4th and 5th largest eigenvalues, which are roughly of the same magnitude as the 3rd largest.

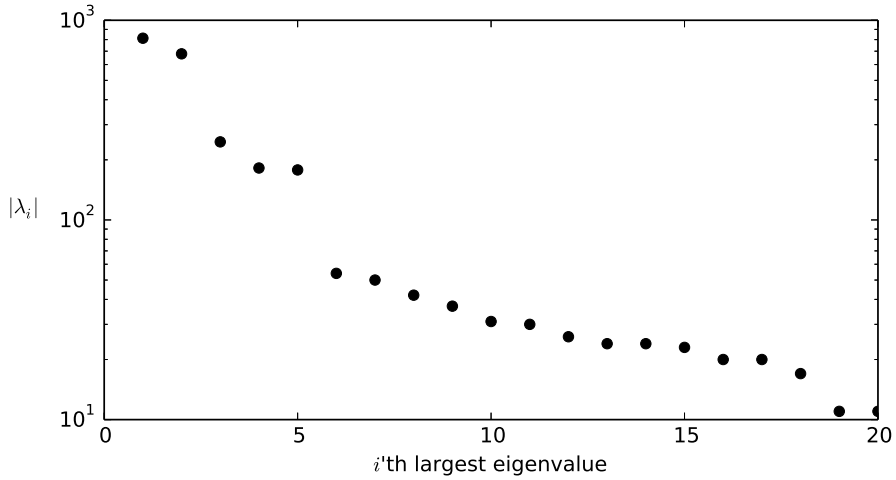


Figure 5.12: The top twenty eigenvalues of $P_{dist}[\mathbf{C}_f]$. The first $m = 3$ eigenvalues contain 61.8% of the power. The 4th and 5th largest eigenvalues are roughly similar in value to the 3rd largest, suggesting that more refined low-dimensional embeddings should incorporate their directions as well.

Each of the 244 litemotifs l_i from the dominating set has a corresponding three-dimensional representation $\mathbf{x}_i$, which can be further simplified by placing the $\mathbf{x_i}$'s in a discrete three-dimensional grid where each voxel is of size $0.85\text{Å} \times 0.85\text{Å} \times 0.85\text{Å}$. The choice of $0.85\text{Å} = c/2$ for the size of the voxels is based on using a scaled value of the edge cutoff c.

Given the voxel representation, it is possible to make a three-dimensional

histogram of the entire litemotif collection. We recall that each litemotif in the original collection associates itself with a single litemotifs l_i from the dominating set. As a result, each l_i has a count for the number of litemotifs in the original collection which associate with it. We use this count to assign a value to the voxel where $\mathbf{x}_i$ maps. The possibility for different $\mathbf{x}_i$'s occupying the same voxel can be accounted for by simply adding up all their associated counts.

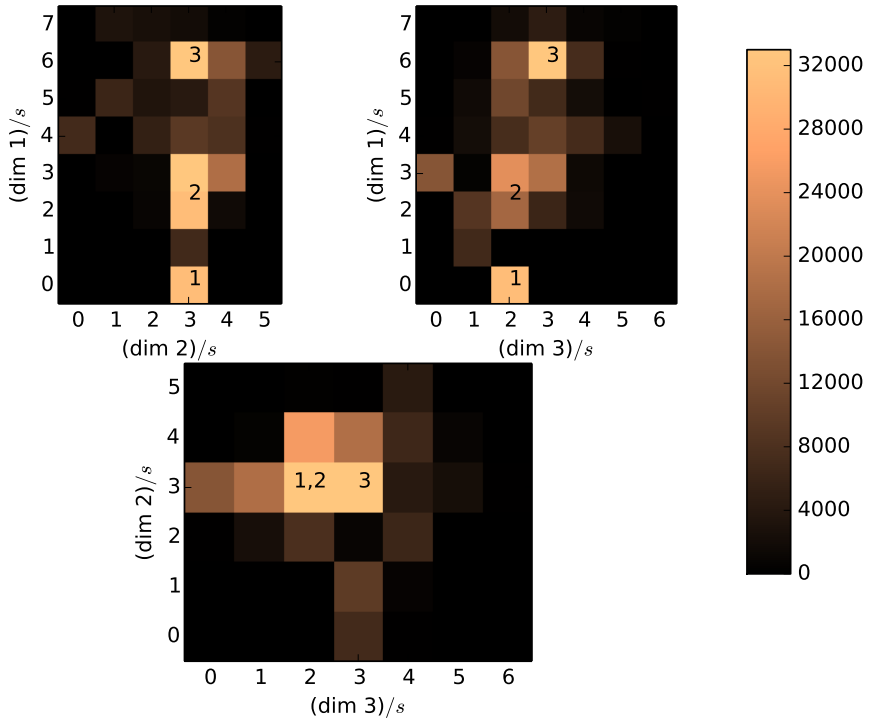


Figure 5.13: Orthogonal projections of the three-dimensional histogram of a "3+3" SC-SC litemotif collection. The numbering system is such that the each integer represents the integer multiple of $s = 0.85$Å. The three bright regions, numbered 1, 2, and 3, show high counts of litemotifs.

Figure 5.13 shows the orthogonal projections of the three-dimensional histogram. The colors represent the number of associated litemotifs contained within the projected voxels. From these projections, it is clear that there are three regions with high concentrations of associated litemotifs.

One distinguishing feature between the different regions is the distance between the two middle C_α atoms of the litemotifs. By computing the mean of these distances over all the litemotifs found in each region, we note that the middle $\mathbf{C}_\alpha$ atoms in voxel 1 seem closer to each other than the ones found in region 2 (which spans two voxels) and voxel 3. These differences are made clear based on the visualization of the litemotifs found in each of the regions, shown in Figures 5.14, 5.15, and 5.16, as well as the values shown in Table 5.2.

Table 5.2: Three regions in the three-dimensional grid contain a high number of associated litemotifs, with over 30,000 in each. There are 16 l_i's in the region, and they represent 38% of a 252,860 "3+3" SC-SC litemotif collection. Also, we note that the mean distances of the middle $\mathbf{C}_\alpha$ atoms are different for each of the regions, with region 3 containing litemotifs which are "spread out".

