THESIS

SOFTWARE FOR THE USE OF PROTEIN FRAGMENT RECOMBINATION AND REGRESSION IN PROTEIN STRUCTURE DETERMINATION AND DESIGN

Submitted by

Mark Lunt

Department of Chemical and Biological Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2015

Master's Committee:

Advisor: Christopher Snow

Nick Fisk
Asa Ben-hur

ABSTRACT

SOFTWARE FOR THE USE OF PROTEIN FRAGMENT RECOMBINATION AND REGRESSION IN PROTEIN STRUCTURE DETERMINATION AND DESIGN

Recombination of protein structural fragments, in combination with regression-based scoring schemes, provides an alternative to existing iterative strategies for conducting a search over protein conformations. We developed software to define astronomically large combinatorial protein conformation search spaces, and to efficiently search those spaces. We demonstrate that such methods may be applicable to the structure prediction of cytochrome P450 chimeras. More generally, we demonstrate that such methods can be used to produce high-quality protein structural models given only low-resolution X-ray diffraction data.

TABLE OF CONTENTS

I. INTRODUCTION AND BACKGROUND

MOTIVATION

A key difference between small molecules and proteins is that proteins have hundreds of degrees of freedom that determine their conformation, leading to an astronomical number of possible protein conformations. Several grand-challenge problems in computational structural biology are intractable in part because of the inability of current methods to efficiently search through the space of protein conformations. These problems include protein structure prediction, protein design, and the refinement of protein structure models to better fit experimental data.

Current algorithms for searching large conformational spaces tend to be iterative, evaluating one conformation at a time. Even when the evaluation calculation is rapid, such methods are unequal to the required conformational sampling tasks. For example, consider a refinement problem which would benefit from a systematic search of protein conformations that are similar to a non-ideal starting conformation. Even the set of similar protein conformations is far too large to enumerate. For some problems, it is feasible to limit the conformational space to the combinatorial placement of discrete sidechain positions. In this case, finding the optimal combination is still a very challenging computational problem (i.e. NP-complete), but powerful combinatorial optimization algorithms have been developed (Kirkpatrick 1983, Desmet 2002).

This thesis develops the tools necessary to pursue a robust approach to searches over protein conformations. Instead of executing an iterative search through protein conformations, testing tweaks for each step, we reframe the protein conformational search as a combinatorial optimization problem. To do so, we break the protein into discrete blocks, provide alternative poses for each block, and deploy powerful combinatorial optimization and approximation algorithms to efficiently search for the best pose combinations.

The long range goal here is ambitious, developing a novel sampling approach with broad applicability to important problems in computational protein modeling and design. Given the timeframe of a Master's thesis, we had to be quite selective as to which applications we could pursue. Ultimately, we pursued several model applications. **(1)** Template-based protein structure prediction in the absence of experimental data (Section III). This is a very challenging problem, and we limited our efforts here to demonstrating that pose combinations could closely approximate an actual goal structure. **(2)** We next applied pose recombination to the modeling of protein structures that are themselves recombined at the sequence level. Specifically, we recombined fragments from a family of cytochrome P450 crystal structures (Section IV). We were able to recapitulate target chimeras with high fidelity (<1Å) using fragments drawn only from other chimeras. **(3)** Our final application was the preparation of high-quality protein models that fit low-resolution X-ray diffraction (XRD) data (Sections IV and V). In this case, we efficiently identified combinations of poses drawn from homologous proteins that could accurately fit simulated low-resolution XRD data, circumventing several limitations associated with the conventional refinement of models to fit low-resolution XRD data.

BACKGROUND: ITERATIVE PROTEIN CONFORMATIONAL SEARCH METHODS

We define an iterative protein conformational search method to be any method that executes a search through protein conformations and evaluates a single conformation at each step. Iterative refinement methods are surprisingly pervasive. Molecular dynamics simulations and conventional Monte Carlo protein structure prediction fall into this category, as do simulations that support the refinement of models to fit Nuclear Magnetic Resonance spectroscopy (NMR) data or XRD data.

In the case of XRD, an initial model is iteratively altered to minimize the discrepancy between the observed reflections and the reflections predicted by the model. Almost all XRD refinement algorithms depend upon human intervention. Several rounds of refinement are typically applied, as are various algorithms that reposition the model to best explain the data (Emsley 2010). Due to the central importance of XRD structures to the ongoing revolution in structural biology, significant effort has been spent trying to design algorithms that assist with molecular replacement (Stein 2008, Vagin 2010, Hahn 2010, Winn 2011) and iterative refinement techniques (Perrakis 1999, Langer 2008, Emsley 2010, Adams 2010, Murshudov 2011). However, a human being must typically review the output of the algorithm to check for accuracy and computer error.

BACKGROUND: LOW-RESOLUTION X-RAY DATA

Several factors cause iterative refinement methods to struggle with low-resolution (>3Å) data. First, there is a tendency for the refinement process to lead to significant bias as indicated via a large gap (>0.05) between the R_{work} and R_{free}. Second, automated refinement steps often distort the structure. For example, sidechains may be pushed into the mainchain density. Third, manual refinement steps are challenging in that even highly experienced practitioners can find it difficult to interpret low-resolution density.

Although refining low-resolution models can be challenging, the models are still valuable scientific data. Such models prove useful in the absence of higher-resolution alternatives. For example, Borhani *et al.* examined low-resolution data to gain insight into the shape of complexes that bind lipoprotein to lipid molecules; a structure of 4Å resolution was sufficient (Borhani

1997). Similarly, it is possible to glean valuable structural information about polymorphic enzymes with low-resolution data (Bourne 1999).

Insight into structure can also be used to guide subsequent modeling; later efforts at interpreting X-ray data benefit from low-resolution models. Molecular replacement can make use of a lower-resolution model as a starting point, and even low-resolution models can form families with recognizable features. Additionally, structures and complexes are growing larger and low-resolution data is becoming more commonplace (Jiang 2001). For these reasons, several methods which attempt to make use of lower resolution data have been devised.

The simplest use of low-resolution data models is to serve as a useful starting point for later work. Low-resolution structural information can lead to improved models, as demonstrated by Stuart *et al.* when examining a bacteriophage. They used molecular replacement to obtain a low-resolution (7Å) model that was then improved upon with electron microscopy. This combination of techniques works particularly well on viruses (Stuart 2013). Many methods seek to inform low-resolution structures with additional Sources of data that suggest specific structural information absent in the X-ray data (Ward 2013). For example, Koparde *et al.* incorporated hydrostatic interactions to achieve high-quality structures from low-resolution data (Koparde 2011). Additionally, the bond angles of homologous proteins have been used to guide low-resolution XRD interpretation (Schröder 2010).

One of the intrinsic challenges of developing new methods to improve models derived from low-resolution data is that increasing the sampling of alternative structures will also tend to increase the bias by over-fitting noisy data. Such an increase in bias can be avoided by determining the possible structures up front; our method, which recombines structures but otherwise does not alter them, should be resilient to $R_{work} – R_{free}$ bias.

BACKGROUND: PROTEIN RECOMBINATION

Recombination in protein design typically refers to recombination at the level of the primary sequence. DNA sequences that encode protein sequences are recombined, resulting in protein “Chimeras” that incorporate sequence segments from two or more parent sequences. It is a routine technique used by protein engineers who create libraries of diversified amino acid sequences in pursuit of improved variants. Such libraries form sequence search spaces for the design process.

Recombination is one of several ways of adding variety to the sequence library. Other methods include random mutagenesis and targeted mutagenesis (Bloom 2005). Recombination has the advantage that it searches through a space that can be reasonably expected to have good answers “baked in”. If both parents are reasonable designs, Chimeras thereof are expected to be highly enriched in functional variants.

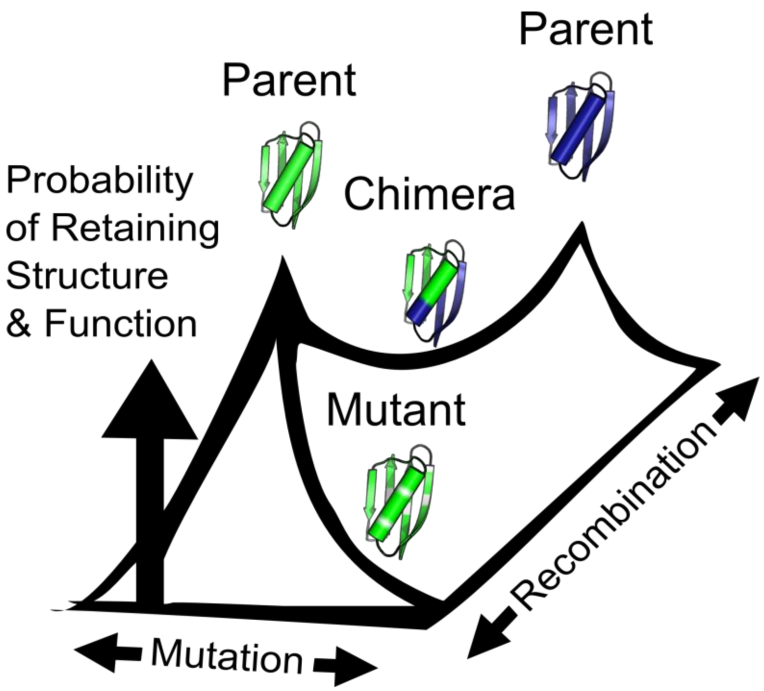

Figure 1: Chimeras represent structures that have a high probability of retaining the properties present in their parents. Unlike a mutant, a Chimera occupies a part of conformation space likely to contain preferable answers.

Fragments of the Parent proteins are rejoined at areas of the sequences called crossover sites. A crossover site represents the part of the Chimera that separates two different sequence sources. These sites are chosen with a variety of techniques that are designed to minimize the disruption of coherent and stable fragments within the protein (Voigt 2002, Zheng 2009); proteins are inherently divisible into substructures larger than individual amino acids but smaller than whole proteins. The technique of recombination has become a mainstay for protein engineers seeking to improve the stability and activity of proteins. It is particularly interesting that the Arnold lab was repeatedly able to rationalize the thermostability of protein chimeras using only crude regression models that account for 1-body contributions from each sequence block (Voigt 2002, Otey 204, Li 2007, Trudeau 2013). These Results suggest that protein fragments can make surprisingly modular contributions to the overall protein structure and stability. We expect that protein structure models produced via fragment recombination are particularly well suited for accurately representing the structures adopted by protein chimeras. This was the impetus for applying structural recombination to the modeling of the cytochrome P450 structures (Section III) obtained by the Arnold laboratory (Snow, unpublished results).

BACKGROUND: REGRESSION & ENERGY FUNCTIONS

Regression is a powerful tool to uncover the relationship between a dependent variable and one or more independent variables. In the current case, the dependent variable is the output value from a calculation (particularly a computationally expensive calculation) applied to a protein structure. Meanwhile, the independent variables correspond to the presence or absence of various mutually exclusive options such as amino acids in a design calculation, sidechain positions in a repacking calculation, or backbone fragments in a discrete backbone search problem. Necessarily, the regression model only approximates the results from the more expensive calculation. The benefit is the dramatic increase in speed, since the predicted score for any discrete combination covered by the regression model can be computed nearly instantaneously. Thus, if the regression model has sufficient accuracy, it can be used to effectively search through enormous search spaces.

Energy functions are used to evaluate structures and test them for plausibility. The Rosetta energy function (Das 2008) is a well-known example, as are the energy functions employed by AMBER (Salomon-Ferrer, 2013). Both of these are scoring functions, although there are important differences, with the former including “knowledge-based” terms derived from protein structure statistics, and the latter intended as a “force field” containing only physics-based, molecular mechanics terms. Generally, energy functions take the form of a sum of interaction terms that approximate various interactions between atoms in a protein. The complexity of energy functions used for protein design is often immense. Electrostatic interactions, bond angles, solvation energy; many mathematical terms may be combined in an effort to improve the accuracy of an energy function. Regression models can be used to approximate these energy functions.

