Published in final edited form as:
Int J Data Min Bioinform. 2009 ; 3(2): 205–227.

Spherical-Harmonic Decomposition for Molecular Recognition in Electron-Density Maps

Frank P. DiMaio,
Departments of Computer Sciences and Biostatistics & Medical Informatics, University of Wisconsin, Madison, WI, USA, Email: dimaio@u.washington.edu

Ameet B. Soni,
Departments of Computer Sciences and Biostatistics & Medical Informatics, University of Wisconsin, Madison, WI, USA, E-mail: soni@cs.wisc.edu

George N. Phillips Jr., and
Departments of Biochemistry and Computer Sciences, University of Wisconsin, Madison, WI, USA, Email: phillips@biochem.wisc.edu

Jude W. Shavlik
Departments of Computer Sciences and Biostatistics & Medical Informatics, University of Wisconsin, Madison, WI, USA, E-mail: shavlik@cs.wisc.edu

Abstract

An important problem in high-throughput protein crystallography is constructing a protein model from an electron-density map. DiMaio et al. (2006) describe an automated approach to this otherwise time-consuming process. One important step involves searching the density map for many small protein fragments, or *templates*. The previous approach uses Fourier convolution to quickly compare some *rotation* of the template to the entire density map. We propose to instead use the spherical-harmonic decomposition of the template and of some region in the density map. In this new framework, we are able to eliminate areas of the map from the search process if they are unlikely to match to any templates. We design several “first-pass filters” for this elimination task, including one filter which uses a set of rotation-invariant descriptors (derived from the spherical-harmonic decomposition) of a sphere of density to train an accurate classifier. We show our new template-matching method improves accuracy and reduces running time, compared to our previous approach. Protein models constructed using this matching also show significant accuracy improvement. We extend our method to produce a structural-homology detection algorithm that, due to its use of electron-density maps, is more sensitive than sequence-only methods.

Keywords

spherical harmonics; protein-structure determination; electron-density map interpretation

1 Introduction

There has been significant research interest in high-throughput protein crystallography (Berman and Westbrook, 2004), where X-ray crystallography is used to rapidly determine a protein's three-dimensional conformation. One bottleneck in the process is producing a protein model from the electron-density map. The *electron-density map* – essentially a three-dimensional image of a protein – is produced as an intermediate result in crystallography.

Interpreting this electron-density map is the final step of X-ray crystallography. Interpretation begins with the density map and the (provided) amino-acid sequence(s) of the protein forming the crystal, and produces a complete 3D molecular model of the protein. Interpretation finds the Cartesian coordinates of every atom in the protein. In poor-quality density maps, interpretation may take several weeks of a crystallographer's time.

In DiMaio et al. (2006), we developed a method, ACMI, which automatically produces a backbone trace in poor-quality electron-density maps. A backbone trace is an important intermediate step in computing a complete (all-atom) molecular model. An important – but computationally expensive – subprocess in our previous work requires searching the density map for a set of pentapeptide (5-amino-acid) *templates*. Searching the map considers all possible 3D rotations of the template at every 3D location in the map, resulting in a 6-dimensional search problem. ACMI uses Fourier convolution (Cowtan, 1998) to quickly compute the squared-density difference between the density map and a single rotation of some template at all possible translations simultaneously.

We introduce ACMI-SH, which considers the spherical-harmonic decomposition (Kirillov, 1994) of a template's electron density and the electron density in some local region in the map. This decomposition lets us efficiently match all rotations of the template fragment at a single location. "Convolution" over rotations (as opposed to translations) allows ACMI-SH to mask – that is, to eliminate from consideration – some (x, y, z) locations in the density map. Specifically, we propose a "first-pass filter" that eliminates points that are not likely to match *any* template. At these locations, ACMI-SH assigns a low similarity score without performing a rotational search, significantly reducing the overall runtime.

We also show that a simple-filter method is effective, allowing ACMI-SH to eliminate 80% of the density map from its search without degrading performance. Using this filtering, we are able to produce improved protein models relative to a full search in less running time. Improved accuracy results from the finer angular sampling our faster approach allows, and perhaps most importantly, the substantial number of false negatives thrown out by the first-pass filter. A followup experiment shows that we can utilize a spherical-harmonic decomposition to generate a set of rotation-invariant features for use with supervised learning methods. These methods can provide further improvements in our first-pass filter.

Lastly, we extend the template-matching problem of (a) finding small-fragment matches to a density map to (b) the problem of searching for *whole*-protein matches to a density map. Our whole-protein search detects structural homologs without requiring the structure of the target protein. This search could be helpful when solving structures of new proteins, particularly when experimental phasing is challenging. We show that our extended algorithm finds several structural homologs and, while requiring a raw electron-density map, outperforms BLAST (Atschul et al., 1990) a popular sequence-only, homology-detection algorithm, at finding structurally similar proteins.

2 Automatic Density Map Interpretation

2.1 Protein Crystallography Background

Interpreting an electron-density map produces an all-atom protein model from the three-dimensional image. Figure 1 illustrates the task. In this figure, the electron-density map is illustrated as an isocontoured surface. Figure 1a shows a sample electron-density map, into which an interpreted model has been placed. Sticks indicate bonds between atoms in the interpreted model. Figure 1b presents a simplified representation of the protein, a *backbone trace*. A backbone trace represents the location of one central atom, occurring in each amino acid, the alpha carbon (or Cα).

One measure of density map quality is the map *resolution*. When placed in an X-ray beam, some protein crystals diffract better than others. In general, the larger the scattering angles of the diffracted rays, the better the resolution, resulting in more easily recognizable atomicity in the maps. Resolution is defined as the inverse of the finest spacings of the largest scattering angles, according to Bragg's law.

At excellent resolutions (2Å or better) individual atoms are visible, and automated interpretation is usually straightforward, primarily with the atom-based method ARP/wARP (Perrakis et al., 1997). However, when the resolution is worse than about 2.5Å or the map contains noise – due to data collection or experimental inaccuracy – it can take weeks of a crystallographer's time to complete a backbone trace.

2.2 Overview of ACMI

ACMI – our previous method (DiMaio et al., 2006) – produces high-confidence backbone traces from poor-quality density maps. The method is model-based, using the provided sequence of the protein to construct a model. Our previous work shows that – with poor-resolution density maps – it is able to identify amino acids more accurately then alternative approaches.

Given a protein's linear amino-acid sequence, ACMI constructs a pairwise Markov-field model (Geman and Geman, 1984). A pairwise Markov field defines some probability distribution on a graph, where vertices are associated with random variables, and edges enforce pairwise constraints on those variables. In ACMI's protein model, each vertex corresponds to an amino acid, and the random variables describe the location and orientation of each Cα. Edges enforce pairwise structural constraints on the protein.

Figure 2 shows the Markov field model associated with some protein. The probability of some backbone model $\mathbf{U} = \{u_i\}$ (where u_i is the position and orientation of the ith Cα) is given as

$$P(\mathrm{U}=\{u_i\}) \infty \prod_{\text{aminoacid } i} \psi_i(u_i) \times \prod_{\substack{\text{aminoacids } i,j \\ i \neq j}} \psi_{ij}(u_i, u_j)$$

This first product models how well an amino acid matches some location in the density map; the second models the global structural constraints on the protein.

The vertex potential ψ_i at each node i can be thought of as a “prior probability” on each alpha carbon's location, given the density map. One way to think of this is as there being an “amino-acid finder” associated with each vertex.