Region number	Number of associated litemotifs	Number of dom. set elements collapsed to this voxel	Mean distance between middle $\mathbf{C}_\alpha$s
1	31,260	2	3.83Å
2	34,169	8	5.75Å
3	30,788	6	9.40Å

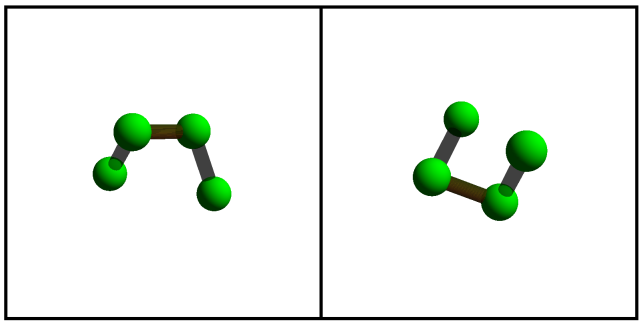


Figure 5.14: Litemotifs in voxel 1. These litemotifs are actually made of four C_α atoms, two of which are found in both three residue subsequences. Based on the definitions in Section 5.1, such overlapping constructions are not prohibited, and are in fact encouraged.

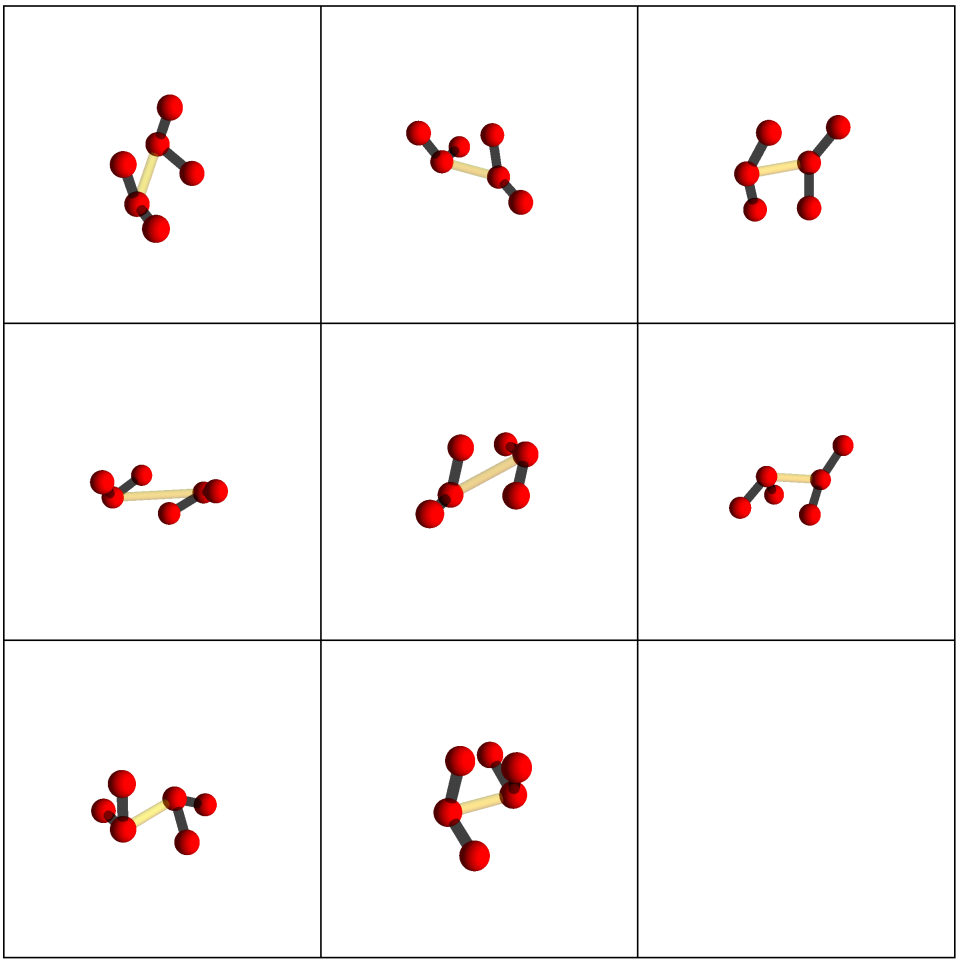


Figure 5.15: Litemotifs in region 2. The yellow tube connects the two middle C_α atoms of the three residue subsequences.

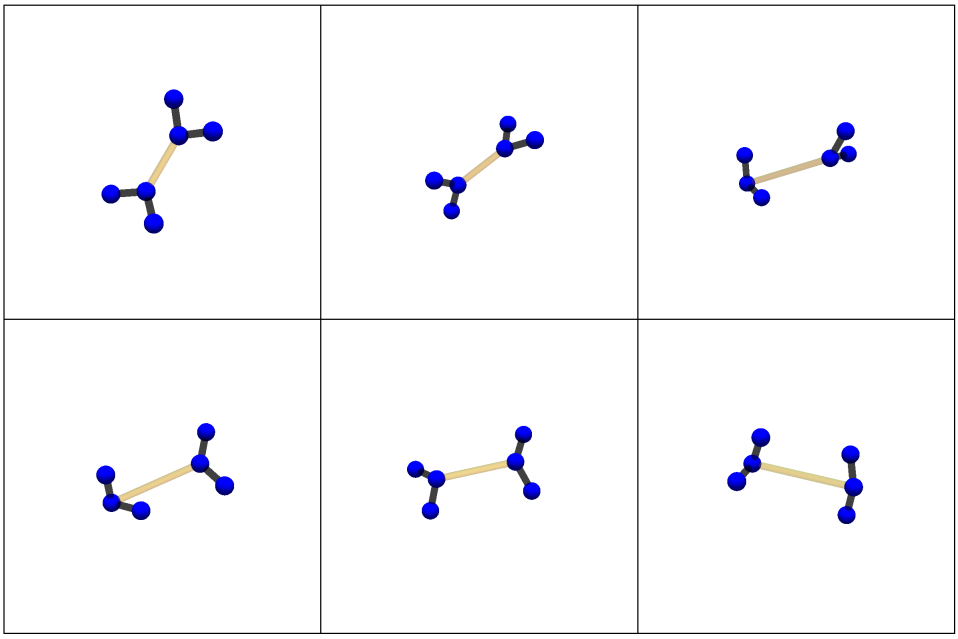


Figure 5.16: Litemotifs in voxel 3. Like in Figure 5.15, the yellow tube connects the two middle C_α atoms. The subsequences are spaced farther in these litemotifs than those from litemotifs contained in the two other regions.