BACKGROUND: CLUSTER EXPANSION

The use of regression to accelerate otherwise intractable protein calculations has been popularized in recent years by Gevorg Grigoryan with the particular nomenclature of Cluster expansion (MacKerell 1998, Stein 2008, Vagin 2010). Cluster-expansion is a regression-dependent technique that was initially made to study alloys (Apgar 2009), but it has been put to use on proteins by Grigoryan *et al*. In particular, cluster expansion techniques have been used to approximate a variety of functions (i.e. the Rosetta energy function score for varying protein designs, or the specificity of coiled coils) using a model that depend upon a sum of protein sequence terms. At heart, cluster expansion relies on regression to fit the expensive calculation (e.g. the stability of a protein evaluated via repacking calculations). The terms may be single (e.g. 1-body terms: residue-10-is-an-arginine), pairwise (e.g. 2-body terms: arginine-10-and-glutamate-18-are-both-present), triple, or higher-order. Regression is used to determine the value of the terms. The key benefit is that the dependent variable, the expensive calculation, can be arbitrarily sophisticated. An important limitation, however, on the sophistication of the regression model is the need for training data sufficient to avoid over-fitting a large number of free parameters.

One drawback of cluster expansion in particular, and regression in general, is the necessity of training the regression model with a large set of initial calculations; typical training sets contain tens of thousands of calculations. To save computational time, it is preferred to have the minimum training set possible (as evaluating each member of the training set can be expensive), but up to an extent a larger training set leads to a better approximation. Eventually increasing the size of the training set will not lead to improved accuracy; at this point it has been saturated, and adding additional terms to the regression is more likely to lead to an improvement (Hahn, 2010).

Perhaps surprisingly, the regression performed by Grigoryan *et al.* found a flexible backbone easier to approximate than a rigid backbone (Grigoryan 2006). One might expect that allowing the backbone to move would increase the computational load of a design strategy. However, their regression was more accurate even with fewer terms if the backbone was allowed to move. In this case it was suggested that allowing the backbone to move minimized confounding clashes between individual residues. This meant that pairwise terms were sufficient for the flexible backbone regression; whereas 3-body terms were typically necessary to adequately represent the clashes that a flexible backbone avoids.

Ng and Snow also found that lower-order terms were sufficient for the prediction of an energy function. Specifically, the AMOEBA (Ehrenreich 1994) polarization energy function, which is not pairwise decomposable, was approximated for combinatorial sidechain optimization. Higher-order terms are subject to a combinatorial explosion, and therefore become computationally prohibitive. However, lower-order (first, second and third order) terms were shown to be sufficient to accurately approximate the multi-body polarization effects. In addition, sets of lower-order terms can be used to predict which higher-order terms are relevant. If the pairwise terms for amino acids at three positions had significant magnitude, it is worth attempting to add a third-order term for those three amino acids. Snow and Ng's work revealed that one can filter out (i.e. ignore) almost 80% of third order terms and thereby reduce the complexity of the regression with this simple check. Techniques like this can also be applied to terms above third order (Ponder 2010).

The method we developed is similar to those outlined above. However, instead of individual amino acids, our method works upon fragments of structure. It is subject to similar constraints as those outlined for Cluster Expansion methods. Specifically, a larger training set means more accuracy (until saturation); fewer terms are required if clashes are allowed to minimize in some way; and finally, we developed techniques to reduce the search space by eliminating terms. This is a powerful way of approaching a difficult problem; the method of Cluster Expansion can be applied to a variety of sequence-dependent design problems (Ng 2011). Our method is designed to be similarly flexible and applicable to a variety of problems.

BACKGROUND: CYTOCHROME P450 CHIMERAS AND PROTEINS HOMOLOGOUS TO THE TEM1 Β-LACTAMASE

We tested our method on two different families of proteins. The first was a set of cytochrome P450 Chimeras that had been constructed by the Arnold lab (Otey 2004, Li 2007). Cytochrome P450s are diverse and catalyze a variety of reactions. They have already been used for drug design, but are promising enzymes as well (Salomon-Ferrer, 2013). We possess structural information computed from X-ray data for 21 different Chimeras generated from 3 Parent proteins. Approximating the structure of these Chimeras was the first challenge we pursued (Section III).

The second family of proteins that we considered was the β-lactamase family. Specifically, we considered a large number of proteins homologous to the TEM1 β-lactamase from *E. coli*. This family is well studied, with hundreds of crystal structures, because these enzymes confer antibiotic resistance to β-lactam containing antibiotics such as ampicillin (Negron 2013).

II. OUTLINE OF THE METHOD AND SPECIFIC CAPABILITIES

OVERVIEW

The method follows a generic algorithm regardless of the particular problem. The general technique is described below.

First, Sources of structure are identified. Sources may represent structural information from multiple entries in the Protein Data Bank, or they may be configurations that are diversified from a single structure. The Sources are divided into Blocks. Different possible configurations for each Block are obtained and called Poses (Fig. 2A). These Poses can be recombined into Pose Combinations, or Combos for short. An initial set of Combos are scored (Instantiated) (Fig. 2B). This set is divided into two subsets; one to test the regression, and one to train the terms of the regression (Fig. 2C). The regression itself approximates Instantiation values for the Training set (Fig. 2D). Then, the resulting regression model can be applied to rapidly predict low-scoring Combos, called Targets. The Targets are also Instantiated and their information is added to the training set, where it can be used to form an update regression. This process repeats until a good approximation can be achieved with the regression and an optimal set of Targets has been found (Fig. 2F). If Instantiation represents a slow

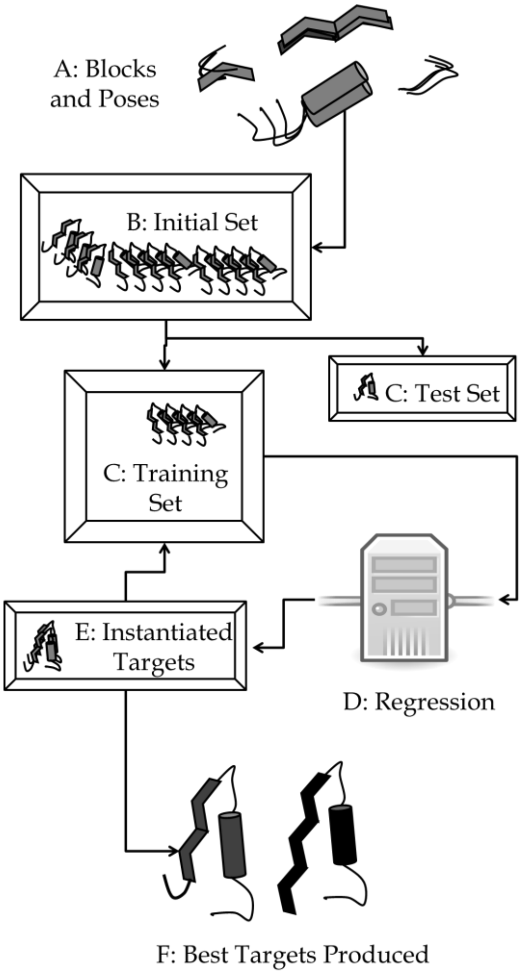

Figure 2: Method Overview. The complexities involved at each step will be expanded upon in Section II.

calculation, even a modestly accurate regression model approximation thereof can accelerate the search for superior combinations (Apgar 2009, Ng 2011).

BLOCK BOUNDARY DIVISIONS

The protein backbone must be divided into segments. The interfaces of these segments are called Block Boundaries. Block Boundaries occur at specific residues and can be chosen in a variety of ways.

The simplest method for choosing Block Boundaries is to manually select them after inspecting the structural model in PyMol or another protein visualization program. In this case, Block Boundary selection relies on biophysical intuition. For example, Blocks Boundaries could separate secondary structure into individual Blocks; a loop should be disconnected from a helix which itself should be removed from a beta sheet. PyMol in particular is useful for manual selection as its 'cartoon' display automatically distinguishes secondary structure.

Automatic block boundary selection methods were tested as well. Two automatic boundary selection methods were used in particular; block division with Normal Modes analysis, and a measurement we designed called Divergence.

Normal mode analysis involves simulating the protein as an elastic network, which may be thought of as a mesh of points connected by various springs. Software made by Joseph N. Stember and Willy Wriggers (Stember 2009) performs this calculation. This software produces several 'snapshots' (coordinate files) of a protein backbone simulated to be flexing along its normal modes. The variance among the coordinates for individual residues was used to select block boundaries at points of inflection. Inflection points make logical sense as Block

Boundaries; to some extent, the protein segments contained within move as rigid subgroups with those points as hinges.

Normal mode analysis produces Block Boundaries at residues that are inflection points for variety among the snapshots provided by the analysis. The variety among groups of homologous proteins can similarly guide Block Boundary selection. First the structural coordinates for the backbones for a group of homologous proteins are aligned. Then, for each of the *N* alpha carbons in the backbone of the Goal model, the sum of distances between that alpha carbon in each of the P Source backbones (α) and the equivalent alpha carbons in every other Source backbone (β) is calculated. This sum represents the magnitude of the spread of the structures for each residue, i, in the backbone. It is a measure of variety that we called the Divergence, *D*, of that residue, where *r* is the distance between the alpha carbons:

$$r_{i,\alpha\beta} = \sqrt{\left(x_{i,\alpha} - x_{i,\beta}\right)^2 + \left(y_{i,\alpha} - y_{i,\beta}\right)^2 + \left(z_{i,\alpha} - z_{i,\beta}\right)^2}$$

$$Divergence(i) = \sum_{\alpha=1}^{P} \sum_{\beta=1}^{P} r_{i,\alpha\beta}$$

Block Boundaries are chosen at residues with a Divergence just below the mean Divergence for all residues, the logic being that a transition from large variation to small variation among Source structures marks a natural Block Boundary point. Areas with large variation will require more Poses, while areas with less variation will have redundancies among the Sources—it is good to separate these regions. Note that the minimum Pose size is three residues; if a suggested Block boundary would trigger Poses smaller than this, it was skipped. Otherwise, the size and number of Poses is not fixed.

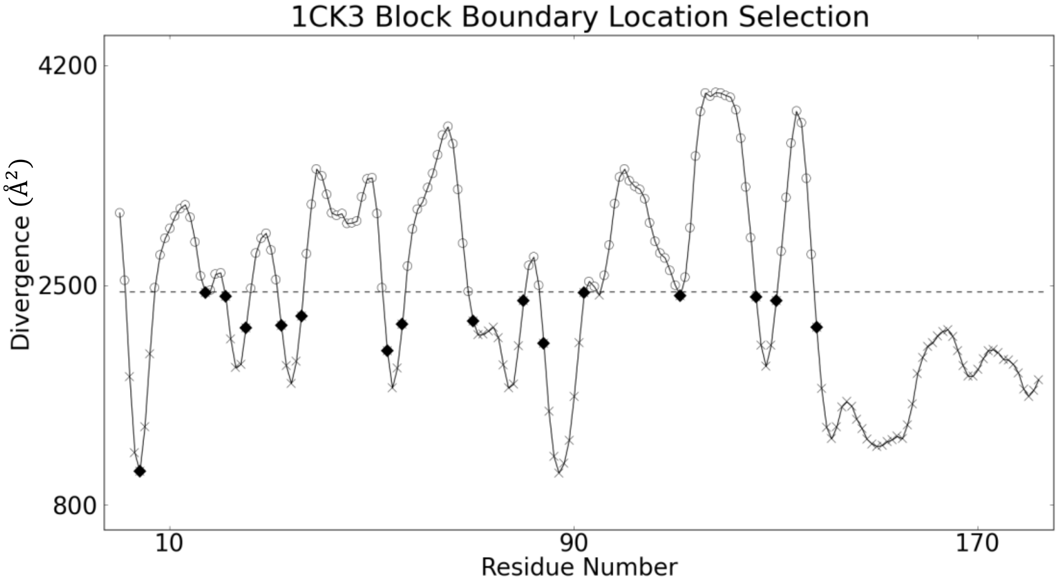

Figure 3: Block Boundary location selection. Block boundary points (♦) are automatically chosen at transition points from high Divergence (○) to low Divergence (x).

CLUSTERING ALGORITHMS

Redundancy is eliminated through the use of a clustering algorithm. In particular, the Affinity Propagation software utility designed by Frey *et al.* was used (Frey 2007). This clustering algorithm chooses exemplars from a list of potential representatives. It requires a list of potential representatives and similarity scores between each pair in the list as inputs. In our case, the potential representatives come from groups of homologous proteins, while the measure of similarity is the sum of standard deviation (SSD) between the alpha carbons of each pair of proteins. The clustering algorithm is used initially to eliminate redundant Sources from consideration when calculating the Divergence and choosing Block Boundaries. Source pairs with a low SSD value between them are redundant. Their only effect is to flatten the graph of the Divergence, so they can be safely removed.