The edge potentials, ψ_{ij}, which enforce structural constraints on the protein, are further divided into two types: adjacency constraints ψ_{adj} model interactions between adjacent residues, while occupancy constraints ψ_{occ} model interactions between residues distant on the protein chain (though not necessarily spatially distant in the folded structure). Adjacency constraints make sure that adjacent Cα's are about 3.8Å apart; occupancy constraints make sure no two Cα's occupy the same 3D space. The graph is fully connected with edges enforcing occupancy constraints.

A fast approximate-inference algorithm finds the most likely location of each Cα, given the density map. For each amino acid in the provided protein sequence, ACMI's inference algorithm returns a probability distribution of that amino-acid's Cα location in the density map.

This paper concerns improved computation of the vertex potentials ψ_i. Accurate computation of these potentials is critical to ACMI's performance. ACMI's “amino-acid finder” considers a 5-mer (a 5-amino-acid sequence) centered at each position in the protein sequence and builds a set of small template pentapeptides (5-amino-acid structures) from a database of previously solved structures. ACMI clusters these pentapeptides into distinct groups and then searches the map against a representative example from each cluster.

Matching a template to the map uses Fourier convolution (like FFFEAR from Cowtan (1998)) to compute the *squared density difference* of one rotation of a template to the entire density map. Finally, ACMI uses a tuning set to convert squared density differences into a probability distribution over the electron-density map. Although efficient, one disadvantage of ACMI is that we are forced to search the entire density map for each template. The Fourier convolution does not allow us to search in only some locations in the map.

2.3 Other Approaches

Several methods have been developed to handle poor-quality, low-resolution density maps, where atom-based approaches like ARP/wARP fail to produce a reasonable model. In addition to ACMI, TEXTAL by Ioerger and Sacchettini (2003) and RESOLVE by Terwilliger (2003) both aim to automatically interpret maps around 3Å resolution.

TEXTAL attempts to interpret poor-resolution density maps using ideas from pattern recognition, which summarize regions of density using a set of rotation-invariant features. RESOLVE's automated model-building routine uses a hierarchical procedure in which helices and strands are located by an extensive search of all rotations and translations, then are extended iteratively using a library of known tripeptides.

At poor resolutions, both methods have difficulty correctly identifying amino acids. Our previous work shows that ACMI outperforms both TEXTAL and RESOLVE in interpreting poor-resolution maps. Additionally, both algorithms have a tendency to produce a very segmented chain in poor-resolution maps, requiring significant human labor to fix.

3 The Fast Rotation Function

We report herein a new technique for computing prior probabilities that results in improved interpretation accuracy. Our method is based on spherical-harmonic decomposition and is similar to the fast rotation function used in molecular replacement (Crowther, 1972; Trapani and Navaza, 2006), as well as for shape matching in other domains (Healy, Hendriks, and Kim, 1993; Huang et al., 2005).

Spherical harmonics $Y_l^m(\theta, \varphi)$, with order $l = 0, 1, \ldots$ and degree $m = -l, -(l-1), \ldots, l$, are the solution to Laplace's equation in spherical coordinates. They are analogous to a Fourier transform, but on the surface of sphere. They form an orthogonal basis set on the sphere's surface. Any spherical function $f(\theta, \varphi)$ can be written

$$f(\theta,\varphi)=\sum_{t=0}^{\infty}\sum_{m=-l}^{l} a_{\mathrm{lm}} \cdot Y_l^m(\theta,\varphi)$$

The key advantage of such a representation is that several different “fast rotation” algorithms exist to quickly compute the cross correlation of two functions on a sphere as a function of rotation (Trapani and Navaza, 2006; Kostelac and Rockmore, 2003). That is, given (real-valued) functions $f(\theta, \varphi)$ and $g(\theta, \varphi)$ on the sphere, we want to compute the *cross correlation* between them as a function of rotation angles $\vec{r}$,

$$C_{\mathrm{fg}}(\vec{r}) = \iint f(\theta,\varphi)\cdot \mathrm{R}(\vec{r})\cdot g(\theta,\varphi)\cdot \sin\theta d\theta d\varphi \tag{1}$$

If the functions *f* and *g* are band-limited to some maximum bandwidth *B* (or can be reasonably approximated as such), then these fast rotation functions quickly compute this cross correlation given the spherical-harmonic decomposition of *f* and *g* (running in $O(B^4)$ or $O(B^3 \log B)$ as opposed to the naive $O(B^6)$) (Kostelac and Rockmore, 2003; Risbo, 1996). A full derivation is shown by Kostelac and Rockmore (2003).

This bandwidth *B* we choose affects the fidelity with which fine details in the signal are reconstructed. In general, choosing too low of a value for *B* will lose important information in the signal, while setting *B* too high results in significant slowdown. Furthermore, eliminating some high frequency components in the signal may be desirable (for example, it may reduce noise).

4 Methods

This section describes three applications that utilize spherical-harmonic decomposition and the fast rotation function to accomplish pattern recognition tasks in the domain of electron-density interpretation. We first describe the fast template-matching method, which provides a vertex potential function in ACMI using the fast rotation function to quickly and effectively match template structures to local areas of density. Next, we describe a method that extends the fast rotation function to searching a database of solved structures for structural homologs to the target protein that created our density map. We end with a description of a map-filtering method which utilizes the fact that we no longer need to perform FFT over the entire map to eliminate areas of the map that will likely not yield template matches, thus saving computational efforts. We use properties of spherical-harmonic decomposition to create a rotation-invariant set of features that can be used in developing a classifier for eliminating these areas.

4.1 Fast Template Matching

We derive an improved vertex potential from the fast rotation function in Section 3. An overview of our local-match procedure appears in Algorithm 1, and is illustrated in Figure 3.

Algorithm 1: ACMI-SH's template matching.

```
input : amino-acid sequence Seq, density map M
output: Vertex potentials $\psi_i(y, r)$ for $i = 1\ldots N$
$(\mu_{CC}, \sigma_{CC})$ ← learn-from-tuneset()
foreach residue i do
    PDBfrags_i ← lookup-in-PDB($Seq_{i-2:i+2}$)
    foreach frag ε PDBfrags_i do
        template ← compute-dens(frag)
        templCoef ← SH-transform(template)
        foreach point $y_j$ ε M do
            if is-filtered-out($y_j$) then next $y_j$
            signal ← sample-dens-around($y_j$)
            $sigCoef \leftarrow$ SH-transform($signal$)
            CC $\leftarrow$fast-rotate($templCoef, sigCoef$)
            foreach $rotation\ r_k \ \varepsilon$ R do
                $z_k \leftarrow (\mu_{CC} - CC_k)/\sigma_{CC}$
                $p_{null} \leftarrow$ normCDF $(z_k)$
                $\psi_i(y_j, r_k) \leftarrow (1 - p_{null})/p_{null}$
            end
        end
    end
end
```

When searching for some pentapeptide, we begin by computing the density we would expect to see given the pentapeptide (one models each atom with a Gaussian sphere of density). We then interpolate this calculated density in concentric spherical shells (uniformly gridding θ – φ space) extending out to 5 or 6 Å (chosen to cover most of the density in an average pentapeptide) in 1Å steps. A fast spherical-harmonic transform computes spherical-harmonic coefficients corresponding to each spherical shell using a recursion similar to that used in fast Fourier transforms (Healy et al., 2003).

Similarly, we interpolate the density map using the same set of concentric spherical shells around some grid point, and again, take the spherical-harmonic transform of each spherical shell's density. Given these two sets of spherical-harmonic coefficients – one corresponding to the template and one corresponding to some location in the density map – a fast implementation of Equation 1 computes the *cross correlation* over all rotations of the template pentapeptide. ACMI-SH uses the implementation of Kostelac and Rockmore (2003).