5.5 Results for the other litemotifs

Relative to the side-chain to side-chain contact, the hydrogen bond is a much stricter contact which requires the participation of particular atoms falling within a certain distance of each other. Furthermore, in the case of the the "3+1" litemotifs, there is an added constraint on not just the hydrogen and oxygen atoms, but on the atoms they are covalently bonded to. Based on these considerations, it is reasonable to see that the mean RMSDs computed from random collections of 10,000 litemotifs, as shown in Table 5.3 are much lower than the mean RMSD computed for the "3+3" SC-SC litemotifs.

Using a fixed resolution cutoff of $c = 1.30$Å, graphs and dominating sets for each of the litemotif types were generated for collections of various sizes. At that fixed resolution, the dominating sets and diversities indices for the litemotif

Table 5.3: Mean RMSDs for the litemotifs. A histogram consisting of 10,000 random litemotifs was generated for each of the litemotif types, like in Figure 5.5. The mean RMSD between litemotifs constructed via hydrogen bonds are lower than those constructed via side-chain contacts.

Litemotif type	Mean RMSD
"3+3" side-chain contact	3.29Å
"3+3" hydrogen bonded	2.58Å
"3+1" O on residue	0.42Å
"3+1" H on residue	1.33Å

types grew at different rates. As shown in Figure 5.17, the dominating sets largely seem to saturate for the "3+1" types at around 10,000 nodes, but seem to grow unbounded for the "3+3" types. This is clear when we compare c to the mean RMSDs of each types, since the 1.30Å cutoff is on par or greater than the mean RMSDs for the "3+1" types, while it is less than the mean RMSDs for the "3+3" types. However, the diversity indices, as shown in Figure 5.18, have all saturated, with the "3+3" SC-SC types saturating at a significantly higher value than the others.

Based on Figure 5.17, for collection sizes of around $\sim 10^5$ nodes, the dominating sets are on the order of 1% in size of the "3+3" SC-SC collections, 0.1% in size of the "3+3" via HB collections, and 0.01% in size of the "3+1" collections. These represent significant reductions in the number of litemotifs needed for the prediction algorithms, which would correspond to significant savings in computation time. If we were only interested in the more common litemotifs, we could roughly see a further order of magnitude reduction in the number of litemotifs to include in our collections, based on Figure 5.18.

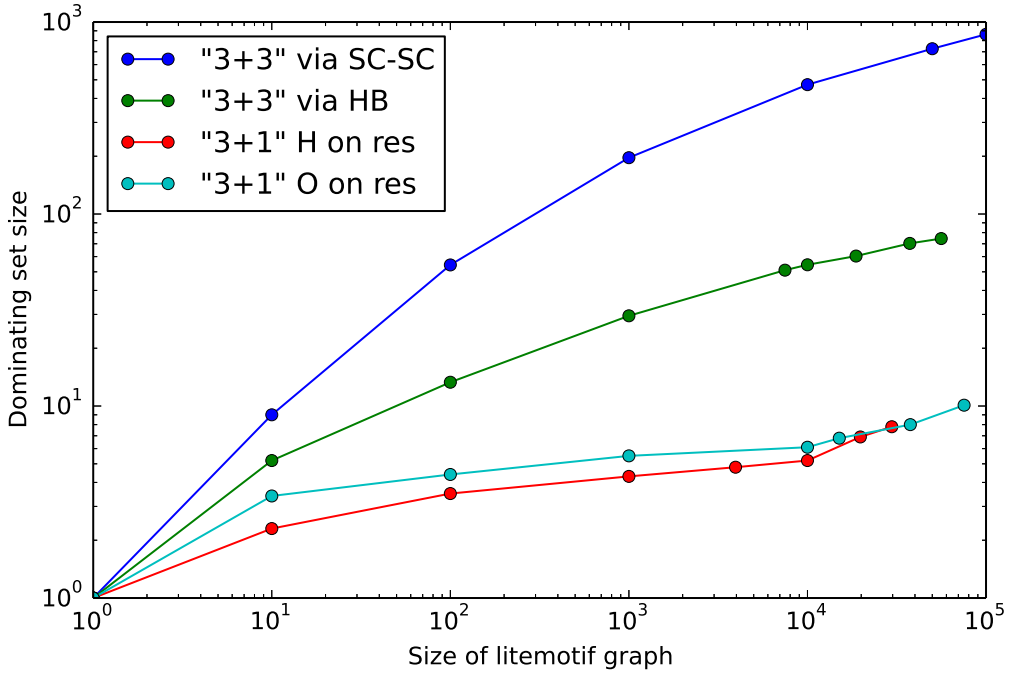


Figure 5.17: Using a resolution cutoff value of $c = 1.30$Å, the dominating sets for the graphs of the different litemotif types grow at different rates. The "3+1" types seem to be close to saturation, with dominating sets staying below 10 litemotifs in the range of graph sizes studied. For the "3+3" types, there is significant, unbounded growth, particularly for the SC-SC variety. Table 5.3 shows that the mean RMSDs for the "3+1" types are either near or below the cutoff c, but above for the "3+3" types.

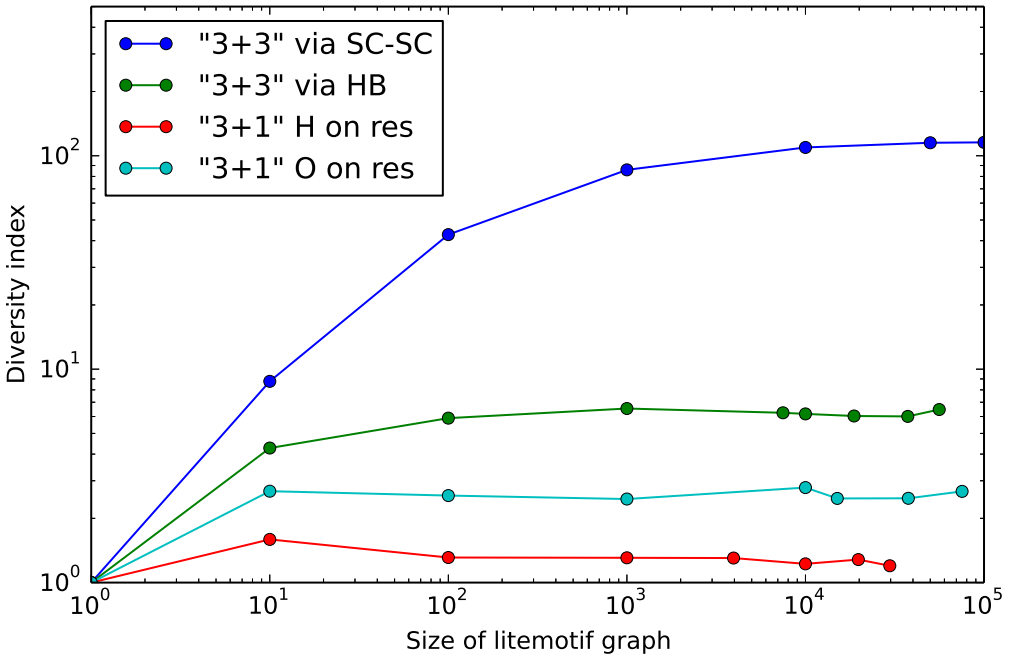


Figure 5.18: The dominating sets, whose sizes were computed in Figure 5.17, were used to compute the diversity indices for the graphs of the different litemotif types. Saturation is observed for all types, but the SC-SC variety saturates at a far higher value than the other litemotifs.