We also used Affinity Propagation to eliminate Poses that are too similar to each other from consideration. In this case we are comparing the structure of specific Poses instead of that of entire Sources. The elimination of redundancy speeds downstream calculations and minimizes the size of the search space. In the specific case of interpreting low-resolution XRD with homologous proteins, 16 or approximately 9% of the Poses for each Block were eliminated with affinity propagation. This reduced the number of combinations in the search space by 80%. The removal of Poses can be tuned with a constant supplied to the Affinity Propagation software.

POSE GENERATION AND THE POSEMANIPULATOR OBJECT

Poses can be constructed in a variety of ways. A simple way of obtaining Pose variety is choosing homologous proteins from the PDB and using them as Sources. In this case, all Source structures are aligned to a Basis and mutated with SHARPEN (www.sharp-n.org) to match the sequence of the Goal structure. Finally, the structures are optimized (combinatorial sidechain repacking) with SHARPEN (Loksha 2009).

A more complicated method of Pose generation involves Normal Modes Analysis (15). This program, as mentioned above, produces snapshots of flexing protein backbones. These snapshots can serve as superposition targets for the placement of individual Poses. In this case, the variety generated is not among the relative coordinates of the Poses; it is in the position of the Poses themselves. Furthermore, the rotations and translations calculated for one flexing backbone can be applied to Poses that originate from another, or to Poses that have been displaced with other methods.

Pose generation is assisted by the **PoseManipulator** object. This Python object accepts a list of Sources as input and can perform various operations on them to generate new Poses for downstream use. It can:

- Align Sources and Poses to specific structures with SHARPEN
- Divide Poses into smaller Poses by adding Block Boundaries
    - using normal modes
    - using Divergence
    - By visual inspection
- Add Poses using snapshots produced by normal mode analysis on a specific structure
    - by aligning Poses to snapshots
    - by applying the motion between snapshots to Poses
- Remove redundant poses with Affinity Propagation
- Create data structures needed for downstream processing

The **PoseManipulator** object helps keep things clear and consistent. It defines the search space. Sometimes a problem will require three thousand Poses; keeping track of these Poses would be a potential source of errors without the assistance of an object. Additionally, the search space can be reduced by removing redundancy among Poses and by intelligently choosing Block Boundaries. The **PoseManipulator** object ensures that the space contains useful, unrepeated structural information and that the numbering for Poses and the Block Boundaries is consistent.

INSTANTIATION OF TARGETS AND THE BACKBONESEARCH OBJECT

The **BackboneSearch** object keeps track of Pose Combinations (Combos) and their instantiated value. Instantiation is simply the scoring of any Combo; it may be the SSD of that Combo when compared to a standard, or the Rosetta Energy score of the structure, or any other measure related to proteins. Instantiation itself is the process that regression learns to approximate.

A certain number of random Combos are instantiated at the outset. These Combos are separated into two groups; the Training Batch and Test Batch. The Training Batch is used in a later step to kick off the regression, while the Test Batch is held in reserve to test the progress of regression. A **BackboneSearch** Object is capable of generating random Combos for these Batches.

In addition to generating random Combos, the **BackboneSearch** can:

- Store the values resulting from Instantiation calculations
- Store Instantiated values for partial Combos (if the Instantiation method permits it)
- Retrieve pairwise, triple, or higher-order terms for a specific combination
    - Suggest new terms to add to the regression by their presence in a combination
- Systematically instantiate Combos corresponding to segments of a larger Combo
- Keep track of and save Taboo (disallowed) combinations
- Represent the data above concisely using a hypergraph data structure

Two of these features merit additional explanation. First is the ability to use Taboo terms; Taboo terms are pairs (or triples, or higher-order terms) that are not 'allowed' to be used in any

Combo. It was with Rosetta in mind that we added the functionality. If instantiation involves rebuilding the Combo and calculating its Rosetta energy, then a Taboo combination might correspond to a set of Poses with an irreconcilable clash. Taboo terms need not be pairwise, even in the case of clashing structure; sometimes the choice of three or more specific Poses will lead to a clash. The inclusion of Taboo terms speeds later steps by removing implausible Combos from the search space.

To speed up calculations, we use a hypergraph datastructure. Hypergraphs are a generalization of the generic graph structure. Whereas edges in a graph always connect two nodes, the hyperedges in a hypergraph connect an arbitrary number of nodes. Thus, the hypergraph data structure is perfectly suited to store calculations or approximation terms that are defined not just over 2-body, but also 3-body terms, 4-body terms, etc. Anticipating the utility of such models, the SHARPEN software platform contains an EnergyHyperGraph datastructure. To our knowledge, this is a unique feature of the SHARPEN package compared to other protein modeling software platforms.

The **BackboneSearch** object can be fundamentally transformed with the use of an energy hypergraph. The object is typically set to automatically Instantiate a combination if the Combo has not previously been Instantiated. Instead of Instantiation, however, the value of a Combo can be calculated by adding up the values of the edges of a hypergraph that correspond to the terms in that Combo. This process takes microseconds, leading to significant time savings. A properly constructed hypergraph is an end result of a trained regression model, and allows for a rapid approximation of Instantiation for any Combo. Such an approximation is particularly useful when determining new Targets.

REGRESSION AND THE REGRESSIONPROBLEM OBJECT

After the Training Batch is constructed, regression can start in earnest. First, terms are chosen. The regression software will try to ascribe values to these terms so that any Combo can be approximated just by adding up the appropriate terms. Single terms are nearly always considered. These terms correspond to individual Poses directly. Higher order terms are often used as well. In the jargon of cluster expansion, these terms are called pairwise, contiguous double, etc. They are discussed below.

The CVXopt software package (S. Anderson, 2013) is used to perform a regression with the Test Batch supplying the needed Combos and Instantiated values. Once that is complete, any Combo can be quickly approximated by summing the terms that are logical subsets of the combination.

The search problem can benefit from repeatedly running regression, to improve the approximation as new combinations are tested. To organize the learning process, we rely on **RegressionProblem** objects with the following capabilities:

- Supplying terms to add to the model
    - all single terms
    - all pairwise terms
    - all contiguous double terms (explained below)
    - triple terms from error (explained below)
    - triple terms from presence in low-or-high-scoring Combos
- Automatic Target Instantiation
- Dissection of Combos
- Updating the regression model

- Producing plots and other feedback for measuring progress

The ability of regression to produce accurate approximations is predicated on the ability of terms to stand in for more complicated processes. Term selection is important. Suppose a backbone is divided into 16 Blocks with 8 Poses each. There will be at least $16 * 8 = 128$ 1-body terms—an eminently manageable number. Single terms are ubiquitous, but higher order terms are optional. A common strategy is to simply include all pairwise terms: in the case above that would be $105 * 8^2 = 6720$ additional terms.

An excessive number of terms will result in overfitting. Consequently, it is desirable to eliminate terms whenever possible. A simple way of eliminating pairwise terms is to only consider 'contiguous terms'—terms that correspond to Poses in adjacent Blocks. Poses that are adjacent are more likely to interact with each other and are thus prime targets for additional terms. This elimination reduces the number of added terms to $15 * 8^2 = 960$.

Adding the entirety of triple or higher order terms requires that the problem be significantly more tractable than sixteen blocks with eight poses each. However, it is possible to selectively add terms instead. Triple terms may be added in response to unexpected deviations from predictions. Terms found in Combos that are poorly predicted are prime candidates for addition to the regression model. Terms may also be added to improve the approximation in certain score regimes. For example, it is desirable to add terms corresponding to triples that are present in low-scoring Combos because the low-scoring area of the search space needs to be approximated accurately for the search to be successful.

Controlling the areas in which the regression is accurate is not limited to term selection. Regression weighting is another strategy we employed. Weighting allows low-scoring Combos to disproportionately affect the magnitude of the terms in the regression.

$$W_{combo} = \mathrm{e}^{\left(\frac{\mathrm{SC}_{combo,shifted}}{\mathrm{SC}_{min,shifted}} * F\right)}$$

The weight W of a Pose combination in the regression is proportional to the exponent raised to a factor F multiplied by the ratio of the Pose combination's shifted score, $SC_{combo,shifted}$, to that of the minimum scoring Combo, $SC_{min,shifted}$. The shifted score is just the score of a Combo minus the maximum Score that has been found. This means that all SC are negative, with $SC_{min,shifted}$ being the largest in magnitude. F, the weight factor, is typically 10. This means that the weight of the lowest score has weight $W_{min} = \mathrm{e}^{(10)} = 22026$ while the weight of the highest score is $W_{max} = \mathrm{e}^{(0)} = 1$. With a weight factor of 10, the lowest scoring combination is weighted at about twenty thousand times the importance of the highest scoring combination. The effect is to consider the squared error in prediction for the lowest scoring combination twenty thousand times more important than the squared error in the highest scoring combination. This reflects the fact it is most important to correctly approximate the lowest scores. Even so, the regression tends to produce reasonably accurate predictions for higher scoring Combos.

Weighting like this is useful because (in general) lower scores tend to correspond to better solutions to problems. The traditional R_{free} and R_{work} measures are best when low, as is the Rosetta Energy function and RMSD from a specific Goal structure. Instantiation methods are constructed such that lower scoring combinations are preferred.

In addition to weighted regression, the **RegressionProblem** object can also perform ridge regression. This technique penalizes terms that deviate from zero. The ridge regression strategy can be useful to suppress overfitting. It is important, however, to setup the problem so that the value of the terms should indeed be small numbers. In the cases we have tested, we have used a free fitting constant so as to permit the remaining terms to adopt small values without degrading the overall fit.

TARGET SELECTION AND ITERATION

A group of Targets are chosen; these are Combos that are predicted to score favorably after Instantiation. If only single terms were used, then the Targets with the best predictions are directly computable by adding each of the Poses with the lowest score for each Block together. More generally, a low-scoring Target is found by applying the CHOMP FasterPacker function, based upon the Faster algorithm designed by Desmet *et al.* (Desmet 2002). This method was originally designed for selecting protein sidechain conformations. It utilizes 'batch relaxation'—a technique of altering all sidechain positions (or in this case, all Poses) at once, and changing them to minimize an energy score. Subsequently, it iteratively substitutes alternate sidechain conformations at each position (in this case, Poses at each Block) and each pair of positions, while accepting favorable changes with some probability.

To generate diverse Targets, MonteCarlo substitution is applied to the Target found by FasterPacker. A MonteCarlo substitution with increasing temperature can be used to generate any number of Targets, but typically 1000 were selected. By using MonteCarlo we can collect diversity in the Combinations. A wide variety of Combos will be chosen as Targets when most combinations in the training set have similar scores; conversely, when certain Combos are scoring very well, Targets are probabilistically likely to be similar to those Combos.

The Targets are then Instantiated and added to the Training Batch, which is then used to choose new Targets. The sequence of Regression / Target-selection / Instantiation is repeated until a convergence criterion is satisfied; typically a cessation of improvement in the average score of the Targets is the end point. After several rounds of regression, a reasonable approximation can be achieved, as can an ensemble of favorable Targets. If the goal is to improve the approximation for the search space at all levels, the error in predictions can be examined instead of Target score. Similarly, Target selection itself can be random or based upon previous errors in predictions. Targets that are a few substitutions away from previously poorly-predicted Targets will likely provide an improvement to the regression, as they contain interactions that the regression previously misjudged.

COMBO DISSECTION

Combos that are found to have unusually high/low scores or unusually high error may be "dissected" to gain insight. Dissection involves Instantiating sub-combinations formed from the Pose selections within the Combo. Pose pairs then Pose triplets are systematically instantiated; corresponding terms are added to the regression to capture the higher order effects of these groups of Poses. For example, it may be found that three specific Poses incur a penalty to the score when they are chosen together. For example, if Instantiation involves repacking the side chains, one can easily imagine three poses necessitating unusual side chain positions. The addition of a term that corresponds to the three Poses may improve the accuracy of the approximation found by Regression. The added term will likely be contain a penalty—in that case, the regression is effectively deciding the penalty that those three poses will receive if they are chosen in conjunction with each other. Of course, this method of Dissection requires that Instantiation be a process that can be applied to incomplete combinations; some methods of Instantiation require full Combos and Dissection cannot be applied to them.