After computing the cross correlation, we compute the vertex potential ψ_i as the probability that a particular cross correlation value was *not* generated by chance. That is, we assume that the distribution of the cross correlation between some template's density and some random location in the density map is normally distributed with mean μ and variance σ^2:

$$C_{\mathrm{fg}} \sim N(x;\mu,\sigma^2)$$

We estimate these parameters μ and σ^2 by computing cross correlations between the template and random locations in the map. Given some cross correlation x_c, we compute the expected probability that we would see score c$_i$ or higher by random chance,

$$p_{\mathrm{null}}(x_c)=P(X \geq x_c;\mu,\sigma^2)=1-\Phi((x_c-\mu)/\sigma)$$

Here, $\Phi(x)$ is the normal cumulative distribution function. Each amino-acid's potential is then $(1 - p_{null})/p_{null}$.

For a given template, ACMI-SH scans the density map **M**, centering the template at every location $(x_{\mathrm{i}}, y_i, z_i) \in$ **M**. At each location, we sample concentric spheres of density around (x_i, y_i, z_i), take the spherical-harmonic transform, and compute the cross correlation between the template and density map around (x_i, y_i, z_i) as a function of 3D rotation angles $\vec{r}= (\alpha, \beta, \gamma)$.

Convoluting in rotational space rather than Cartesian space (as in FFFEAR (Cowtan, 1998)) offers a several advantages. First we only have to search the *asymmetric unit* of the protein crystal – that is, only the smallest non-repeated portion of the density map – rather than the entire map. This factor alone typically accounts for a four to six-fold speedup, but depends on the symmetry of the crystal. Additionally, convoluting in rotational space allows the use of a “first-pass filter” that only considers some small portion of the density map that is likely to match templates. We perform a rotational search only for the points that pass this filter. A comparison of several such filters is presented in the Section 4.3.

There are other changes between ACMI and ACMI-SH as well. Because ACMI-SH samples spherical density shells, the template for which we are searching is a fixed-size sphere around the center of each template structure. This sphere includes many (but not all) atoms from the pentapeptide; in addition, it includes atoms from other portions of the protein located nearby. This contrasts with ACMI, where each template was arbitrarily shaped: a mask was extended to 2.5Å away from each atom in the template pentapeptide.

We feel this is advantageous as it captures the context of each pentapeptide: for example, if some 5-mer *always* occurs on the surface of a protein, all of that 5-mer's templates will be on the protein surface, and will be reflected in the cross-correlation scores. That is, a template on the surface of a protein will match best to regions of the map on the surface of the protein. Alternatively, one could use a fixed-size sphere to align a template to the map, then compute the correlation coefficient over some arbitrary-shaped region; in our experience this produces no improvement in matching accuracy, and incurs non-trivial overhead.

A final difference between ACMI and ACMI-SH is that – in ACMI – we cluster the template structures (from the PDB) to produce a *minimal subset* for which we search. ACMI-SH no longer clusters these templates. In ACMI, clustering serves mainly to reduce computational costs. Due to improved efficiency of ACMI-SH, we are able to search for a greater number of fragments than before. Even if we wanted ACMI-SH to cluster templates, we run into trouble. ACMI clusters pentapeptides using RMS deviation as a distance metric. In ACMI-SH, templates are now fixed-sized spheres, which often includes atoms not in the pentapeptide. This makes RMS deviation, which does not take these atoms into account, an ineffective measure for similarity between two templates.

Therefore, ACMI-SH simply searches for *every* template pentapeptide in the protein data bank corresponding to a particular 5-mer sequence.[1]

4.2 Comparing Maps to a Database of Structures

The fast rotational alignment presented in the previous section is not limited to small templates. The previous section's fast rotational alignment may be used to match larger protein fragments – or even entire proteins – into an electron-density map. This section describes the use of our fast rotational alignment to quickly compare a database of structures against an electron-density map. Such a tool may be useful in finding structural homologs to the target protein, even when no solved structure exists. In particular, such an algorithm may be able to detect remote homologs - proteins with similar structure but low sequence similarity. Sequence-only methods, such as BLAST fail in these cases. Having such structural homologs available may greatly aid a crystallographer in map interpretation. Finally, determining structural homologs may give key insights into a protein's function even if the density map is of too poor quality to produce an atomic model.

[1] We remove proteins in our testbed from this database before testing.

At the most abstract level, our approach considers the spherical-harmonic decomposition of a set of concentric spheres of density that cover the majority of each solved density map in a database (this database may contain experimental as well as computed density data). Assuming we know the translational correspondence between each template and the density map, we may compute similarity between the two. We use our fast rotational alignment to quickly match an entire protein to the density map, as demonstrated in Figure 4.

If a single monomer of the target protein may be masked in a density map, then finding this correspondence is straightforward: we may simply take the center of mass of both map and structure. Alternatively, an approximate center of mass may be manually located in the density by a crystallographer. The remainder of this paper assumes the density corresponding to a single monomer has been separated from the remainder of the density map. We take the center of mass of the density map in this masked region as the center of sampling. For the solved structures, we can take the center of mass for a single monomer or alternatively search for domain matches by taking multiple center of masses through *k*-means (MacQueen, 1967) clustering.

Ideally, each solved structure in a database would come with its original electron-density map. Unfortunately, this data is not widely available, so – as in the previous section – we calculate the density we would expect to see given an atomic model. As done by the Ccp4 program Sfall (CCP Project, Number 4, 1994), we model the scattering of each atom using a five-term Gaussian approximation.

We formalize the problem as follows:

> Given an electron-density map and a set of previously solved protein structures, find the solved structures that match the density map best and are thus candidate structural homologs to the target protein.

Algorithm 2 provides the details of our structure-database search procedure, which we will refer to as Shed (Structural Homology using Electron Density). Figure 4 contrasts our method with Blast (Atschul et al., 1990). Blast compares the sequence of a target protein against a database of known proteins and their sequence. When no structure is available for a target protein, sequence homology can be used to imply structural homology. Shed, on the other hand, uses the target protein's non-interpreted density map to compare against a data set of density maps from solved structures. Both return alignment scores indicating the degree of similarity between the target protein and each protein in the solved structure database.

Algorithm 2: Shed's structure database search.

input :	directory of structures **PDB**, (masked) density map M, number of centers *K*
output:	correlation coefficient CC_i between each structure in directory $i = 1\ldots |\mathbf{PDB}|$ and **M**

```
COMMap ← center-of-mass(M)
signal ← sample-sphere-around(COMmap)
sigCoef ← SH-transform(signal)
foreach structure $PDB_i$ do
    $CC_i$ ← 0
    for $k = 1\ldots K$ do
```

```
      $\mathbf{C^k} \leftarrow$ multiple-COMs($PDB_i$, $k$)
      **foreach** $COMtemplate \in \mathbf{C^k}$ **do**
        **foreach** $offset\ o \in \{-1, 0, 1\}^3$ **do**
          $template \leftarrow$
            comp-dens($PDB_i$, $COMtemplate + o$)
          $templCoef \leftarrow$ SH-transform($template$)
          $\mathbf{tempCC} \leftarrow$fast-rotate($templCoef$, $sigCoef$)
          $maxCC \leftarrow \max_{rot} \mathbf{tempCC}$
          $CC_i \leftarrow \max\{CC_i, maxCC\}$
        end
      end
    end
  end
```

4.3 Filter Template Search Space Using Rotation-Invariant Features

Our previous work performed Fourier convolutions over the entire map to efficiently match a template to a map. One disadvantage to this approach is that the entire map must be considered in every calculation. In protein structure determination, however, the number of locations containing a template match is very small compared to the size of the map - on the order of 1 Cα in 1000 grid points. Fortunately, ACMI-SH does not require this constraint since rotational alignments are done independently at each point in the map. A significant reduction in computation could be achieved if we can efficiently eliminate the areas of the map not containing templates before performing a fast rotation alignment to each template.