5.6 Conclusion

Using tools from graph theory and dimensionality reduction, we showed how large collections of litemotifs can be studied to assess whether they thoroughly represent a distinct enough set of possible litemotifs. We noted the challenge in assessing exactly how many possible litemotifs can be constructed because rarity of certain litemotifs makes it difficult to count them all. But, we were able to quantify a sense of how many "common" litemotifs exist through the diversity index, and assess how that number grows as collections grow larger.

For the purposes of predicting protein structures, the non-redundant collection of litemotifs obtained through finding a dominating set helps cut down on computational time, sometimes by many orders of magnitude. Also, the collections we studied seem to be well-represented by the common litemotifs. Perhaps future work could focus on exploring and better understanding the set of possible rare litemotifs. The prediction algorithm would greatly benefit from enlarging the collection of possible litemotifs it could consider, so developing the tools to study and identify rare litemotifs seems like a natural next step in building a more complete litemotif collection.

APPENDIX A

PROTEIN ALIGNMENT AND GRAPH TOOLS

A.1 Comparing protein structures

A common measure to gauge similarity of proteins and their substructures is the root mean squared deviation (RMSD). Given two equivalent sets of N atoms, $\mathbf{P}$ and $\mathbf{Q}$, the RMSD is defined as

$$\mathrm{RMSD}(\mathbf{P}, \mathbf{Q}) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \mathrm{dist}(P_i, Q_i)^2} \tag{A.1}$$

where P_i and Q_i are the ith atoms in $\mathbf{P}$ and $\mathbf{Q}$, respectively. In order to compare a pair of structures, it is necessary to align one structure with respect to the other. Assuming that the set of coordinates are rigid, the alignment can done by minimizing the RMSD, or other similarity measures, via rotations and translations.

A.1.1 Alignment calculation

The following method, devised by Arun *et al.* [64], treats the alignment problem of two sets of points as a least-squares minimization problem. Given two sets of coordinates of N points,

$$\begin{aligned} \mathbf{P} &= \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\} \\ \mathbf{Q} &= \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{p}_N\} \end{aligned}$$

the goal is to find a rotation matrix R and a vector $\mathbf{t}$ that minimizes the cost function

$$C(\mathbf{P}, \mathbf{Q}) = \sum_{i=1}^{N} ||\mathbf{p}_i - (R\mathbf{q}_i + \mathbf{t})||^2 \tag{A.2}$$

which is equivalent to the RMSD.

Before any alignment is attempted, it is necessary to "center" each set of coordinates by translating the center of mass to the origin, i.e.,

$$\begin{aligned} \mathbf{x}_i &= \mathbf{p}_i - \mathbf{p} \\ \mathbf{y}_i &= \mathbf{q}_i - \mathbf{q} \end{aligned}$$

where $\mathbf{p}$ and $\mathbf{q}$ are the centers of masses of $\mathbf{P}$ and $\mathbf{Q}$, respectively. Also, we define the matrices

$$X = \begin{bmatrix} \leftarrow & \mathbf{x}_1 & \rightarrow \\ & \vdots & \\ \leftarrow & \mathbf{x}_N & \rightarrow \end{bmatrix} \qquad Y = \begin{bmatrix} \leftarrow & \mathbf{y}_1 & \rightarrow \\ & \vdots & \\ \leftarrow & \mathbf{y}_N & \rightarrow \end{bmatrix}$$

where X and Y are $N \times 3$ matrices with coordinates of the ith atoms in their ith rows. The details of the derivation can be found in [64], but the procedures for finding R and $\mathbf{t}$ are as follows:

1. Compute the 3×3 correlation matrix $H = X^T Y$.
2. Compute the singular value decomposition of H,

$$H = U\Sigma V^T. \tag{A.3}$$

3. The rotation matrix R is given by

$$R = V \begin{bmatrix} 1 & & \\ & 1 & \\ & & \det(VU^T) \end{bmatrix} U^T. \tag{A.4}$$

4. The translation $\mathbf{t}$ is given by the formula

$$\mathbf{t} = \mathbf{q} - R\mathbf{p}. \tag{A.5}$$

A.2 Finding a dominating set

Given a list of litemotifs $L = \{l_1, \ldots, l_n\}$, a dominating set $D = \{d_1, \ldots, d_m\}$ can be found via the algorithm given in Algorithm 1. Note that the algorithm does not calculate all pairwise RMSDs since any litemotif similar to a previously seen one is not considered. This cuts down on the number of operations, often significantly so. The run-time complexity is $\mathcal{O}(mn)$, which can be significantly less than $\mathcal{O}(n^2)$ when $m \ll n$. On various trials, performing pairwise RMSD calculations over all pairs within a reasonable timeframe on a single processor was possible on graphs with a few tens of thousands of nodes. Bypassing the actual graph construction and directly searching for a dominating set was possible on graphs with hundreds of thousands of nodes.

One consequence of the approach outlined in Algorithm 1 is that no two nodes in D are adjacent to each other. Depending on the graph structure, this means that a dominating set could be much larger than the minimum dominat-

ing set, as the example in Figure A.1 illustrates. Exactly how much larger such dominating sets are expected to be compared to the smallest dominating set, on average, is a question worth investigating.