INSTANTIATION METHODS

It was mentioned earlier that Instantiation is simply the scoring of a Combo. We have focused on three different types of Instantiation, outlined below, but these are by no means the only forms that Instantiation might take. The only requirement for an Instantiation method is that it be a function that accepts a Combo and produces a score.

For example, consider a search space representing possible protein conformations. It would be useful to know if this space contains a close match to candidate target structures before expending the energy function computations necessary to search the space. If a Goal has been chosen for recapitulation then the sum of squared deviations (SSD) to ensure the construction of a quality search space. The squared coordinate differences between every alpha carbon are summed for a pair of proteins.

$$SSD = \sum_{n=1}^{N} \left(x_{n,\alpha} - x_{n,\beta}\right)^2 + \left(y_{n,\alpha} - y_{n,\beta}\right)^2 + \left(z_{n,\alpha} - z_{n,\beta}\right)^2$$

n: specific residue

N: total number of residues

α, β: specific proteins

x,y,z: alpha carbon coordinates

Root mean square deviation is very similar. It is often used as a measure of differences between two proteins structures. For our purposes, RMSD is calculated for alpha carbon coordinates unless otherwise noted.

$$RMSD = \sqrt{\frac{1}{N}\left[\sum_{n=1}^{N}\left(x_{n,\alpha} - x_{n,\beta}\right)^2 + \left(y_{n,\alpha} - y_{n,\beta}\right)^2 + \left(z_{n,\alpha} - z_{n,\beta}\right)^2\right]}$$

The difference between RMSD and SSD is of scale. Whereas RMSD is a widespread and concise way of measuring and summarizing the differences between two structures, SSD "penalizes" deviations without masking those deviations through the mean or the root. We prefer SSD as a target for Instantiation since it is "extensive" rather than "intensive", and better mimics the scaling of an energy function.

A reasonable use of our method is to approximate one protein in terms of another; however, for testing it is useful to be aware of the best possible approximation. We can test the structural differences between a Combo and a Goal that is regarded to be the 'Gold Standard'--a best answer by some definition. This method of instantiation makes it easy to see how close it is possible to get to a Goal structure from the provided Poses in a test case (Section III).

A practical challenge for our method is the approximation of the Rosetta energy function. This function is used to gauge how reasonable a protein conformation is when one considers various interactions between the atoms of the protein and the solvent. It takes into account charged groups, hydrogen bonding, ideal bond lengths and angles, and hydrophobic interactions (23). Instantiation, in this case, involves repacking the side chains of a Combo and evaluating the Rosetta energy function on the result. Structures with lower Rosetta energy scores are more plausible protein conformations. A suitably trained regression model could dramatically accelerate the identification of the most favorable combinations according to the Rosetta energy function.

The final method of instantiation that we demonstrated is somewhat more complicated. We endeavored to fill low-resolution X-ray data (XRD) with fragments of homologous proteins. This challenge is highly practical; determining quality structures from low-resolution X-ray data is a contemporary challenge. The validity of a structure that fills X-ray data is typically measured via the R-values, R_{work} and R_{free}. R-values are a measure of how well a model explains collected XRD. The predicted structure factors are compared to the observed structure factors, normalized, and summed. R_{work} is used to guide XRD interpretation and contains the majority of structure factors, while R_{free} consists of structure factors held in reserve as a sanity check. If there is a large R_{work}-R_{free} gap, the model is said to be biased (in favor of R_{work}). R-values are widely used in protein structure determination.

However, we devised an alternate metric that was superior for the regression-based optimization process. Specifically, the score coming out of the Instantiation and the target quantity for the regression model was the Accumulated Correlation Coefficient (ACC) score as defined below.

$$ACC = 100 * \sum_{n=1}^{N} \frac{1 - CC_n}{N}$$

Correlation Coefficient (CC) is the correlation between the implied density surrounding a residue *n* in a proposed structure and the density calculated from X-ray data and the proposed structure. Density is measured at a cloud of 'probe points' around each residue, and a CC of 1 represents perfect correlation. PHENIX can generate a list of CC scores for a structure it has refined (Adams 2010). The sum of all of CC for all of N residues captures how well an entire structure fits into XRD.

We used ACC to measure how well a Combo filled X-ray data for a few key reasons. Chief among them is that poor real-space correlation coefficient (CC) in a single region is likely to noticeably penalize the entire score, whereas R_{work} might shift imperceptibly. However, the per-residue normalization and the fact that good scores are closer to zero were also features we desired.

Note that, initially, ACC score was not normalized by residue. Without normalization Poses that are missing residues are misidentified as more accurate solutions; their ACC score would be inherently lower. At first this was a non-issue, as every Pose had the same number of residues; however, with increasing variety among the source structures it became an issue and was corrected. The un-normalized ACC score will be called uACC, and its equation appears below.

$$uACC = \sum_{n=1}^{N} 1 - CC_n$$

III. INITIAL TESTS AND RESULTS

ROSETTA ENERGY FUNCTION AND REPACKING

Predicting the Rosetta energy function was our first test of the code's functionality. The goal is to see if the Rosetta energy is a good metric for distinguishing plausible Combos. We divided the (relatively) small protein 3FYM (130 residues) into 6 Blocks for this test.

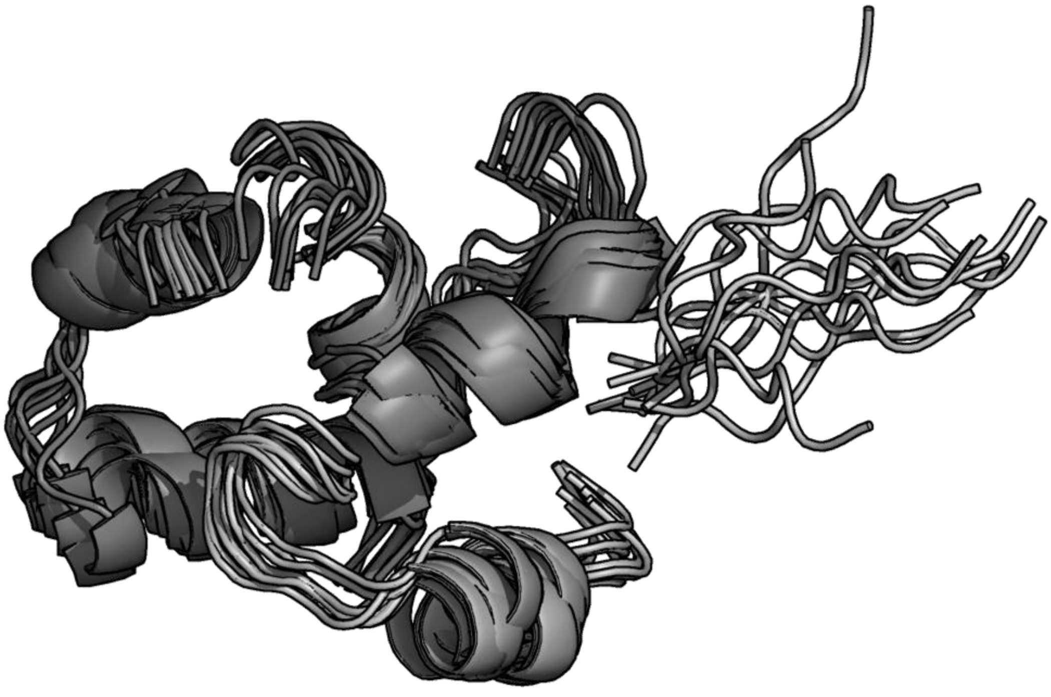

Figure 4: Blocks and Poses for protein 3FYM. There are 6-blocks, each with 13 Poses generated via a normal mode sampling approach.

The first test was to see if the repack-Rosetta method of instantiation could distinguish the Gold Standard from other positions. Alternate poses were generated using Normal modes. Instantiation included a 'linker rebuilding' step where the ends of each of the Poses were rebuilt in order to form a contiguous backbone. In this case the Gold Standard was the original 3FYI structure from the Protein Data Bank. This test is not arbitrarily easy; it is not obvious that the

Gold Standard will have the lowest Rosetta energy score, or that the software will correctly recognize that the combination of Poses from the Gold Standard will produce that score. Perhaps the poses from the Gold Standard are only favorable in conjunction with each other? Maybe the search space is not favorably shaped, and other combinations are local minima? However, the test was mostly successful; the Rosetta energy typically allowed the regression to choose the Gold standard at 7 of 8 positions.

SSD TESTING

The RegressionProblem code was next applied to a simple test problem; use fragments of proteins to directly build the structure of another protein. Instantiation in this case is simply measuring the SSD difference between the fragments indicated by a Combo and the Goal structure. This sort of test is less difficult, but it has implications for the use of fragments in these sorts of problems. Specifically, SSD testing can be used to determine whether the generated space can recapitulate the Goal at all. It removes the intermediate requirement of distinguishing reasonable answers by measuring Combos in some way. This technique obviously cannot be used to solve real problems (for which there is not yet an answer), but it is important to show that the method can get to an answer if one exists.

In this case, the Goal structures were cytochrome P450 Chimeras made by the Arnold lab (Otey, 21, and Li, 20). The source structures were other Chimeras that have been successfully crystallized. Poses came from Chimeras with an identical Parent at a specific Block within the protein. No sequence alteration was performed.

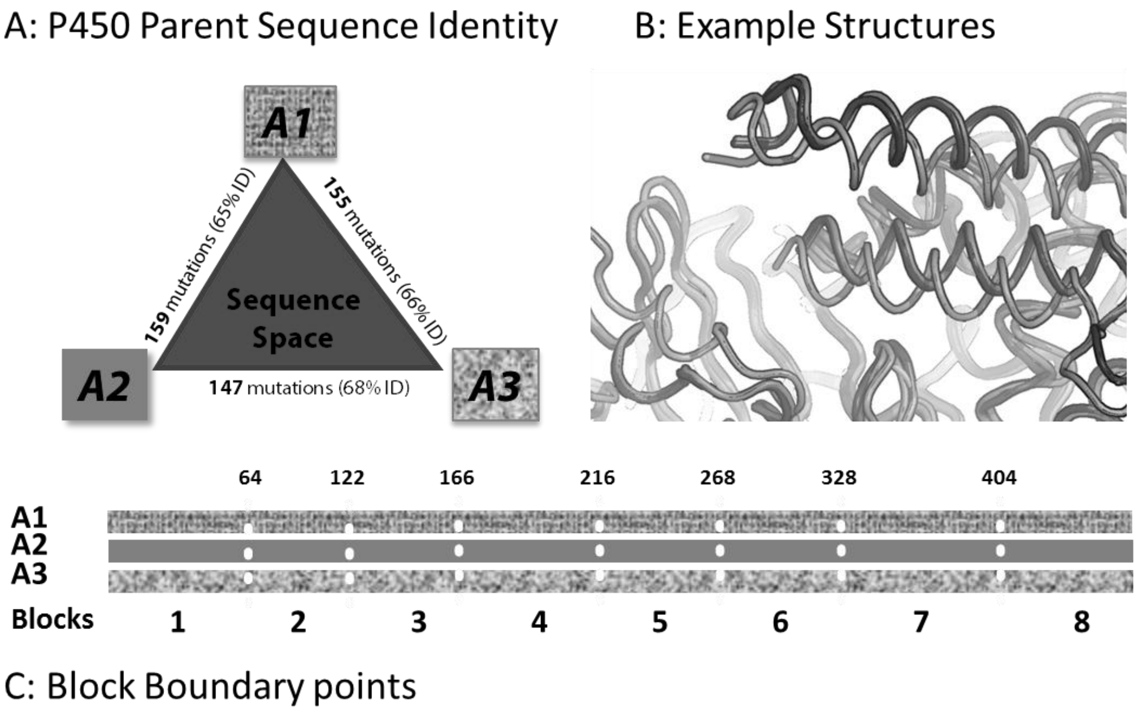

Figure 5: Overview of P450 Chimeras. (A): Chimeras were constructed from 3 P450s that served as parents. They share 65-68% sequence identity. (B): PyMol visualization of P450 Chimera Poses that have been moved. The closely aligned structures are 0.25Å distant while the third structure is 1.7Å away. (C): Block boundary points and the residue number they occur at. With 3 Sources and 8 Blocks, there are $8^3 = 512$ possible Combos.