In Section 5.2, we compare several simple “first-pass filters” that use information from the density map to estimate the likelihood that a template is centered at some location in the map. Three of these filters are based upon the observation that in density maps, especially poor-resolution maps, Cα locations correspond to the highest-density points in the map (Leherte et al., 1997). We consider filtering points based on the point's density, as well as the average density in a 2 or 3Å radius around each point.

We also consider a filter based on the *skeletonization* of the density map (Greer, 1974). Skeletonization, similar to the medial axis transformation (Blum, 1967) in computer vision, gradually “erodes” the density map until it is a narrow ribbon approximately tracing the protein's backbone and (in high-resolution maps) sidechains. We consider filtering each point based upon its distance to the closest skeleton point. This is the first-pass filter used by CAPRA (Ioerger and Sacchettini, 2002) to eliminate points from the density map.

Finally, we consider a filter based on a set of rotation-invariant descriptors proposed by Kondor (2007), derived from spherical-harmonic decompositions. These features describe a region of density in a way that does not change as the region of density is rotated. Although these features are more time-consuming to compute than the previous filters, computation time is significantly less than that of a full rotational alignment of hundreds of templates at a point in the map.

Briefly, Kondor (2007) generalizes the bispectrum of a Fourier series to spherical harmonics. The bispectrum is a way of representing a signal in a way that is shift-invariant, yet uniquely identifies the original signal (up to translational shifts). Kondor (2007)'s representation – given a function band-limited to bandwidth B, where B is some discrete value greater than 0 – produces O(B^3) descriptors that are invariant to rotations of the original signal, yet are able to uniquely reconstruct the signal (up to rotations). This is a more powerful representation than a signal's power spectrum, in which vastly different signals may have the same power spectrum.

Using these features, we consider training a support vector machine (SVM) (Cristianini and Shawe-Taylor, 2000) to recognize whether a region will match any template in our data set. SVM is a supervised learning method that learns a set of weights α_i for each example (corresponding to some distance – or *kernel* – function $K(x_i, x_j)$); when a new example x is encountered, the weighted sum of distances $\sum_i \alpha_i K(x, x_i)$ is used to classify the example. Thresholding this sum at different values allows us to trade off the precision and recall of the classifier. In this case, the classifier learns weights to separate regions likely to contain template instances from those unlikely to contain instances.

5 Results

This section evaluates ACMI-SH using five different performance measures. The first two measures are simple tests of ACMI-SH: we first show the error introduced by band-limiting density templates, then we compare several different first-pass filters in two sets of experiments. Our third test compares ACMI to ACMI-SH, in terms of matching accuracy and running time as rotational sampling (the resolution of the $\theta - \varphi$ grid) varies. Fourth, we use both ACMI's and ACMI-SH's vertex potentials as inputs to ACMI's inference engine, and compare the resulting protein models to other approaches. The experimental setup in this fourth section is the same as we used in our previous work (DiMaio et al., 2006). The last section is an evaluation of our Shed search algorithm discussed in Section 4.2

Our data set for testing comes from a set of ten model-phased electron-density maps from the Center for Eukaryotic Genomics at the University of Wisconsin–Madison. The maps are natively all of fairly good resolution – 1.5 to 2.5Å – and all have crystallographer-determined solutions. To test algorithm performance on poor-quality (≥3.0 Å) data, we *smoothly* truncated the structure factors at 3Å and 4Å resolution, and recomputed the electron-density maps. Truncating in this fashion gives maps virtually identical to maps natively at a particular resolution.

5.1 Errors in Band-Limiting Density

Rotationally aligning two regions of density using spherical harmonics requires that we compute spherical harmonics of both the density map and the template density up to some band limit B. This band-limited signal will be somewhat different than the original signal. Figure 5 shows the average *squared density difference* between the original sampled density and the bandwidth-limited density as B is varied. The dotted line in this figure shows the squared density between two random regions, as a baseline (this measure does not depend on bandwidth limit or resolution).

This figure shows a bandwidth limit $B = 12$ accurately models the original density, with density difference $< 10^{-3}$ for 3Å resolution maps and $< 10^{-4}$ for 4Å resolution maps. The difference between two random signals is around 2.

Trapani and Navaza (2006) provides a rule of thumb for the bandwidth limit in Patterson maps (an experimental map related to the electron-density map), where the band limit B relates to the density map resolution d and the radius r by the formula $B \approx 2\pi r/d$. In this application, where we use a radius of 5Å (thus $B \approx 10$ for a 3Å map and 8 for a 4Å map), the rule produces reasonable bandwidth limits.

5.2 First-pass Filtering

A significant advantage of ACMI-SH over our previous work is that our new approach allows us to filter out regions of the map that are very unlikely to have a C*α* (the center of each template corresponds to a C*α*), without needing to perform a computationally expensive rotational alignment. This section compares five different first-pass filters, all of which are quickly computed, in two sets of experiments.

5.2.1 Simple Density Filters—As defined in Section 4.3, there are four simple filters that rely on only the map in question and an average density value for each point in that map: point density, skeletonization, average over 2Å sphere, and average over 3Å sphere. Figure 6 compares the performance of these four simple filters at both 3Å and 4Å resolution. These plots show, on the *x*-axis, the portion of the entire map we consider (sorted by our filter criteria), while the *y*-axis shows the fraction of true C*α* locations included. For example, a point at coordinates (0.2, 0.9) means a filter for which – at *some* threshold value – we look at only 20% of the density map and still find 90% of the true C*α* locations. Somewhat surprisingly, the simplest filter, the point density, performs the best at both resolutions at all thresholds. This is surprising considering the point density filter ignores features in the surrounding region. The other methods, however, are likely smoothing out distinctive features by averaging the density and thus losing important information. The experiments in Sections 5.3 and 5.4 consider using the point-density as a first-pass filter, eliminating a conservative 80% of the density map from rotational search.

5.2.2 SVM Filter Using Bispectrum Features—As a followup experiment, we consider comparing the point-density first-pass filter to a filter using a trained support vector machine (SVM) model (Cristianini and Shawe-Taylor, 2000), as motivated in Section 4.3. For each point in the density map, we extract Kondor (2007)'s real-valued numeric features from the spherical-harmonic decomposition of $R = 5$ concentric spheres of density centered at that point. We use a bandwidth of $B = 8$ and shell width of 1Å, producing a set of features of size RB^3 for each point in the grid, plus the density value of the point. If the grid point lies within a short distance of a true C*α* ($\leq \frac{\sqrt{3}}{2}$ grid units, the maximum distance from a C*α* to its closest grid point), it is labeled as positive for being a place to search for a template, otherwise it is considered negative when we evaluate ground truth. Features are normalized per map.

An SVM model, unlike the four previous simple filters, requires training to learn a decision boundary. To properly evaluate our SVM filter, we employ the commonly used 10-fold cross validation procedure. Typically, 10-fold cross validation divides our initial data set of examples into 10 subsets of examples. Of these 10 subsets, 9 are pooled together to train the SVM model – the algorithm takes these examples along with their ground truth labels and builds a model that learns how to separate the positives from the negatives. The last subset is then used for validation, or testing – the ground truth labels are held aside while the examples are given to the model to predict a label. We refer to this subset as the test-set. We can then compare the predicted labels against the ground truth labels to evaluate the accuracy of the model. This is repeated 10 times such that each subset is used exactly once for validation. In our experiment, each map constitutes one subset of examples since we have 10 maps. Since there are much fewer C*α*s than points in the grid, we have a large negative bias in the training partition. This can cause problems in training a model, so we modify our procedure by removing enough negative examples in our training partition to create an equal balance. This is neither necessary nor desired for the test-set.