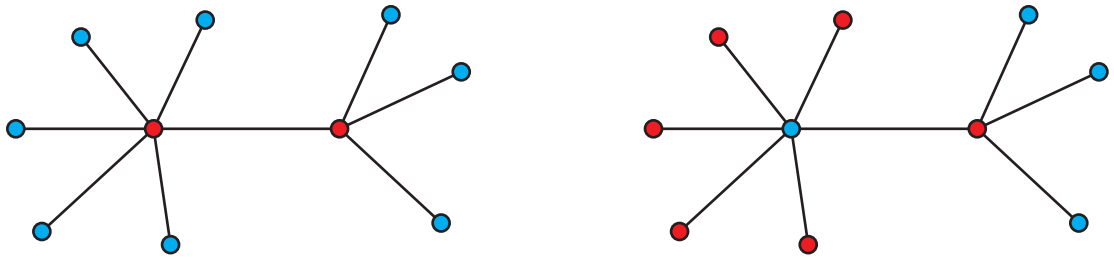


Figure A.1: A limitation of the dominating set algorithm outlined in Algorithm 1 is that no two nodes in a dominating set are adjacent. This will cause the algorithm to skip the smaller dominating set in this graph, shown left, in favor of a larger dominating set, shown right. Note that the red nodes are part of the dominating sets.

Using a slightly modified version of Algorithm 1, it is possible to assign to every node, a node from the dominating set with which it is adjacent to. This is used to generate the probability distributions of the frequency of certain substructures with respect to the dominating sets, which are a crucial part of computing the entropy in Chapter 5.

Data: A list of substructures L, a list M, of length n, initialized to contain 1, and a distance cutoff d_{cutoff}.

Result: The list M is changed where $M[i] == 1$ if substructure $L[i]$ is in the dominating set D, else $M[i] == 0$.

```
for i = 0 to n − 1 do
    if M[i] then
        for j = i + 1 to n do
            RMSD = compute_RMSD(M[i], M[j]);
            if RMSD < d_cutoff then
                M[j] = 0;
            end
        end
    end
end
```

Algorithm 1: Algorithm for finding a dominating set from a list L of substructures.

BIBLIOGRAPHY

[1] D. Sayre, "Prospects for long-wavelength x-ray microscopy and diffraction," Springer Lecture Notes in Phys. **112**, 229–235, (1980).

[2] J. Miao, P. Charalambous, J. Kirz, and D. Sayre, "Extending the methodology of X-ray crystallography to allow imaging of micrometre-sized noncrystalline specimens," Nature **400**, 342 -344 (1999).

[3] D. Shapiro, P. Thibault, T. Beetz, V. Elser, M. Howells, C. Jacobsen, J. Kirz, E. Lima, H. Miao, A. M. Neiman, and D. Sayre, "Biologial imaging by soft x-ray diffraction microscopy," Proc. Natl. Acad. Sci. USA **102**(43), 15343–15346 (2005).

[4] M. M. Seibert, T. Ekeberg, F. R. Maia, M. Svenda, J. Andreasson, O. Jönsson, D. Odić, B. Iwan, A. Rocker, D. Westphal, M. Hantke, D. P. DePonte, A. Barty, J. Schulz, L. Gumprecht, N. Coppola, A. Aquila, M. Liang, T. A. White, A. Martin, C. Caleman, S. Stern, C. Abergel, V. Seltzer, J. M. Claverie, C. Bostedt, J. D. Bozek, S. Boutet, A. A. Miahnahri, M. Messerschmidt, J. Krzywinski, G. Williams, K. O. Hodgson, M. J. Bogan, C. Y. Hampton, R. G. Sierra, D. Starodub, I. Andersson, S. Bajt, M. Barthelmess, J. C. H. Spence, P. Fromme, U. Weierstall, R. Kirian, M. Hunter, R. B. Doak, S. Marchesini, S. P. Hau-Riege, M. Frank, R. L. Shoeman, L. Lomb, S. W. Epp, R. Hartmann, D. Rolles, A. Rudenko, C. Schmidt, L. Foucar, N. Kimmel, P. Holl, B. Rudek, B. Erk, A. Hömke, C. Reich, D. Pietschner, G. Weidenspointner, L. Strüder, G. Hauser, H. Gorke, J. Ullrich, I. Schlichting, S. Herrmann, G. Schaller, F. Schopper, H. Soltau, K. U. K§hnel, R. Andritschke, C. D. Schröter, F. Krasniqi, M. Bott, S. Schorb, D. Rupp, M. Adolph, T. Gorkhover, H. Hirsemann, G. Potdevin, H. Graafsma, B. Nilsson, H. N. Chapman, and J. Hajdu, "Single mimivirus particles intercepted and imaged with an X-ray laser," Nature **470**, 78–81 (2011).

[5] N. D. Loh, C. Y. Hampton, A. V. Martin, D. Starodub, R. G. Sierra, A. Barty, A. Aquila, J. Schulz, L. Lomb, J. Steinbrener, R. L. Shoeman, S. Kassemeyer, C. Bostedt, J. Bozek, S. W. Epp, B. Erk, R. Hartmann, D. Rolles, A. Rudenko, B. Rudek, L. Foucar, N. Kimmel, G. Weidenspointner, G. Hauser, P. Holl, E. Pedersoli, M. Liang, M. S. Hunter, L. Gumprecht, N. Coppola, C. Wunderer, H. Graafsma, F. R. Maia, T. Ekeberg, M. Hantke, H. Fleckenstein, H. Hirsemann, K. Nass, T. A. White, H. J. Tobias, G. R. Farquar, W. H. Benner, S. P. Hau-Riege, C. Reich, A. Hartmann, H. Soltau, S. Marchesini, S. Bajt, M. Barthelmess, P. Bucksbaum, K. O. Hodgson, L. Strüder, J. Ullrich, M. Frank, I. Schlichting, H. N. Chapman and M. J. Bogan, "Fractal morphol-

ogy, imaging and mass spectrometry of single aerosol particles in flight," Nature **486**, 513–517 (2012).

[6] N.-T. D. Loh and V. Elser, "Reconstruction algorithm for single-particle diffraction imaging experiments," Phys. Rev. E **80**, 026705 (2009).

[7] N. D. Loh, M. J. Bogan, V. Elser, A. Barty, S. Boutet, S. Bajt, J. Hajdu, T. Ekeberg, F. R. N. C. Maia, J. Schulz, M. M. Seibert, B. Iwan, N. Timneanu, S. Marchesini, I. Schlichting, R. L. Shoeman, L. Lomb, M. Frank, M. Liang, and H. N. Chapman, "Cryptotomography: reconstructing 3D Fourier intensities from randomly oriented single-shot diffraction patterns," Phys. Rev. Lett. **104**, 225501 (2010).

[8] H. T. Philipp, K. Ayyer, M. W. Tate, V. Elser, S. M. Gruner, "Solving structure with sparse, randomly-oriented x-ray data," Opt. Express **20** (12), 13129–13137 (2012).