The simplest test was to hide the Goal structure itself among the Sources. With instantiation being a simple comparison of structures, it was easy for the regression to recognize the exact answer among the sources provided. The initial search for the ‘correct’ answer among others was mostly a sanity check.

A more challenging test involved seeing if a good answer is available among the structures from all other Chimeras. This is an important question; if the Chimeras cannot approximate each other structurally, there is no point in downstream attempts to predict the favorable Chimera structure. The idea was to apply our technique to Goals with known structures to assess the likely utility of the method for predicting unknown structures.

The second test was also successful; the search space does indeed contain structures close to that of the Goal. The simple SSD instantiation method demonstrated that the search space contained solutions for two test Goal structures, X38.4 and X8.5, with SSD<600 (or RMSD<1.18).

IV. FITTING X-RAY DATASETS VIA RECOMBINATION

CHIMERAS AND UACC

For our primary application focus the specific goal was to accurately recreate a Chimera's structure using the X-ray data for that chimera and the structure of other Chimeras. Instantiation consisted of PHENIX rigid-body refinement followed by ACC scoring. The ACC score was not normalized during these experiments, and will be called uACC (unnormalized ACC) for short.

The first test of this instantiation method was to distinguish the Gold standard from other Sources. A specific chimera named X8 was chosen as the Goal. The dataset was low-resolution (3.06Å), motivating the use of recombination as an alternative to conventional refinement methods. We were cognizant of the possibility that other chimera fragments could better fill the X-ray data than the previously made structure. The previously made XRD structures might not be perfect; it is possible that the Gold Standard, or the structure previously made to explain the XRD, could be worse than the best Combo. Also, the Gold standard uACC had to be quantified before it could be used as a benchmark, so we therefore Instantiated the complete Gold Standard at the outset. It had a uACC score of 87.6. If any combination produced a score of 87.6 or lower we would consider it a success. We also coded the capability of considering uACC score by Block to gain insight into each Pose's contribution. To Instantiate large numbers of combinations we used the ENS HPC computer cluster. The chimera had 24 Blocks (8 Blocks for each of 3 Chains) with 11-53 Poses per block. The total search space size was $1.11 * 10^{36}$ possible combinations.

A normal weighted regression with only single terms was only able to produce a lowest uACC score of 92.2, with six Poses of 24 different from the Gold standard after 3 rounds. In response, we altered the regression in various ways to see what would be necessary for the Gold

Standard to be found. It became apparent that a certain Combination, which we termed the 'Silver Standard', was often the endpoint for our calculations. It differed at 4/24 Poses.

Table 1: Summary of Strategies to improve Regression Performance on Chimeras

Strategy	Result
Single terms	Mismatch at 6 positions
All double terms	Overfitting, unable to effectively search
Contiguous double terms	Silver standard (mismatch at 4 positions)
High-error triples	Unable to effectively search
Low-scoring triples	Unable to effectively search
Answer 'seeded'	Silver standard
Answer 'seeded' with high energy weighting	Silver standard
Training set is a set of low-scoring Combos	Silver standard
Training set is 2-substitutions away from Gold Standard	Silver standard

At first we applied changes that we would use on a real problem; eventually we tried debugging by directly encouraging the algorithm to find the answer. Most methods ended up converging to the Silver standard, which had a uACC score of 88.8.

After repeated failed attempts that all chose the same solution, we became suspicious of an error in our assumptions. It took a while to track down; the Block-by-Block instantiation which was used on the Gold standard mistakenly skipped the last residue of each Pose, forcing a uACC score depression of 4. The real Gold Standard value was 91.7, indicating that the Silver standard was actually better scoring than the Gold Standard. The regression had been succeeding for months.

The next step was to determine the level of regression training required to find the best answer. X8 remained the Goal, but it was removed from potential Sources. The new problem's lowest scoring discovered Combo has a score of 90.29. Various initial training set sizes were tested while various numbers of targets were added in each round. The results appear below.

Table 2: Regression Performance for X38.8 Chimera Target versus Training Set Size

Target set size:	3000		1000		500		100	
Initial Size	Score*	Rounds	Score	Rounds	Score	Rounds	Score	Rounds
10000	91.6296	11	90.6936	9	90.2942	10	91.6136	9
5000	91.6891	6	90.6936	8	90.2942	8	91.4116	12
2500	91.6891	9	90.6936	11	90.2942	12	91.4116	13
1250	91.9159	8	91.8034	9	94.3083	12	104.3206	6
625	91.6136	9	98.0664	12	112.8274	2	111.3412	3

*The best score in any round

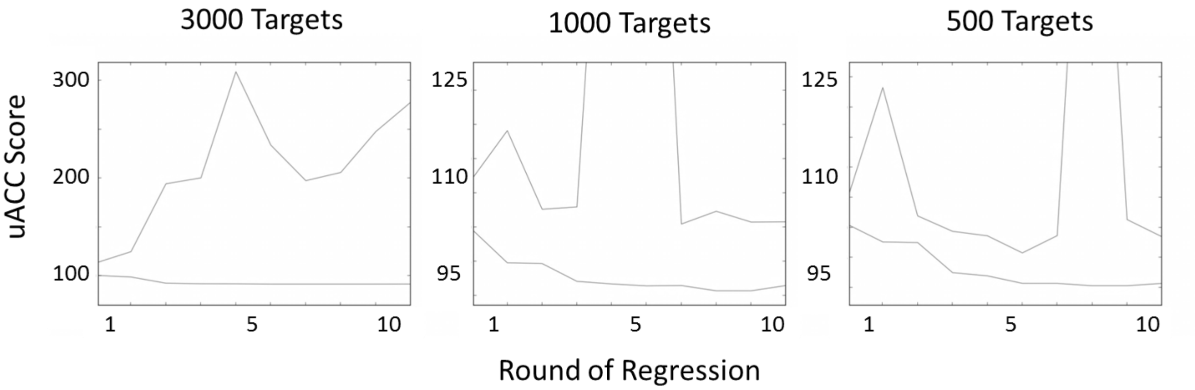

Figure 6: The convergence of the regression with a starting size of 5000 Combos and varying numbers of targets. The upper line is the average score of the newly instantiated Targets during each round, while the lower line is the lowest score. With too many targets, the average climbs endlessly. The middle and right figures have spikes that represent the regression temporarily making poor predictions, in which the average climbs rapidly when a group of targets are mistakenly thought to be low-scoring, and lowers again when the group of targets is better approximated.

In this case, the strategy was weighted regression with 4/5ths of the initial set forming the training set. The experiment had an important result; as long as the size of the original training set is past a certain minimum, the final convergence depends mostly upon the number of targets per round. Initial set size is still of paramount importance. With too small of an initial set it cannot make progress toward a solution, but with too large of a set the regression required a greater number of rounds to converge. It may be confusing that large initials sets slowed the progress of the regression in terms of number of rounds, but this can be rationalized as follows: with a large initial set the (poorly scoring) initial Combos cause the regression to prefer specific values for the terms that accurately reflect the contribution of each Pose to *poorly scoring* Combos, but not *favorable* Combos. The regression performance suffers when it must compensate for a larger amount of “noise” in the form of poor-scoring combinations.

Other interesting phenomenon occurred during the regression process. In some cases, presumably due to insufficient Target diversity, the regression appeared to converge upon local minima. In that case, the lowest Target energy stagnated. Additionally, the average Target energy rose, which is difficult to explain. What is clear, is that a balance must be struck between rapid progress and flexibility.

The tests of uACC on the set of Chimeras provided insight into the functionality of our software. They also afforded opportunities for bug fixing and expansion of capabilities. However, we realized that Chimera recombination did not provide a sufficiently rapid route to publication; if the software became too specialized for Chimeras, it would be contrary to our goal of easing general searches through spaces composed of protein fragments. Protein Chimeras might behave in specific ways particularly amenable to Regression. We therefore changed direction to focus on a protein family with published structures.

A NEW SEARCH SPACE: TEM1 HOMOLOGOUS PROTEINS

Instead of continuing work with Chimeras, we switched to the β-lactamase family, specifically proteins homologous to TEM1. Chimeras that have been recombined and recrystallized might be a special case, and our goal was to show the general nature of our algorithm. In addition, we wished to firmly show that our method could be a starting point for the interpretation of low-resolution X-ray crystallography data.

TEM1 is a β-lactamase, a family of proteins with plentiful structural data. A set of 137 homologous proteins was gathered with the BLASTP (Altshul 1997) alignment tool. From this set, multiple Goal structures were chosen; the structures were ordered from low to high resolution, with low-resolution structures favored as Goals. For each Goal, the X-ray crystallography data available in the Protein Data Bank was downloaded and truncated. Any reflection of resolution < 3 Å was removed. While this is not the ideal method to simulate low-resolution data, alternative methods such as simulating low-resolution data using the MLFSOM program proved difficult to automate. Additionally, Goal structures were required to be single chain— only PDB entries with one backbone in each unit cell were considered. This constraint merely simplified the problem and ensured that the performance on each Goal structure would be directly comparable to that of the other Goal structures. The final constraint placed on Goal structures was that they must be different from all previously chosen Goal structures, with a sequence identity (I) of less than 0.8. This means that the protein backbone between two Goal structures was not allowed to match at 80% or more of the positions. Sequence identity was calculated from the output of the Tcoffee alignment program (NotreDame 2000), which takes advantage of known PDB structures to align sequences.

The Sources used for the recapitulation of each Goal were restricted to homologous proteins with sequence identity between 0.3 and 0.8 that were not also Goal structures themselves. Valid Sources were similar to a specific Goal structure (I>0.3), but not so similar as to be redundant (I<0.8). The difference requirement was meant to ensure that point mutants and other structures with too much similarity to the Goal did not weaken the test of the method; the similarity requirement prevents the search space from being unduly large. If Source proteins had multiple backbones per unit cell in their PDB entries, each backbone was considered separated and multiple Poses were taken from that Source.

The sequences of all Source structures for each specific Goal structure were aligned with Tcoffee one more time. Redoing the alignment ensured that restricted structural information from Goal and non-Source structures would not accidentally influence the sequence alignment.

For each Goal structure, the Source structure with the greatest sequence identity to the Goal was set aside as a Basis. All Sources including the Basis were then remodeled by replacing sidechains with the cognate Goal structure amino acids. Sidechain positions were optimized with the FasterPacker combinatorial rotamer optimization algorithm (Desmet 2002) in SHARPEN—its original use. Next, all mutated Sources were aligned to the Basis. This alignment step is essential for choosing Block Boundaries.

Block Boundaries were initially chosen using Divergence. This led to 11-16 Blocks per Goal structure. Each Source was chopped into Poses. Then, Affinity Propagation was used to eliminate redundancy among the Poses, and a final search space was constructed. For example, we attempted to recapitulate the structure of 2CC1 by dividing the sequence into 16 Blocks. The number of non-redundant Poses for each Block was between 71 and 178 (geometric mean 167.8) leading to a search space size of $3.96*10^{35}$ possible Combos.