Classification is done using an RBF kernel in an SVM using the *SVM^light^* (Joachim, 1999) package. To set the complexity parameter *C* as well as the kernel width γ, the training examples in each fold are divided into two sets, 80% for actual training (called our training set) and 20% for tuning (called our tuning set). The set of values considered for *C* are 1, 10, 100, and 1000 while *γ* ranges over 0.0001, 0.001, 0.01, and 0.1. The pair of values for *C* and *γ* that produce the largest area under the curve of the function described in Figures 6 and 7 for the tuning set is chosen for validation.

Figure 7a plots the averaged results for both the SVM filter and point-density filter on the 3Å maps in our data set. The SVM filter outperforms the density filter over the entire graph in 3Å maps. This result is consistent across all maps. To analyze the relative speedup of SVMs over a simple filter, we look at the fraction of the map analyzed to acquire 95% of the correct C*α* locations (0.95 on the y-axis of Figure 7). Using this metric, SVMs provide a 31% reduction in number of examples needed to be analyzed by ACMI-SH, relative to a simple point density filter alone. In fact, the difference in area under the curve and number of examples evaluated is statistically significantly better for the SVM filter with a p-value less than 0.001 according to a two-tailed t-test.

The results for 4Å maps, shown in Figure 7b, are not as convincing. While the SVM filter does better at most levels of C*α*s kept, the performance only matches that of the density filter above the 90% level. The area under the curve is slightly better for the SVM filter, but the percentage of examples evaluated at 95% C*α*s kept actually goes up slightly, although the values are statistically insignificant. It seems likely that in poorer resolution maps – where few fine details are visible – the bispectrum features provide little additional information over the density values alone. In some cases, using these additional features may even hurt performance by overfitting the data.

The maps in our data set are all well-phased density maps with poor resolution. While density proves to be a fairly consistent indicator for well-phased maps, a point-density filter does not perform as well in poorly phased maps. DiMaio et al. (2007b) describes in a detail a set of ten experimentally phased (as opposed to model-phased) density maps from the Center for Eukaryotic Genomics at the University of Wisconsin–Madison. Figure 7c shows the results of running the point-density filter and SVM filter on this data set. Both filters show decreased performance relative to the well-phased maps. The SVM filter, however, shows the same performance improvement (compared to the point-density filter) as in the 3Å data set. It reduces the number of candidate C*α*s by 31%, filtering more than 75% of the map while retaining 95% of true C*α* locations.

5.3 Template Matching

This section compares ACMI and ACMI-SH's template-matching performance. We compare the performance of both algorithms as the angular sampling of our density template is varied. Given the sequences for each of the ten proteins in our test-set, we considered searching for 10 randomly chosen amino acids in each protein (100 amino acids total). For each amino acid, we found at least 50 template pentapeptides with similar 5-mer sequences.

To test ACMI, we cluster these pentapeptides based on the RMS deviation of their optimal alignment, and select a representative structure from each cluster. Further details are in the original ACMI paper (DiMaio et al., 2006). When testing our improved implementation (ACMI-SH), we perform no such clustering. Instead, we search the entire map for each of the 50+ templates. For ACMI-SH, we filter out all points below the 80th percentile density, assigning them some low probability.

Figure 8 compares the performance of ACMI to our improved search using spherical harmonics (ACMI-SH) in both 3Å and 4Å density maps. In this plot, the *x*-axis measures the running time of the algorithm (in seconds), while the *y*-axis measures the per-amino-acid log-likelihood that matching gives the true solution. Higher likelihoods are better; the more likely the true model, the more likely its structure will be recovered by inference.

It is interesting to note here that Acmi-SH, even at its lowest bandwidth limit, offers equal or better accuracy then the previous approach, in significantly less running time.

5.4 Comparison of Protein Models Produced

In previous work, we compared the performance of Acmi on these maps to two other automated techniques specialized to low-resolution maps: Ioerger and Sacchettini (2003)'s Textal and Terwilliger (2003)'s Resolve, both described in Section 2.3.

We test Acmi-SH on these same maps, using the same experimental methodology. Figure 9 compares the accuracy of the Cα model predicted by Acmi-SH with that of Acmi, Resolve, and Textal. Figures 9a and 9b show the average Cα RMS error and percentage of amino acids located over the ten structures. Figures 9c and 9d show scatter plots in which each individually solved electron-density map is a point. The *x*-axis indicates Acmi's error (or percent amino acids correctly identified); the *y*-axis shows the same metric for Acmi-SH.

On these maps, Acmi uses $\theta = 20°$ angular discretization, while Acmi-SH was run with a bandwidth $B = 12$, and a filter that eliminated a conservative 80% of points based on the density of each point.

Here Acmi-SH shows a clear improvement over all other approaches. Both Figures 8 and 9 show the greatest improvement in 4Å-resolution maps. Even with this improved accuracy, the running time of Acmi-SH is about 60% of that of Acmi (see the middle dots in Figure 8).

The accuracy increase in using spherical harmonics likely comes from several different places. The increased efficiency allows a finer angular sampling: the bandwidth limit $B = 12$ is analogous to a 15° angular spacing. This increased efficiency also lets us search for each individual template – without clustering – which may help accuracy somewhat. Searching for a 5Å sphere, which captures the context of a particular amino acid (i.e. is an amino-acid typically on the surface or in the core of the protein?) may be improving the matching as well. Finally, band-limiting the signal, which throws out the highest-frequency components, may help eliminate noise from the density map.

5.5 Structure Database Search

To test Shed, the structure-database search algorithm from Section 4.2, we evaluate alignments of our 10 poor quality maps against a database of solved structures. Since there is not a large repository of density maps publicly available, we simulate this by generating calculated-density maps from PDB coordinates from a large set of protein chains in the Protein Data Bank as outlined in Section 4.2. We perform spherical-harmonic decomposition on a sphere centered at the center of mass of the protein chain, sampling density in 16 concentric spherical shells extending to 32Å. For the test-set density map, a similar set of spheres – extending outward from the density map's center of mass – are sampled.

As mentioned, one difficulty is that the density map may contain many molecules in the asymmetric unit. To overcome this issue, we assume that a human isolated the area of the map where one molecule exists and masks the rest of the map out. There are automated methods, such as Findmol (Mckee et al., 2005), which attempt to do this, but results on our low resolution maps were not always good. For our purposes, we manually masked the density map by keeping density values for all grid points within 7Å of a Cα in one monomer of the protein. An additional problem is that noise in the density map may skew the center of mass. To account for this, we searched a 2Å×2Å×2Å grid around each center of mass.

To account for structures with multiple domains, we optionally use *k*-means clustering (MacQueen, 1967) to choose the best set of centers in the solved structures assuming each input structure contains 1, 2, or 3 domains. Briefly, *k*-means clustering divides the density map into *k* partitions, each represented by its center of mass. Each grid point joins a partition based on which of the *k* center of masses (initialized randomly) it is closest to. The *k* centers are updated and the process is repeated until the values converge. The optimal alignment between each sampled sphere on the grid and the calculated-density map, over all center of masses and grid searches, is returned as the score between the maps.

To test the algorithm, we download a list of 6529 protein chains from the Pisces protein-sequence culling server (Wang and Dunbrack, 2003). The data set contains all PDB entries with resolution ≤3.0Å, percentage amino acid identify cutoff of 30%, and R-factor cutoff of 1.0. Each of our ten 3Å resolution maps is part of the test-set and aligned against each chain in the culled PDB data set, using the search method from Algorithm 2 and the methodology above. As a comparison, we use Blast (Atschul et al., 1990) to see if the performance of our search algorithm is better than a sequence-only method. While structural-homology detection is not Blast's original intent, sequence homology is a reasonable proxy to structural homology when structural coordinates do not exist. For ground truth, the held-aside solved PDB structure for our test maps is aligned to each query chain using DaliLite (Holm and Sander, 1996; Holm and Park, 1999), a dynamic-programming method which finds the optimal alignment between two PDB files taking into account sequence and structure.