[9] K. Ayyer, H. T. Philipp, M. W. Tate, V. Elser, and S. M. Gruner, "Real-Space x-ray tomographic reconstruction of randomly oriented objects with sparse data frames," Opt. Express **22** (3), 2403–2413 (2014).

[10] C. H. Yoon, P. Schwander, C. Abergel, I. Andersson, J. Andreasson, A. Aquila, S. Bajt, M. Barthelmess, A. Barty, M. J. Bogan, C. Bostedt, J. Bozek, H. N. Chapman, J. M. Claverie, N. Coppola, D. P. DePonte, T. Ekeberg, S. W. Epp, B. Erk, H. Fleckenstein, L. Foucar, H. Graafsma, L. Gumprecht, J. Hajdu, C. Y. Hampton, A. Hartmann, E. Hartmann, R. Hartmann, G. Hauser, H. Hirsemann, P. Holl, S. Kassemeyer, N. Kimmel, M. Kiskinova, M. Liang, N. D. Loh, L. Lomb, F. R. Maia, A. V. Martin, K. Nass, E. Pedersoli, C. Reich, D. Rolles, B. Rudek, A. Rudenko, I. Schlichting, J. Schulz, M. Seibert, V. Seltzer, R. L. Shoeman, R. G. Sierra, H. Soltau, D. Starodub, J. Steinbrener, G. Stier, L. Str§der, M. Svenda, J. Ullrich, G. Weidenspointner, T. A. White, C. Wunderer, and A. Ourmazd, "Unsupervised classification of single-particle X-ray diffraction snapshots by spectral clustering," Optics Express **19** (17), 16542–16549 (2011).

[11] T. Ekeberg, M. Svenda, C. Abergel, F. Maia, V. Seltzer, J. M. Claverie, M. Hantke, O. Jonsson, C. Nettelblad, G. van der Schot, M. Liang, D. DePonte, A. Barty, M. Seibert, B. Iwan, I. Andersson, N. Loh, A. Martin, H. Chapman, C. Bostedt, J. Bozek, K. Ferguson, J. Krzywinski, S. Epp, D. Rolles, A. Rudenko, R. Hartmann, N. Kimmel, and J. Hajdu, "Three dimensional reconstruction of the giant mimivirus particle with an X-ray free-electron laser," Phys. Rev. Lett. **114**, 098102 (2015).

[12] M. F. Hantke, D. Hasse, F. R. N. C. Maia, T. Ekeberg, K. John, M. Svenda, N. D. Loh, A. V. Martin, N. Timneanu, D. S. D. Larsson, G. van der Schot, G. H. Carlsson, M. Ingelman, J. Andreasson, D. Westphal, M. Liang, F. Stellato, D. P. DePonte, R. Hartmann, N. Kimmel, R. A. Kirian, M. M. Seibert, K. Muhlig, S. Schorb, K. Ferguson, C. Bostedt, S. Carron, J. D. Bozek, D. Rolles, A. Rudenko, S. Epp, H. N. Chapman, A. Barty, J. Hajdu, and I. Andersson, "High throughput imaging of heterogeneous cell organelles with an X-ray laser," Nat. Photonics **8**, 943–949 (2014).

[13] H. N. Chapman, A. Barty, S. Marchesini, A. Noy, S. P. Hau-Riege, C. Cui, M. R. Howells, R. Rosen, H. He, J. C. H. Spence, U. Weierstall, T. Beetz, C. Jacobsen, and D. Shapiro, "High-resolution ab initio three-dimensional x-ray diffraction microscopy," JOSA A **23**, 1179–1200 (2006).

[14] R. Neutze, R. Wouts, D. van der Spoel, E. Weckert, and J. Hajdu "Potential for biomolecular imaging with femtosecond X-ray pulses," Nature **406**, 752–757 (2000).

[15] M. Born and E. Wolf, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, (Cambridge University Press, 1999).

[16] Z. Huang and K. J. Kim, "Review of x-ray free-electron laser theory," Phys. Rev. Special Topics - Accelerators and Beams **10** (3), 034801 (2007).

[17] D. Hukins, *X-Ray Diffraction by Disordered and Ordered Systems*, (Pergamon Press, 1981).

[18] R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of the phase from image and diffraction plane pictures," Optik **35** (2), 237–246 (1972).

[19] J. R. Fienup, "Phase retrieval algorithms: a comparison," Appl. Opt. **21**, 2758–2769 (1982).

[20] J. R. Fienup and C. C. Wackerman, "Phase-retrieval stagnation problems and solutions," J. Opt. Soc. Am. A **3** (11), 1897–1907 (1986).

[21] V. Elser, "Phase retrieval by iterated projections," J. Opt. Soc. Am. A **20**, 40–55 (2003).

[22] D. R. Luke, "Relaxed averaged alternating reflections for diffraction imaging," Inverse Probl. **21**, 37–50 (2005).

[23] V. Elser and R. Millane, "Reconstruction of an object from its symmetry-averaged diffraction pattern," Acta Cryst. **A64**, 273–279 (2008).

[24] D. C. Champeney, *Fourier Transforms and their Applications*, (Academic Press, 1973).

[25] S. Marchesini, H. He, H. N. Chapman, S. P. Hau-Riege, A. Noy, M. R. Howells, U. Weierstall, and J. C. H. Spence, "X-ray image reconstruction from a diffraction pattern alone," Phys. Rev. B **68**(14), 140101 (2003).

[26] H. N. Chapman, A. Barty, M. J. Bogan, S. Boutet, M. Frank, S. P. Hau-Riege, S. Marchesini, B. W. Woods, S. Bajt, W. H. Benner, R. A. London, E. Plonjes, M. Kuhlmann, R. Treusch, S. Düsterer, T. Tschentscher, J. R. Schneider, E. Spiller, T. Möller, C. Bostedt, M. Hoener, D. A. Shapiro, K. O. Hodgson, D. van der Spoel, F. Burmeister, M. Bergh, C. Caleman, G. Huldt, M. M. Seibert, F. R. Maia, R. W. Lee, A. Szoke, N. Timneanu, and J. Hajdu, "Femtosecond diffractive imaging with a soft-X-ray free-electron laser," Nat. Phys. **2**, 839–843 (2006).