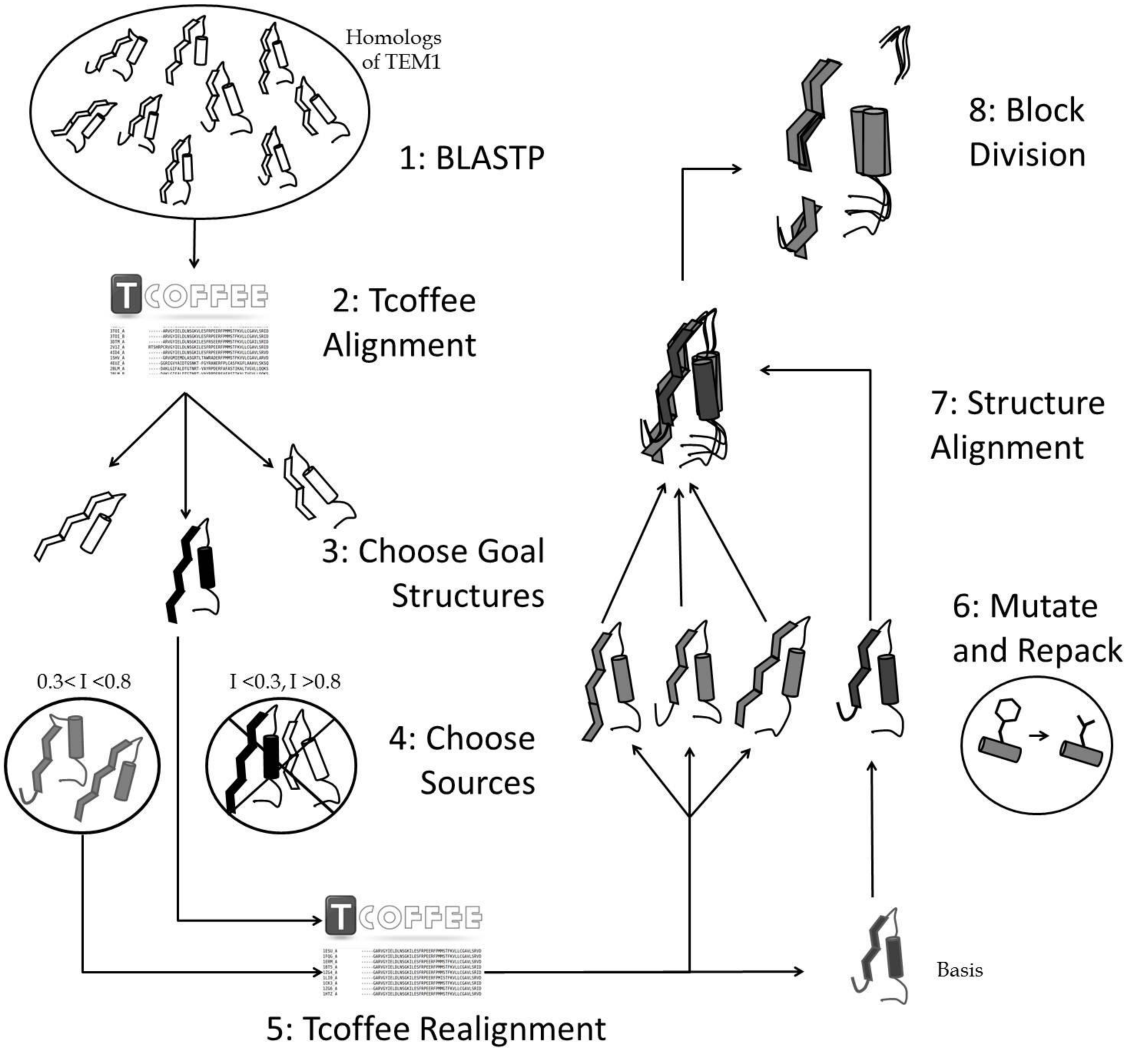

Figure 7: Setup steps for applying our regression software to a set of homologous proteins. 1: Gather the homologous proteins with BLASTP. 2: Use Tcoffee to align them. 3: Choose which proteins will be goal structures. 4: Choose which proteins will be sources. 5: Realign with Tcoffee to avoid bias. 6: Mutate each sequence to match the Goal's sequence, and repack the sidechains. 7: Align all structures. 8: Perform block division with Divergence.

TEM1 HOMOLOGOUS PROTEINS: ITERATED REGRESSION

For each Goal structure we Instantiated 20,000 Combos. This dataset was randomly divided into 16000-member training and 4000-member test sets. 20,000 was chosen to account for the larger number of possibilities within the search space; earlier experiments indicated that the regression will work as long as a certain minimum number of starting Combos are present.

Only single terms were used. We tested higher order terms, but we were surprised to find that they did not afford a noticeable improvement in the accuracy of the regression This was a striking finding given that the correspondence between a model and X-ray diffraction data is explicitly a function of the entire structure, and that contiguous terms at least would improve performance when predicting P450 Chimeras

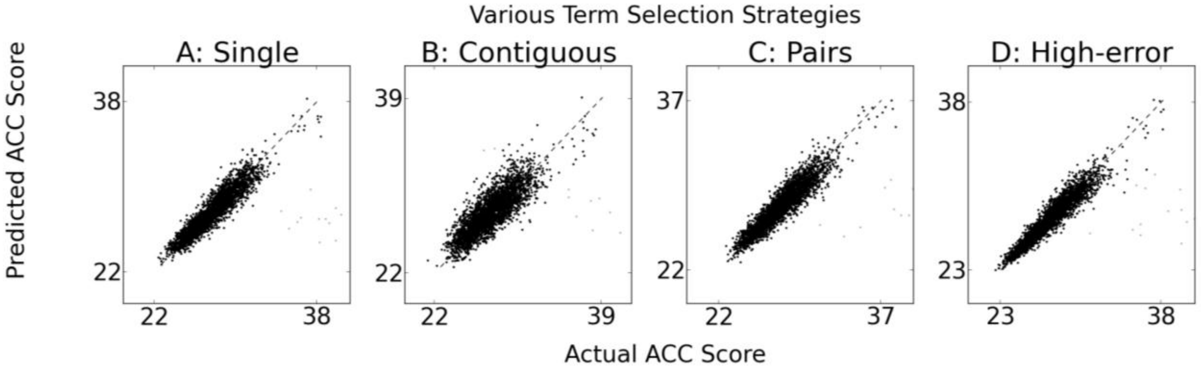

Figure 8: **Various Term Selection Strategies**

The test set held in reserve for each problem was used to gauge how well combinations of terms could approximate the actual ACC score. We define outliers as Pose combinations with an actual ACC score 5 units higher than predicted for the rest of the targets. Less than 1% of random combinations are outliers. (A) The simple approximation contains only 1-body terms and achieves a RMS error (for non-outliers) of 0.76. (B) Surprisingly, including 2-body terms for adjacent Poses (i.e. contiguous terms) decreased the quality of the predictions for the test set (RMS error = 1.26) despite using a training set of 16,000 combinations. (C) Adding terms for all possible Pose pairs reduced the RMS to 0.77, approximately that of single terms. (D) Terms meant to capture pairs and triples with high error finally afforded an improvement: RMS 0.66.

In the case of the homologous proteins, there were $\sim 15 * (\sim 168)^2 = 400{,}000$ possible contiguous double terms. This excessive number might account for the inaccurate nature of the regression with contiguous double terms, but why would the pairwise regression perform better with its astronomically large number ~120*(~168)^2 of terms? Fortunately, single terms proved sufficient in guiding the regression toward accurate models, as explained in the next section.

Another mystery was the presence of outliers in the predicted data. For random Combos, a small number (less than 1%) of Combos are much harder to predict. We rationalized these Combos as particularly difficult for PHENIX to place into the X-ray data; they probably represent a cutoff where PHENIX struggles to improve their placement and simply leaves them where they are initially placed. Chimeras did not produce this issue because Poses from Chimeras are inherently more similar to each other and easier to work with. The absence of extreme outliers in Targets chosen by the regression supports this hypothesis: low-scoring combinations are necessarily amenable to improvement with PHENIX.

1000 Targets that were predicted to have a low ACC score were added to the training set during every round. Despite the utter simplicity of the regression model, it was possible to "learn" which Combos produce the lowest ACC scores. If a batch of Targets contained mostly false positives, its true nature is revealed during instantiation and the misleading Combos are eliminated after a round of regression reveals them to have slightly higher scores. By repeating this process 4 to 15 times, we achieved effective machine learning. We judged a regression model as completely trained (converged) when the best (lowest) ACC score among the newly chosen Combos did not improve from one round of regression to the next.

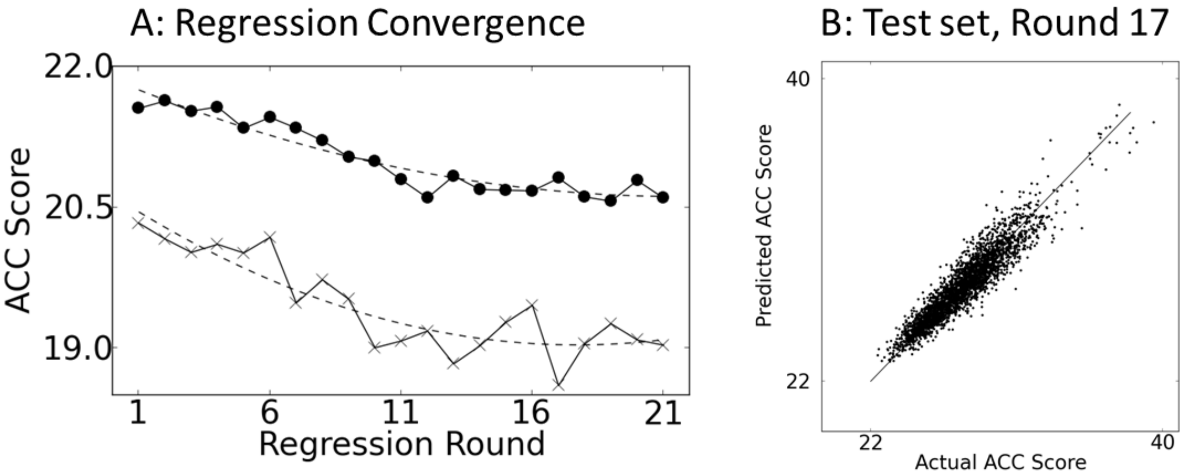

Figure 9: (A) Convergence of the regression can be seen in the gradual improvement of the average Target score (upper line, •) and best Target score in each round (lower line, ×) for Goal 2CC1. Dotted lines are a 2nd-order polynomial fit to illustrate decreasing magnitude of improvements in score for both measures of improvement. (B) The test set affords a sanity check on Regression performance during the round with the lowest Target score (RMS 0.778). 29 outliers (ACC discrepancy >5) of 1000 Combos have been removed.

In addition to training the regression over several rounds, we also used the initially reserved test set to verify the accuracy of our approximation. The verification could be done at each step to check the progress of the regression, but it did not influence the regression itself.

EXPLANATION FOR THE SUCCESS OF SINGLE TERMS WITH X-RAY DATA.

Two subtleties in technique contributed to the effectiveness of this strategy. The first is that the Targets were not chosen simply by taking the lowest predictions (which are trivially easy to find when the prediction is a sum of first-order terms). Instead, we found the lowest prediction and used Monte Carlo selection to choose other targets. Temperature was increased until a minimum number of Combos were readily found. This allowed variety to be preserved in the event that many Combos are suitably low-scoring; the strictly lowest set of Combos might not be representative of low-scoring Combos in general. Additionally, this method of Target selection is still effective if higher order terms are included and the lowest scoring predictions are not trivial to determine.

The second subtlety is that our regression approach was weighted to favor accuracy in low-scoring Combos. The goal was to more accurately predict the ACC for the Combos that better explained the X-ray diffraction data, at the possible expense of being unable to predict poorer scoring Combos. This means that after every round of regression, the old Targets strongly influence the Regression terms for the next round unless they score poorly, in which case they carry little weight. This is part of the explanation for why regression with only first-order terms is so successful; the weighting of score in the neighborhood of the optimal Combo provided another area of flexibility in the method.

COMPUTATIONAL TIME

Initially, 20,000 random Combos were evaluated for each Goal structure using rigid-body PHENIX refinement calculations. Each of these refinement calculations required approximately 420 seconds for a Combo containing 16 Blocks. Thus, a brute force search through the possible Combos would require 10^{27} to 10^{31} years, depending on the Goal structure. As discussed in section V, multiple rounds of regression were effective in improving the score of the models. 15 rounds of regression with 1000 Targets each, and an initial set of 20,000 random Combos, results in a computational time of 170 CPU days. However, this technique is amenable to parallelization; wall time was reduced to 12 hours with a compute cluster containing 350 cores.

V. TEM1 RESULTS AND DISCUSSION

INITIAL SET AND RESULTS

The initial results appear in the table below. A comparison was made between complete Source structures, the Goals themselves, Random combinations, and the best Combos as determined by ACC score. Whole Sources and Goals were tested, as were Goals and Sources divided at the same Block Boundary points used to generate the Combos. With the exception of Goal 4JLF, all regression problems were able to choose a Combo that would be a better starting point than any single source structure. It can be seen that the best Combos recapitulate the Goal structure to better than 0.7Å without human input.