Results are shown in Figure 10 and Table 1. For each map, we sort the alignment scores for Shed and Blast. Figure 10 displays, for each method, the average number of results in the top 5/10/25 for that method that are also ranked in DaliLite's top 5/10/25. In other words, how many of DaliLite's top 5/10/25 results are found in Shed's and Blast's top 5/10/25 results. Table 1 shows the results for each map. For example, Shed's top 5 scores for alignments on Map 1 were respectively ranked 2,3,4,6,1 in our DaliLite ground-truth calculation, meaning that 4 of Shed's top 5 results were in DaliLite's top 5.

On average, our Shed method finds more structural homologs than Blast. If you consider only the top 5 results returned by each method, Shed returns one extra correct structure on average. This advantage grows larger as more results are returned, although both methods begin to return many more false positives. Looking at the specific results in Table 1, with the exception of Map 5, our method does better than or equal to Blast, demonstrating that a density map does provide clues to the three-dimensional structure of a protein that can aid in detecting homologous structures. The bottom row in the table shows the number of “wins” each method has over the other. When 5 results are returned, Shed returned more correct structures 6 times but never returned fewer. This advantage stays relatively the same, winning 7 to 1 when 10 results are returned and 7 to 2 when 25 results are returned.

Several maps, however, did not give great results for either algorithm. While both methods consistently found the best match, maps 2, 3, 4, and 8 did not find many other matches. Looking at the DaliLite results, the alignment scores drop significantly after the top match indicating there were no more significant results to find. Another shortcoming is that the our method did not seem to find many domain, or substructure, matches. That is, most results detected only global similarity in structure. This could be addressed by isolating smaller spheres around the various center of masses. Also, the density map should also be broken into many small domains to match against possible domains in the solved structure. Our algorithm does run slower than Blast, but can perform a large database search in a few hours on a single workstation.

6 Conclusions and Future Work

We describe a significant improvement over our previous work in three-dimensional template matching in electron-density maps. Our previous work used Fourier convolution to quickly search over all (x, y, z) coordinates for some rotation of a template. Instead, we use the spherical-harmonic decomposition of a template to rapidly search all rotations of some fragment at a single (x, y, z) location.

Unlike Fourier convolution, this method allows an initial filtering algorithm to reduce computational time by "masking out" locations in the density map unlikely to contain any template instance. A simple filter allows us to eliminate 80% of density maps while maintaining most of the correct positions for templates. Spherical-harmonic decomposition generalizes to a set of rotation-invariant features, which we use in training an SVM classifier for improved filtering of density points. Our improved template matching offers both improved efficiency and accuracy, compared to previous work, finding substantially better models in about 60% of the running time.

Finally, we extend our template-matching method to handle large protein alignments to a density map. Our SHED framework demonstrates that electron-density maps can be used in a structural-homology search. In the absence of a solved structure, SHED produces more structural homologs than methods that only use protein sequences, such as BLAST. SHED can be useful in the early stages of structure determination and can provide important information from maps which prove too difficult to solve. Future work would need to address the limitations of needing to isolate a single monomer in the density map before the search is performed. One possible solution involves using our k-means procedure to isolate the center of each monomer and then varying the radius of our sphere of sampling.

An interesting future direction involves template searching and ACMI's probabilistic inference. ACMI-SH makes it possible to efficiently search for a fragment at a single location. This suggests an approach where we initial search few locations. As inference in our model proceeds, locations that appear to be promising Cα locations may emerge. We could then search at these locations, in essence using the first few iterations of our inference algorithm as a first-pass filter. This work represents a significant advance in interpretation of poor-resolution of density maps. However, to increase usability by the crystallographic community, more of an effort must be made at reducing running time.

Acknowledgments

This work is supported by NLM grant 1R01 LM008796, NLM Grant 1T15 LM007359, and NIH U54 GM074901 to the Center for Eukaryotic Structural Genomics. This article is an extended version of an earlier paper of ours on the use of spherical harmonics (DiMaio et al., 2007a).

References

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. Journal of Molecular Biology 1990;215:403–410. [PubMed: 2231712]

Berman H, Westbrook J. The impact of structural genomics on the protein data bank. American Journal of PharmacoGenomics 2004;4(4):247–52. [PubMed: 15287818]

Blum, H. A transformation for extracting new descriptions of shape. In: Wathen-Dunn, W., editor. Models for the Perception of speech and Visual Form. 1967. p. 362-380.

Collaborative Computational Project, Number 4. The CCP4 suite: Programs for protein crystallography. Acta Cryst 1994;D50:760–763.

Cowtan K. Modified phased translation functions and their application to molecular-fragment location. Acta Cryst 1998;D54:750–756.

Cristianini, N.; Shawe-Taylor, J. An Introduction to Support Vector Machines: And other kernel-based learning methods. Cambridge University Press; 2000.

Crowther, R. The Molecular Replacement Method. International Science Reviews Series 13. Gordon and Breach; 1972.

DiMaio F, Kondrashov D, Bitto E, Soni A, Bingman C, Phillips G, Shavlik J. Creating Protein Models from Electron-Density Maps using Particle-Filtering Methods. Bioinformatics 2007;23:2851–2858. [PubMed: 17933855]

DiMaio F, Shavlik J, Phillips G. A probabilistic approach to protein backbone tracing in electron-density maps. Bioinformatics 2006;22(14):e81–89. [PubMed: 16873525]

DiMaio, F.; Soni, A.; Shavlik, J.; Phillips, G. Improved Methods for Template-Matching in Electron-Density Maps Using Spherical Harmonics. Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM′07); 2007. p. 258-265.

Geman S, Geman D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Transactions on of Pattern Analysis and Machine Intelligence 1984;6:721–41.

Greer J. Three-dimensional pattern recognition. Journal of Molecular Biology 1974;82:279–301. [PubMed: 4817788]

Healy, D.; Hendriks, H.; Kim, P. Technical Report. Department of Mathematics and Computer Science, Dartmouth College; 1993. Spherical deconvolution with application to geometric quality assurance.

Healy D, Rockmore D, Kostelec P, Moore S. FFTs for the 2-sphere – improvements and variations. Journal of Fourier Analysis and Applications 2003;9:341–85.

Holm L, Park J. DaliLite workbench for protein structure comparison. Bioinformatics 1999;16:566–567. [PubMed: 10980157]

Holm L, Sander C. Mapping the protein universe. Science 1996;273:595–602. [PubMed: 8662544]

Huang, H.; Shen, L.; Zhang, R.; Makedon, F.; Hettleman, B.; Pearlman, J. Surface alignment of 3D spherical-harmonic models: Application to cardiac MRI analysis. Proceedings of MICCAI 2005; 2005. p. 67-74.

Ioerger T, Sacchettini J. Automatic modeling of protein backbones in electron density maps via prediction of C-alpha coordinates. Acta Cryst 2002;D58:2043–54.

Ioerger T, Sacchettini J. The TEXTAL system: Artificial intelligence techniques for automated protein model building. Methods in Enzymology 2003;374:244–70. [PubMed: 14696377]

Joachims, T. Making large-Scale SVM Learning Practical. In: Schlkopf, B.; Burges, C.; Smola, A., editors. Advances in Kernel Methods – Support Vector Learning. 1999.

Kirillov, A. Representation Theory and Noncommutative Harmonic Analysis Encyclopedia of Mathematical Sciences. Vol. 22. Springer; 1994.