[27] H. J. Park, N. D. Loh, R. G. Sierra, C. Y. Hampton, D. Starodub, A. V. Martin, A. Barty, A. Aquila, J. Schulz, J. Steinbrener, R. L. Shoeman, L. Lomb, S. Kassemeyer, C. Bostedt, J. Bozek, S. W. Epp, B. Erk, R. Hartmann, D. Rolles, A. Rudenko, B. Rudek, L. Foucar, N. Kimmel, G. Weidenspointner, G. Hauser, P. Holl, E. Pedersoli, M. Liang, M. S. Hunter, L. Gumprecht, N. Coppola, C. Wunderer, H. Graafsma, F. R. N. C. Maia, T. Ekeberg, M. Hantke, H. Fleckenstein, H. Hirsemann, K. Nass, H. J. Tobias, G. R. Farquar, W. H. Benner, S. Hau-Riege, C. Reich, A. Hartmann, H. Soltau, S. Marchesini, S. Bajt, M. Barthelmess, L. Strueder, J. Ullrich, P. Bucksbaum, M. Frank, I. Schlichting, H. N. Chapman, M. J. Bogan, and V. Elser, "Toward unsupervised single-shot diffractive imaging of heterogeneous particles using X-ray free-electron lasers," Opt. Express **21** (23), 28729–28742 (2013).

[28] M. J. Bogan, W. H. Benner, S. Boutet, U. Rohner, M. Frank, A. Barty, M. M. Seibert, F. R. Maia, S. Marchesini, S. Bajt, B. W. Woods, V. Riot, S. P. Hau-Riege, M. Svenda, E. Marklund, E. Spiller, J. Hajdu, and H. N. Chapman, "Single particle X-ray diffractive imaging," Nano Lett. **8**(1), 310–316 (2008).

[29] H. N. Chapman, P. Fromme, A. Barty, T. A. White, R. A. Kirian, A. Aquila, M. S. Hunter, J. Schulz1, D. P. DePonte, U. Weierstall, R. B. Doak, F. Maia, A. V. Martin, I. Schlichting, L. Lomb, N. Coppola, R. L. Shoeman, S. W. Epp, R. Hartmann, D. Rolles, A. Rudenko, L. Foucar, N. Kimmel, G. Weidenspointner, P. Holl, M. Liang, M. Barthelmess, C. Caleman, S. Boutet, M. J. Bogan, J. Krzywinski, C. Bostedt, S. Bajt, L. Gumprecht, B. Rudek, B. Erk,

C. Schmidt, A. Hömke, C. Reich, D. Pietschner, L. Strüder, G. Hauser, H. Gorke, J. Ullrich, S. Herrmann, G. Schaller, F. Schopper, H. Soltau, K. Kühnel, M. Messerschmidt, J. D. Bozek, S. P. Hau-Riege, M. Frank, C. Y. Hampton, R. Sierra, D. Starodub, G. J. Williams, J. Hajdu, N. Timneanu, M. Seibert, J. Andreasson, A. Rocker, O. Jönsson, S. Stern, K. Nass, R. Andritschke, C. Schröter, F. Krasniqi, M. Bott, K. E. Schmidt, X. Wang, I. Grotjohann, J. Holton, S. Marchesini, S. Schorb, D. Rupp, M. Adolph, T. Gorkhover, M. Svenda, H. Hirsemann, G. Potdevin, H. Graafsma, B. Nilsson, and J. C. H. Spence, "Femtosecond X-ray protein nanocrystallography," Nature **470**, 73–77 (2011).

[30] S. Marchesini, "A unified evaluation of iterative projection algorithms for phase retrieval," Rev. Sci. Instrum. **78**, 011301 (2007).

[31] M. J. Bogan, D. Starodub, C. Y. Hampton, and R. Sierra, "Single-particle coherent diffractive imaging with a soft X-ray free electron laser: towards soot aerosol morphology," J. Phys. B: At. Mol. Opt. Phys. **43**, 194013 (2010).

[32] M. Wentzel, H. Gorzawski, K.-H. Naumann, H. Saathoff, and S. Weinbruch, "Transmission electron microscopical and aerosol dynamical characterization of soot aerosols," J. Aerosol Sci. **34**, 1347–1370 (2003).

[33] S. Schwyn, E. Garwin, and A. Schmidt-Ott, "Aerosol generation by spark discharge," J. Aerosol Sci. **19**, 639–642 (1988).

[34] Standard Reference Material 2975, Diesel particulate matter (industrial forklift), certificate of analysis; National Institute of Standards & Technology, Gaithersburg, MD 20899, November 7, 2000.

[35] M. J. Bogan, W. H. Benner, S. P. Hau-Riege, H. N. Chapman, and M. Frank, "Aerosol sample preparation methods for X-ray diffraction imaging," J. Aerosol Sci. **38**, 1119–1128 (2007).

[36] L. Strüder, S. Epp, D. Rolles, R. Hartmann, P. Holl, G. Lutz, H. Soltau, R. Eckart, C. Reich, K. Heinzinger, C. Thamm, A. Rudenko, F. Krasniqi, K.-U. Kühnel, C. Bauer, C.-D. Schröter, R. Moshammer, S. Techert, D. Miessner, M. Porro, O. Hälker, N. Meidinger, N. Kimmel, R. Andritschke, F. Schopper, G. Weidenspointner, A. Ziegler, D. Pietschner, S. Herrmann, U. Pietsch, A. Walenta, W. Leitenberger, C. Bostedt, T. Möller, D. Rupp, M. Adolph, H. Graafsma, H. Hirsemann, K. Gärtner, R. Richter, L. Foucar, R. L. Shoeman, I. Schlichting, and J. Ullrich, "Large-format, high-speed, X-ray pnCCDs combined with electron and ion imaging spectrometers in a multi purpose

chamber for experiments at 4th generation light sources," Nucl. Instrum. Meth. A **614**, 483–496 (2009).

[37] N. D. Loh, D. Starodub, L. Lomb, C. Y. Hampton, A. V. Martin, R. G. Sierra, A. Barty, A. Aquila, J. Schulz, J. Steinbrener, R. L. Shoeman, S. Kassemeyer, C. Bostedt, J. Bozek, S. W. Epp, B. Erk, R. Hartmann, D. Rolles, A. Rudenko, B. Rudek, L. Foucar, N. Kimmel, G. Weidenspointner, G. Hauser, P. Holl, E. Pedersoli, M. Liang, M. S. Hunter, L. Gumprecht, N. Coppola, C. Wunderer, H. Graafsma, F. R. Maia, T. Ekeberg, M. Hantke, H. Fleckenstein, H. Hirsemann, K. Nass, T. A. White, H. J. Tobias, G. R. Farquar, W. H. Benner, S. Hau-Riege, C. Reich, A. Hartmann, H. Soltau, S. Marchesini, S. Bajt, M. Barthelmess, L. Strueder, J. Ullrich, P. Bucksbaum, M. Frank, I. Schlichting, H. N. Chapman, and M. J. Bogan, "Sensing the wavefront of X-ray free-electron lasers using aerosol spheres," Optics Express **21**, 12385–12394 (2013).