Table 3: 3Å Truncation Recombination-Based Learning Summary

	Best Scoring		Random Average		Whole Sources		Whole Goals		Divided Sources		Divided Goals	
Goal Structure	ACC	RMSD	ACC	RMSD	ACC	RMSD	ACC	RMSD	ACC	RMSD	ACC	RMSD
2CC1	19.95	0.67	27.93	2.48	26.85	1.38	11.35	0.05	23.11	1.23	11.25	0.15
3LY4	20.36	0.71	30.16	2.39	28.28	1.04	15.07	0.03	23.37	0.90	13.69	0.14
1CK3	17.42	0.37	27.43	2.03	21.47	1.04	13.76	0.03	18.02	0.94	13.67	0.12
4JLF	20.88	2.17	27.26	2.96	25.67	1.36	9.60	0.01	21.11	1.22	9.04	0.12
1TDG	18.76	0.66	26.78	2.12	23.16	1.21	13.76	0.03	19.37	1.10	13.50	0.14
3BYD	15.76	0.46	28.54	2.55	20.92	0.56	12.28	0.05	18.76	0.47	12.35	0.13

RMSD values for the lowest scoring Pose combinations as measured by ACC tend to be much more favorable than those of random Pose combinations: between 0.8 and 2 lower (Table 3). The Sources could not approach the combinations' level of accuracy for 4 of the 6 Goal structures, and in the case of 3BYD the best combination was comparable to the best Source. Only goal 4JLF was poorly approximated by the best combination. Dividing the Sources at the Block Boundary points allowed them to perform slightly better, but still not as good as the combinations. As expected, unaltered Goal structures put through a rigid-body refinement come closest to the authentic PDBS. The Goals, however, did best when undivided: division increased alpha carbon RMSD from 0.02-0.05 to 0.11-0.15 (Table 3). Alterations can only worsen a near-perfect model. Notably, ACC could be used to rank all of the structures by RMSD save for the undivided goals, which sometimes score slightly worse than the divided goals despite having a better RMSD.

RATIONALIZING THE COMPARATIVELY POOR PERFORMANCE FOR THE 4JLF GOAL

An examination of the resulting structures allowed for insight into the relative failure of our method to recreate Goal structure 4JLF. Two factors in particular appear to be contributing to the failure. First, there is a section of 4JLF with a shape that cannot be approximated with any Source. All sources sacrifice accuracy at a protein loop to maintain accuracy in a beta sheet. It is likely that carefully placed additional Block Boundary points would allow the problematic loop to be reconstructed. Second, another section of 4JLF appears to frequently be truncated in Target combinations. A single short Pose provides a pathological solution. This truncation is preferred in ACC score because shorter Poses are less likely to face ACC score penalties from poorly approximating structure; adding a penalty to the ACC score for missing residues could potentially eliminate this sort of error. PyMol was used to discover these discrepancies.

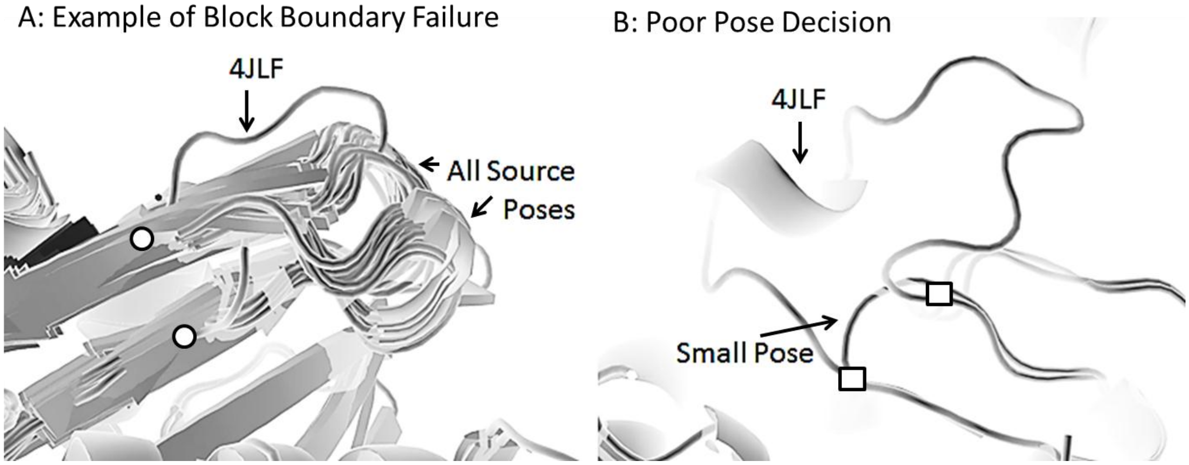

Figure 10: 4JLF and two potential explanations for poor performance. (A): No source pose was able to approximate this loop. All poses sacrificed loop accuracy to better fit the adjacent beta sheet. Superior Block division might occur at the highlighted Residues (○). (B): A small pose was universally chosen to replace a loop. This caused the backbone to skip structure at the highlighted regions.

LOWER-RESOLUTION TARGETS

Further truncation was tested upon three of the Goals. A summary of the results appears in the table below. These are the lowest RMSD values found in each set of data. 1000 random combinations were compared to the 1000 Targets from the round with the lowest ACC score.

Table 4: RMSD from Goal for Best Combo as a Function of Increasing Truncation

	3 Å		4 Å		5 Å		6 Å	
Goal	Best Round	Random Combos	Best Round	Random Combos	Best Round	Random Combos	Best Round	Random Combos
4JLF	1.16	1.21	1.15	0.91	0.88	1.32	1.05	1.10
3BYD	0.47	0.74	0.46	0.75	0.54	0.79	0.96	0.92
2CC1	0.24	0.49	0.29	0.67	0.49	0.71	1.21	0.99

For 3BYD and 2CC1, it can be seen that regression finds a better solution in the 3, 4, and 5 Å cases. At 6 Å resolution, the regression can no longer effectively search and random combinations are comparable to the best of chosen Targets. 4JLF behaves somewhat oddly in this case; resolution truncations actually allow the regression to perform more efficaciously with this goal structure. We suspect that the failure to accurately recapitulate 4JLF in the higher resolution case results from incorrect or misleading sidechain positions for the Poses. As resolution is lessened, it would reduce the impact of sidechain position on the PHENIX calculation as there is not sufficient information to determine the location of the sidechains.

R_{WORK} AS A SCORING METRIC

R_{work} was tested as an alternative to CC within our regression software. The experimental procedure was identical, save for R_{work} being the scoring metric instead of CC. This resulted in additional bias. Moreover, R_{work} did not guide our software to reasonable solutions. Improvements in R_{work} were smaller and required more rounds of regression.

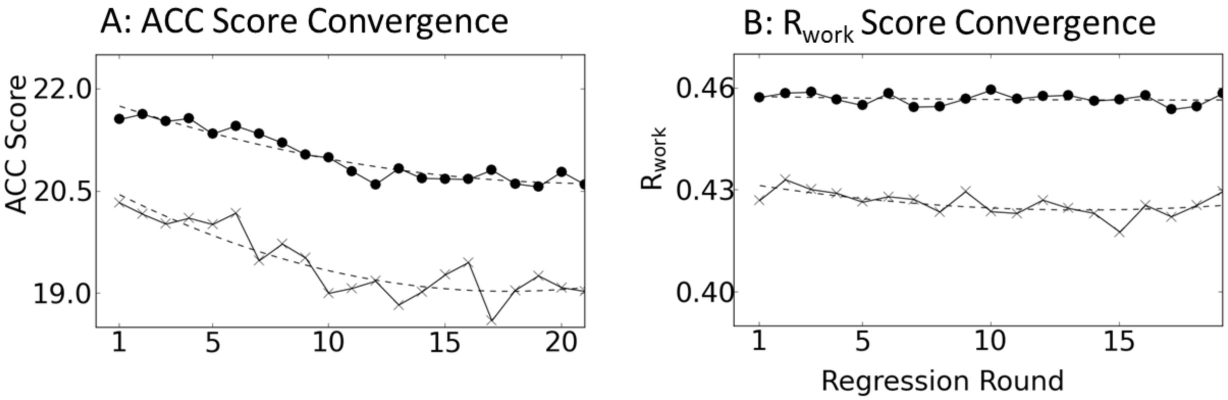

Figure 11: Comparison of ACC (A) and R_{work} (B) as guiding functions for interpreting low-resolution X-ray data for 2CC1. The top line in each graph is the average Target score for each round of regression, while the bottom line is the best score among that round's Targets. Dotted lines are second-order fits for depicting the general trend. Improvements are possible to both metrics, but ACC appears to be more easily optimized.

Although R_{work} did not perform as wells as ACC in guiding the regression, the two scoring methods shared an important characteristic. Bias was not increased when fitting low-resolution data with either method. This is an important result when interpreting low-resolution data.

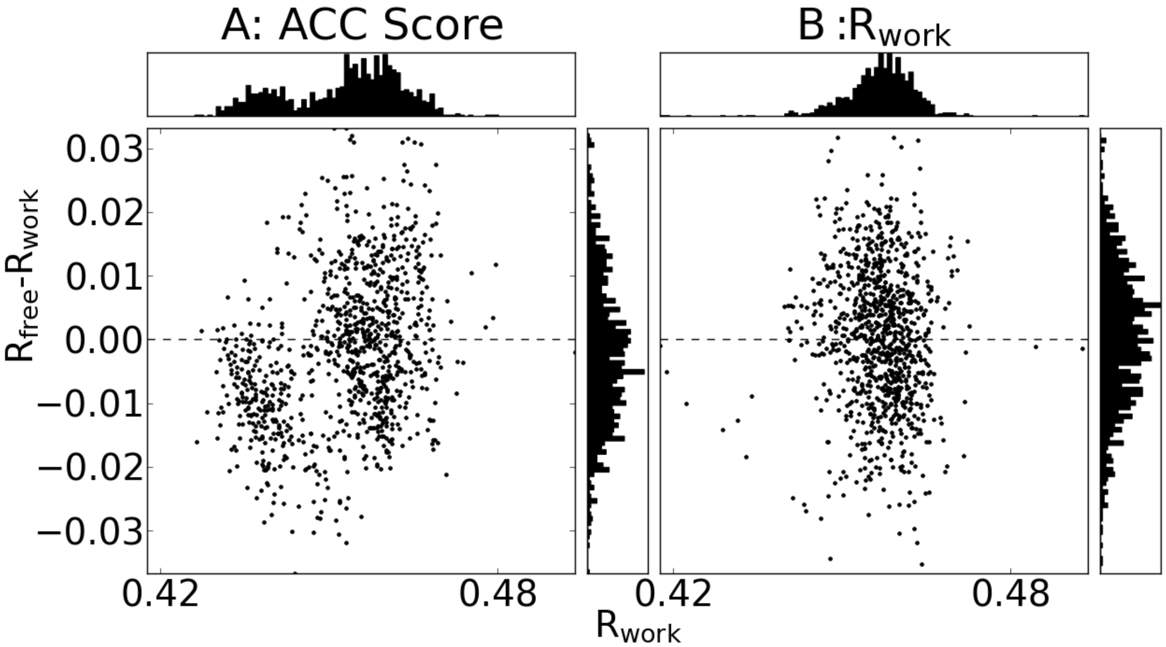

Figure 12*: Bias among Targets.* This figure illustrates the correlation between the ACC score (A) or R_{work} score (B) predicted for Target combinations vs the Bias in R-values for the Targets. Average Bias for ACC: -0.002. Average Bias for R_{work}: 0.0003.

ADDITIONAL ROUNDS OF UNRESTRAINED REFINEMENT

We envision our method as a starting point for further optimization of structures in low-resolution data. As a preliminary test, we used automated and unrestrained (not rigid-body) refinement with PHENIX to attempt to improve the results of the regression. This relaxation of restrictions is expected to increase the R_{work}-R_{free} bias in the Targets from the regression round with the lowest ACC score. Limiting the number of rounds of non-rigid refinement can limit the increase in bias. The results appear below.