Kondor, R. A complete set of rotationally and translationally invariant features for images. 2007. arXiv:cs/0701127v3 [cs.CV], http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0701127

Kostelec, P.; Rockmore, D. Working Paper Series. Santa Fe Institute; 2003. FFTs on the rotation group.

Leherte L, Glasgow J, Baxter K, Steeg E, Fortier S. Analysis of three-dimensional protein images. Journal of AI Research 1997;7:125–59.

MacQueen, JB. Some methods for classification and analysis of multivariate observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability; 1967. p. 281-297.

Mckee EW, Kanbi LD, Childs KL, Grosse-Kunstleve RW, Adams PD, Sacchettini JC, Ioerger TR. FINDMOL: Automated identification of macro-molecules in electron-density maps. Acta Cryst 2005;D61:1514–1520.

Perrakis A, Sixma T, Wilson K, Lamzin V. wARP: Improvement and extension of crystallographic phases. Acta Cryst 1997;D53:448–55.

Risbo T. Fourier transform summation of Legendre series and D-functions. Journal of Geodesy 1996;70:383–96.

Terwilliger T. Automated main-chain model-building by template-matching and iterative fragment extension. Acta Cryst 2003;D59:38–44.

Trapani S, Navaza J. Calculation of spherical harmonics and Wigner d functions by FFT. Acta Cryst 2006;A62

Wang G, Dunbrack RL Jr. PISCES: A protein sequence culling server. Bioinformatics 2003;19:1589–1591. [PubMed: 12912846]

Biography

Biographical notes

Frank DiMaio is a postdoctoral researcher at the University of Washington in the Department of Biochemistry. He received his PhD in Computer Sciences from the University of Wisconsin–Madison in 2007. His research interests include the application of machine learning in computational structural biology. Ameet Soni is a PhD student at the University of Wisconsin–Madison, where he received his MS in Computer Sciences in 2006. His research interests include applications of machine learning in computational biology.

George Phillips is a Professor of Biochemistry and of Computer Sciences. He received his PhD from Rice University in Biochemistry in 1976. His research interests include X-ray crystallography, protein structure-function relationships, and structural genomics. Jude Shavlik is a Professor of Computer Sciences and of Biostatistics & Medical Informatics. He received his PhD in Computer Science from the University of Illinois in 1988. His research interests include machine learning and computational biology.

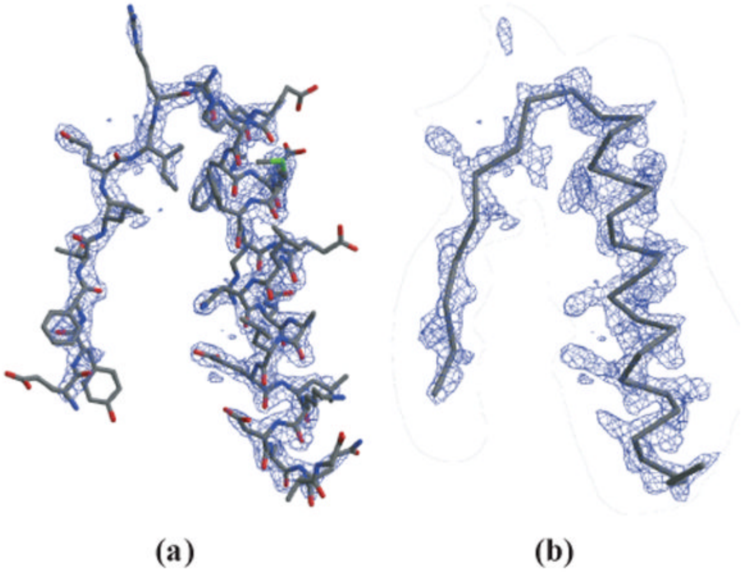


Figure 1.

An overview of density map interpretation: (a) A density map with the solved structure indicated as connected sticks, and (b) a backbone trace, where one central atom (Cα) in each amino acid is located.

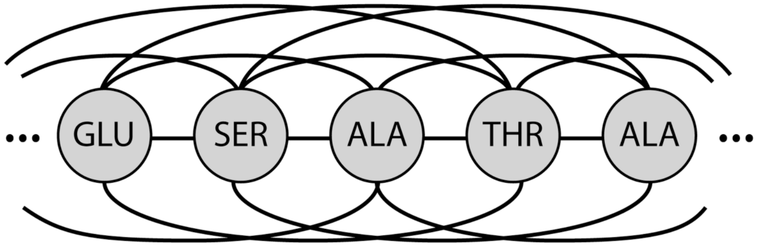


Figure 2.

The undirected graph corresponding to the protein's Markov-field model. The probability of some backbone model is proportional to the product of potential functions: one associated with each vertex, and one with each edge in the fully connected graph.