[38] N. D. Loh, S. Eisebitt, S. Flewett, and V. Elser, "Recovering magnetization distributions from their noisy diffraction data," Phys. Rev. E **82**, 061128 (2010).

[39] A. V. Martin, F. Wang, N. D. Loh, T. Ekeberg, F. R. Maia, M. Hantke, G. van der Schot, C. Y. Hampton, R. G. Sierra, A. Aquila, S. Bajt, M. Barthelmess, C. Bostedt, J. D. Bozek, N. Coppola, S. W. Epp, B. Erk, H. Fleckenstein, L. Foucar, M. Frank, H. Graafsma, L. Gumprecht, A. Hartmann, R. Hartmann, G. Hauser, H. Hirsemann, P. Holl, S. Kassemeyer, N. Kimmel, M. Liang, L. Lomb, S. Marchesini, K. Nass, E. Pedersoli, C. Reich, D. Rolles, B. Rudek, A. Rudenko, J. Schulz, R. L. Shoeman, H. Soltau, D. Starodub, J. Steinbrener, F. Stellato, L. Strüder, J. Ullrich, G. Weidenspointner, T. A. White, C. B. Wunderer, A. Barty, I. Schlichting, M. J. Bogan, and H. N. Chapman, "Noise-robust coherent diffractive imaging with a single diffraction pattern, " Optics Express **20**(15), 16650–16661 (2012).

[40] P. Thibault, V. Elser, C. Jacobsen, D. Shapiro, and D. Sayre, "Reconstruction of a yeast cell from X-ray diffraction data," Acta Crystallogr. A **62**, 248–261 (2006).

[41] F. R. Maia, T. Ekeberg, D. van der Spoel, and J. Hajdu, "Hawk: the image reconstruction package for coherent X-ray diffractive imaging," J. Appl. Crystallogr. **43**, 1535–1539 (2010).

[42] V. Elser, "u^6," http://uuuuuu.lassp.cornell.edu.

[43] H. Rangwala and G. Karypis, *Introduction to Protein Structure Prediction*, (Wiley Series on Bioinformatics, 2010).

[44] A. Fiser and A. Sali, "Modeller: generation and refinement of homology-based protein structure models," Meth. Enzymol. **374**, 461–491 (2003).

[45] A. Roy, A. Kucukural, and Y. Zhang, "I-TASSER: a unified platform for automated protein structure and function prediction," Nature Protocols **5** (4), 725–738 (2010).

[46] J. Jin, X. Chen, Y. Zhou, M. Bartlam, Q. Guo, Y. Liu, Y. Sun, Y. Gao, S. Ye, G. Li, Z. Rao, B. Qiang, and J. Yuan, "Crystal structure of the catalytic domain of a human thioredoxin-like protein," Eur. J. Biochem **269** (8), 2060–2068 (2002).

[47] K. T. Simons, C. Kooperberg, E. Huang, and D. Baker, "Assembly of Protein Tertiary Structures from Fragments with Similar Local Sequences using Simulated Annealing and Bayesian Scoring Functions" J. Mol. Biol. **268**, 209–225 (1997).

[48] J. Lee, S. Y. Kim, K. Joo, I. Kim, and J. Lee, "Prediction of Protein Tertiary Structure Using PROFESY, a Novel Method Based on Fragment Assembly and Conformational Space Annealing," Proteins **56**, 704–714 (2004).

[49] "The RCSB PDB," http://www.rcsb.org.

[50] ChemAxon Ltd., "ChemAxon," http://www.chemaxon.com.

[51] R. L. Gill Jr., J. P. Castaing, J. Hsin, I. S. Tan, X. Wang, K. C. Huang, F. Tian, and K. S. Ramamurthi, "Structural basis for the geometry-driven localization of a small protein," Proc. Natl. Acad. Sci. USA **112** (15), E1908–1915 (2015).

[52] R. Zhou, G. G. Maisuradze, D. Suñol, T. Todorovski, M. J. Macias, Y. Xiao, H. A. Scheraga, C. Czaplewski, and A. Liwo, "Folding kinetics of WW domains with the united residue force field for bridging microscopic motions and experimental measurements," Proc. Natl. Acad. Sci. USA **111** (51), 18243–18248 (2014).

[53] S. Gravel, V. Elser, "Divide and concur: A general approach to constraint satisfaction," Phys. Rev. E **78**, 036706 (2008).

[54] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," Foundations and Trends in Machine Learning **3** (1), 1–122 (2011).

[55] A. Bakan, L. M. Meireles, and I. Bahar I,"ProDy: Protein Dynamics Inferred from Theory and Experiments," Bioinformatics **27** (11), 1575–1577 (2011).

[56] J. A. Garnett, S. Baumberg, P. G. Stockley, and S. E. V. Phillips, "A high-resolution structure of the DNA-binding domain of AhrC, the arginine repressor/activator protein from Bacillus subtilis," Acta Crystallogr. Sect. F Struct. Biol. Cryst. Commun. **63** (11), 914–917 (2007).

[57] K. J. Leath, S. Johnson, P. Roversi, T. R. Hughes, R. A. G. Smith, L. Mackenzie, B. P. Morgan, and S. M. Leaa, "High-resolution structures of bacterially expressed soluble human CD59," Acta Crystallogr. Sect. F Struct. Biol. Cryst. Commun. **63** (8), 648–652 (2007).

[58] H. Ago, K. Hamada, M. Sugahara, C. Kuroishi, S. Kuramitsu, S. Yokoyama, and M. Miyano, "Crystal structure of tt0497 from Thermus thermophilus HB8," To be published (2015).

[59] J. A. Bondy and U. S. R. Murty, *Graph Theory*, (Springer, 2008).

[60] The Gephi Consortium, "Gephi," http://gephi.github.io.

[61] C. E. Shannon, "A Mathematical Theory of Communication," The Bell System Technical Journal, **27** (3), 379–423 (1948).

[62] L. Jost, "Entropy and diversity," Oikos **113**, 363–375 (2006).

[63] R. MacArthur, "Patterns of species diversity," Biol. Rev. **40**, 510–533 (1965).

[64] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," IEEE Trans. on Pattern Analysis and Machine Intelligence **9** (5), 698–700 (1987).