Table 5: Assessing the Impact of Unconstrained Refinement

	0 rounds		1 round		2 rounds		3 rounds	
Goal	Average RMSD	Average Bias	Average RMSD	Average Bias	Average RMSD	Average Bias	Average RMSD	Average Bias
2CC1	0.795	-0.002	0.751	0.018	0.693	0.035	0.656	0.044
3LY4	0.776	-0.007	0.712	0.006	0.668	0.017	0.642	0.022
1CK3	1.035	0.017	1.022	0.047	1.014	0.062	1.009	0.067
4JLF	2.449	0.002	2.433	0.022	2.414	0.042	2.396	0.052
1TDG	1.178	0.024	1.149	0.042	1.142	0.051	1.141	0.054
3BYD	0.499	0.008	0.443	0.033	0.416	0.052	0.405	0.053

As expected, the average bias among the targets went up. However, the RMSD of all targets (excluding those for Goal 4JLF) went down with additional rounds. An average improvement of 0.086 Å is possible before the average R$_{work}$-R$_{free}$ gap increases to 0.04, just below the widely accepted cutoff of 0.05. It is worth noting that improvements made this way have varying effects on the various Goals. 3LY4 achieved an improvement of .13 Å while the bias remained below 0.02; to contrast, 1CK3 only achieved an improvement of 0.026 while passing the acceptable limit for bias. We expect that manual refinement steps could achieve even better results with these low-resolution structure starting points; a refinement that is manually customized to each Goal would probably do better.

VI. METHOD OUTLOOK

The results above suggest that fragment recombination has the potential to successfully accelerate otherwise challenging searches over protein conformations. Here, we show that the method can be used to automate the refinement of high-quality models to fit low-resolution X-ray data. In this case, the search space is semi-discrete since the Cartesian position of the rigid Poses is optimized by PHENIX. In crystallography laboratories (such as the Snow lab) that routinely produce low-resolution crystallographic datasets for a family of related crystals, this framework could provide a significant increase in the quality of the structural models and an immense saving in the time spent in manual refinement of the crystallographic models.

In principle, the same strategy and the same code platform can be applied to protein structure prediction in the absence of experimental data. In particular, this method could be used to compete in the “structure refinement” category of the biennial Critical Assessment of Structure Prediction (CASP) blind structural determination competition. The structure refinement category might also be called “homology model remediation” in which teams attempt to improve upon a slightly-incorrect starting model.

Incorporating protein flexibility and conformation change is also a perennial challenge for other algorithms such as protein-protein docking, inhibitor design, or fitting detailed protein models into low-resolution cryo-electron maps. The methodology described herein might also be used to provide efficient protein conformational search in those scenarios. For example, let us consider computational protein design. Traditionally, it has been difficult to mesh protein backbone flexibility with the combinatorial optimization of amino acid sidechain positions and identities that constitutes the design step. Here, however, we need only consider Poses containing alternative amino acid sequences to convert the presented method to a protein design calculation.

In sum, the method speeds calculations and allows for rapid exploration of otherwise immense search spaces. It was designed to be generally applicable. To ensure continued use, the software has been extensively documented, and several tutorial examples have been prepared to bring new users up to speed.

REFERENCES

Adams, P. D. et al. PHENIX : a comprehensive Python-based system for macromolecular structure solution. Acta Crystallographica Section D Biological Crystallography 66, 213–221 (2010).

Altschul, S. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research 25, 3389–3402 (1997).

Apgar, J. R., Hahn, S., Grigoryan, G. & Keating, A. E. Cluster expansion models for flexible-backbone protein energetics. Journal of Computational Chemistry 30, 2402–2413 (2009).

Bloom, J. et al. Evolving strategies for enzyme engineering. Current Opinion in Structural Biology 15, 447–452 (2005).

Borhani, D. W., Rogers, D. P., Engler, J. A. & Brouillette, C. G. Crystal structure of truncated human apolipoprotein A-I suggests a lipid-bound conformation. Proceedings of the National Academy of Sciences 94, 12291–12296 (1997).

Bourne, Y., Grassi, J., Bougis, P. E. & Marchot, P. Conformational Flexibility of the Acetylcholinesterase Tetramer Suggested by X-ray Crystallography. Journal of Biological Chemistry 274, 30370–30376 (1999).

Das, R. & Baker, D. Macromolecular Modeling with Rosetta. Annual Review of Biochemistry 77, 363–382 (2008).

Desmet, J., Spriet, J. & Lasters, I. Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. Proteins: Structure, Function, and Genetics 48, 31–43 (2002).

Ehrenreich, H. & Turnbull, D. Solid state physics advances in research and applications. Volume 47 Volume 47. (Academic Press, 1994). at <http://site.ebrary.com/id/10259520>

Emsley, P., Lohkamp, B., Scott, W. G. & Cowtan, K. Features and development of Coot. Acta Crystallographica Section D Biological Crystallography 66, 486–501 (2010).

Frey, B. J. & Dueck, D. Clustering by Passing Messages Between Data Points. Science 315, 972–976 (2007).

Grigoryan, G. et al. Ultra-Fast Evaluation of Protein Energies Directly from Sequence. PLoS Computational Biology 2, e63 (2006).

Hahn, S., Ashenberg, O., Grigoryan, G. & Keating, A. E. Identifying and reducing error in cluster-expansion approximations of protein energies. Journal of Computational Chemistry n/a–n/a (2010). doi:10.1002/jcc.21585

Holton, J. M., Classen, S., Frankel, K. A. & Tainer, J. A. The R-factor gap in macromolecular crystallography: an untapped potential for insights on accurate structures. FEBS Journal 281, 4046–4060 (2014).

Jiang, W., Baker, M. L., Ludtke, S. J. & Chiu, W. Bridging the information gap: computational tools for intermediate resolution structure interpretation. Journal of Molecular Biology 308, 1033–1044 (2001).

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by Simulated Annealing. *Science* **220,** 671–680 (1983).

Koparde, V. N., Scarsdale, J. N. & Kellogg, G. E. Applying an Empirical Hydropathic Forcefield in Refinement May Improve Low-Resolution Protein X-Ray Crystal Structures. PLoS ONE 6, e15920 (2011).

Langer, G., Cohen, S. X., Lamzin, V. S. & Perrakis, A. Automated macromolecular model building for X-ray crystallography using ARP/wARP version 7. Nature Protocols 3, 1171–1179 (2008).

Li, Y. et al. A diverse family of thermostable cytochrome P450s created by recombination of stabilizing fragments. Nature Biotechnology 25, 1051–1056 (2007).

Livermore, D. M. beta-Lactamases in laboratory and clinical resistance. Clin. Microbiol. Rev. 8, 557–584 (1995).

Loksha, I. V., Maiolo, J. R., Hong, C. W., Ng, A. & Snow, C. D. SHARPEN-Systematic Hierarchical Algorithms for Rotamers and Proteins on an Extended Network. Journal of Computational Chemistry 30, 999–1005 (2009).

MacKerell, A. D. et al. All-atom empirical potential for molecular modeling and dynamics studies of proteins. J Phys Chem B 102, 3586–3616 (1998).

Murshudov, G. N. et al. REFMAC5 for the refinement of macromolecular crystal structures. Acta Crystallographica Section D Biological Crystallography 67, 355–367 (2011).

Negron, C. & Keating, A. E. in Methods in Enzymology 523, 171–190 (Elsevier, 2013).

Ng, A. H. & Snow, C. D. Polarizable protein packing. Journal of Computational Chemistry 32, 1334–1344 (2011).

Notredame, C., Higgins, D. G. & Heringa, J. T-coffee: a novel method for fast and accurate multiple sequence alignment. Journal of Molecular Biology 302, 205–217 (2000).

Otey, C. R. et al. Functional Evolution and Structural Conservation in Chimeric Cytochromes P450. Chemistry & Biology 11, 309–318 (2004).

Perrakis, A., Morris, R. & Lamzin, V. S. Automated protein model building combined with iterative structure refinement. Nat. Struct. Biol. 6, 458–463 (1999).

Ponder, J. W. et al. Current Status of the AMOEBA Polarizable Force Field. The Journal of Physical Chemistry B 114, 2549–2564 (2010).

The PyMOL Molecular Graphics System, Version 1.7.4 Schrödinger, LLC.

S. Andersen, J. Dahl, and L. Vandenberghe. CVXOPT: A Python package for convex optimization, version 1.1.6. Available at cvxopt.org, 2013.

Salomon-Ferrer, R., Case, D. A. & Walker, R. C. An overview of the Amber biomolecular simulation package. Wiley Interdisciplinary Reviews: Computational Molecular Science 3, 198–210 (2013).

Schröder, G. F., Levitt, M. & Brunger, A. T. Super-resolution biomolecular crystallography with low-resolution data. Nature 464, 1218–1222 (2010).

Stein, N. CHAINSAW : a program for mutating pdb files used as templates in molecular replacement. Journal of Applied Crystallography 41, 641–643 (2008).

Stember, J. N. & Wriggers, W. Bend-twist-stretch model for coarse elastic network simulation of biomolecular motion. The Journal of Chemical Physics 131, 074112 (2009).

Stuart, D. I. & Abrescia, N. G. A. From lows to highs: using low-resolution models to phase X-ray data. Acta Crystallographica Section D Biological Crystallography 69, 2257–2265 (2013).

Trudeau, D. L., Smith, M. A. & Arnold, F. H. Innovation by homologous recombination. Current Opinion in Chemical Biology 17, 902–909 (2013).

Vagin, A. & Teplyakov, A. Molecular replacement with MOLREP. Acta Crystallographica Section D Biological Crystallography 66, 22–25 (2010).

Voigt, C. A., Martinez, C., Wang, Z.-G., Mayo, S. L. & Arnold, F. H. Protein building blocks preserved by recombination. Nature Structural Biology (2002). doi:10.1038/nsb805

Ward, A. B., Sali, A. & Wilson, I. A. Integrative Structural Biology. Science 339, 913–915 (2013).

Winn, M. D. et al. Overview of the CCP 4 suite and current developments. Acta Crystallographica Section D Biological Crystallography 67, 235–242 (2011).

Zheng, W., Friedman, A. M. & Bailey-Kellogg, C. Algorithms for Joint Optimization of Stability and Diversity in Planning Combinatorial Libraries of Chimeric Proteins. Journal of Computational Biology 16, 1151–1168 (2009).

GLOSSARY OF TERMS

ACC Score: A method of instantiation. The normalized and inverted sum of CC scores for each residue in a protein backbone. Can range from 0-100, with a typical range of 15-50.

Block: A section of a protein backbone with several possible structural configurations called *Poses*.

Block Boundary Point: the interface between two *Blocks*, typically indicated by residue number.

Chimera: A recombined protein, composed of two or more *Parents.* The Chimeras talked about primarily in this thesis are those generated by the Arnold lab (21). Three cytochrome P450 proteins were recombined to generate the set. 21 out of the resulting 3^8 possibilities have had their structures characterized.

Combo: short for Pose Combination, this is a set of *Poses* that can be *Instantiated*

Dividing Residues: residues that are at a *Block Boundary Point*.

Energy Function: a calculation made on a protein backbone that is meant to measure some aspect. For example; the Rosetta Energy, a multifaceted measure of a protein's plausibility.

Goal structure: A structure that is being recreated by recombination. Ideally, the structure(s) produced by *Instantiation* will be close to the *Goal* structure.

Gold Standard: When the *Goal* structure is used to generate the search space, the *Combo* that corresponds to choosing the entire *Goa*l structure is called the Gold Standard. A regression that chooses the Gold Standard can correctly distinguish the best answer.

Instantiation: The scoring of a *Combo* by some means. It could be, for example, a sum of differences calculation to a specific protein, or a Rosetta Energy calculation. Regression allows for a fast approximation of *Instantiation*. Good *Combos* are low-scoring.

Parent: A protein that supplies structure to a *Chimera.*

Pose: A specific structural conformation that corresponds to a part of the protein backbone called a *Block*.

RMSD: Root mean square deviation, typically of the coordinates for protein structures. Alpha carbon coordinates are almost exclusively used in this calculation.

Source: A PDB entry that provides structural information used for the generation of *Poses*.

Divergence: The sum of coordinate differences between pairs of alpha carbons for a set of protein backbones on a per-residue basis. *Divergence* is used to choose *Block Boundary* points.

SSD Instantiation: An *Instantiation* method that is a rough measure of the differences between two structures.

Taboo Term/ Taboo Combo: A *Combo* or sub combination of any length that is not capable of being *Instantiated* correctly (for example, because of steric clashes between *Poses*) and is therefore not considered when generating new *Combos*.

Training/Test Batch: A group of previously *Instantiated Combos* used for training or testing a regression