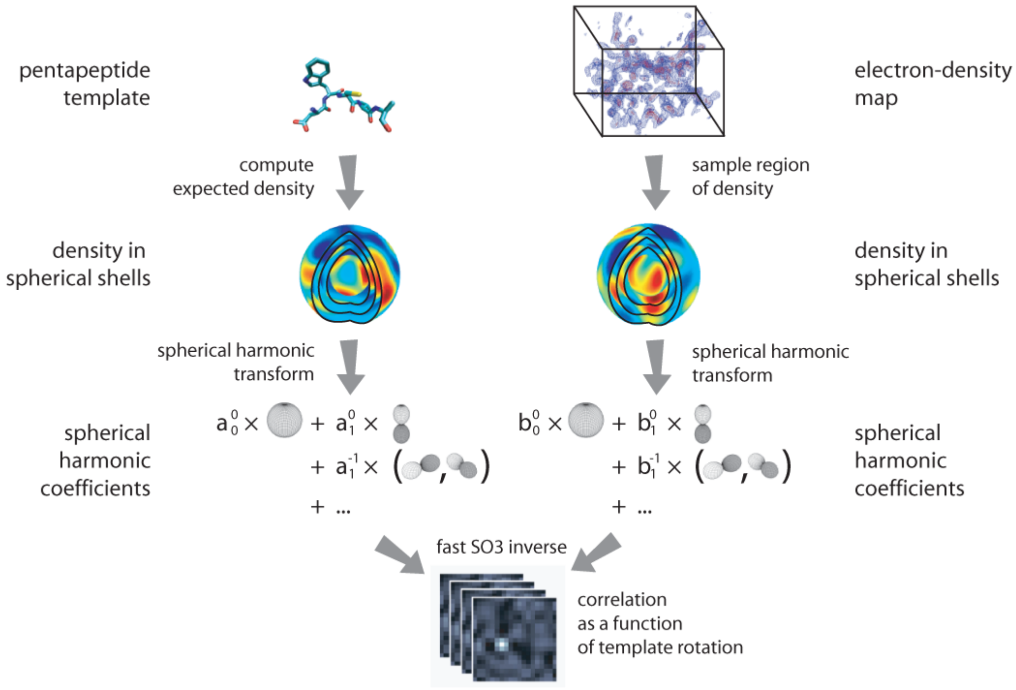
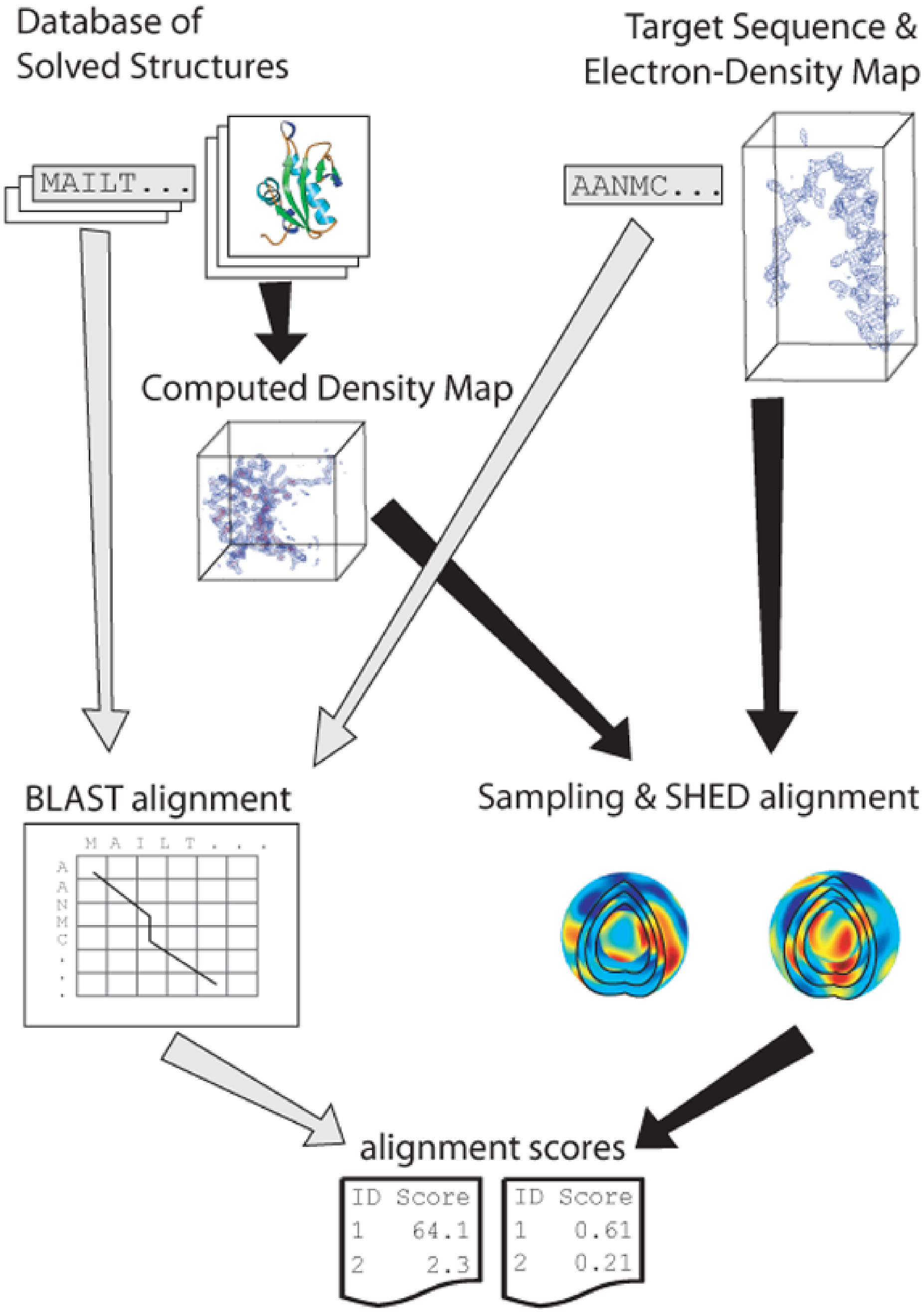

pentapeptide template
electron-density map
compute expected density
sample region of density
density in spherical shells
density in spherical shells
spherical harmonic transform
spherical harmonic transform
spherical harmonic coefficients
$a_0^0 \times$ + $a_1^0 \times$ + $a_1^{-1} \times$ (,) + ...
$b_0^0 \times$ + $b_1^0 \times$ + $b_1^{-1} \times$ (,) + ...
spherical harmonic coefficients
fast SO3 inverse
correlation as a function of template rotation

Figure 3.

ACMI's improved template-matching algorithm. Given some pentapeptide template (left) the expected electron-density is calculated. In the map (right), a spherical region is sampled. Spherical harmonic coefficients are calculated for both, and the fast rotation function computes cross correlation as a function of template rotation.



ID	Score
1	64.1
2	2.3

ID	Score
1	0.61
2	0.21

Figure 4.

A comparison of two homology-search algorithms. SHED (lower right) compares a target protein's electron-density map against a database of solved protein structures. This algorithm is similar to that in Figure 3, except we compare the whole protein structure against the density map instead of just small fragments. BLAST (lower left) considers the sequences of the solved structures and target protein, using a dynamic programming model to measure similarity between two sequences. Black arrows show the movement of density information, while grey arrows indicate the use of sequence information.

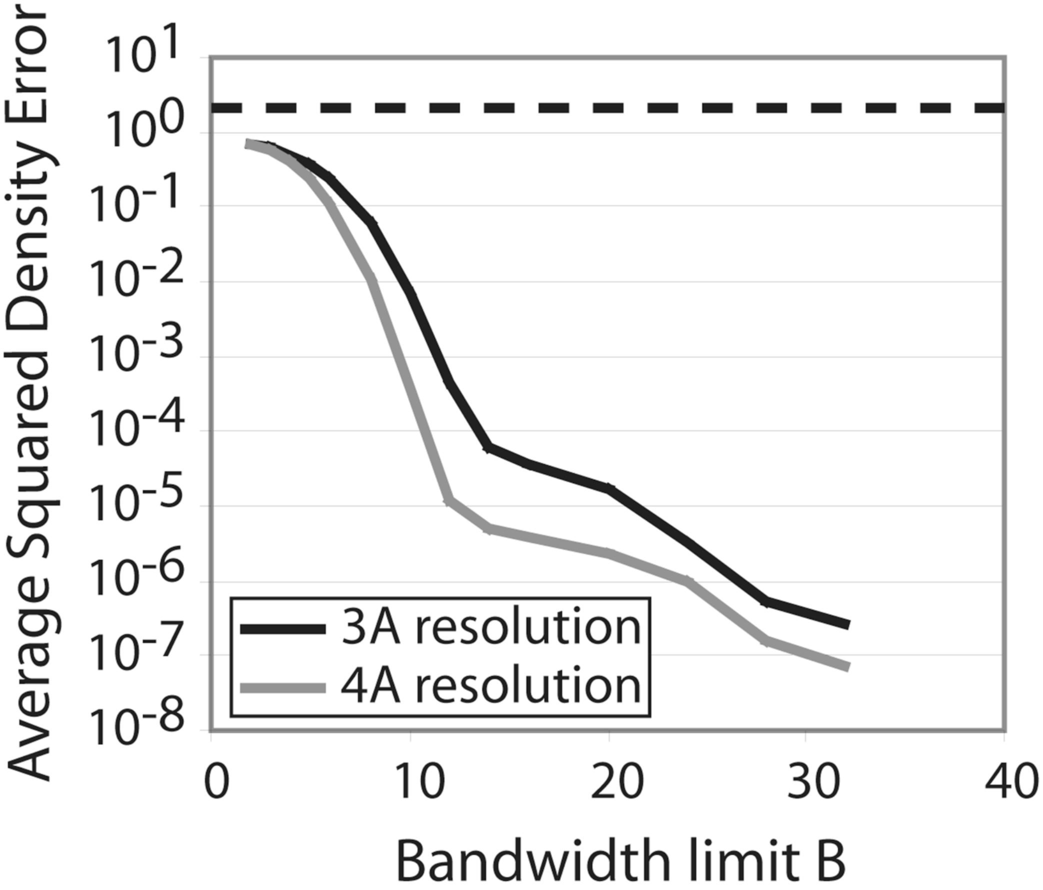


Figure 5.
The average squared density difference between a region of sampled density and the bandwidth-limited region. The dotted line shows the error between two randomly selected regions.

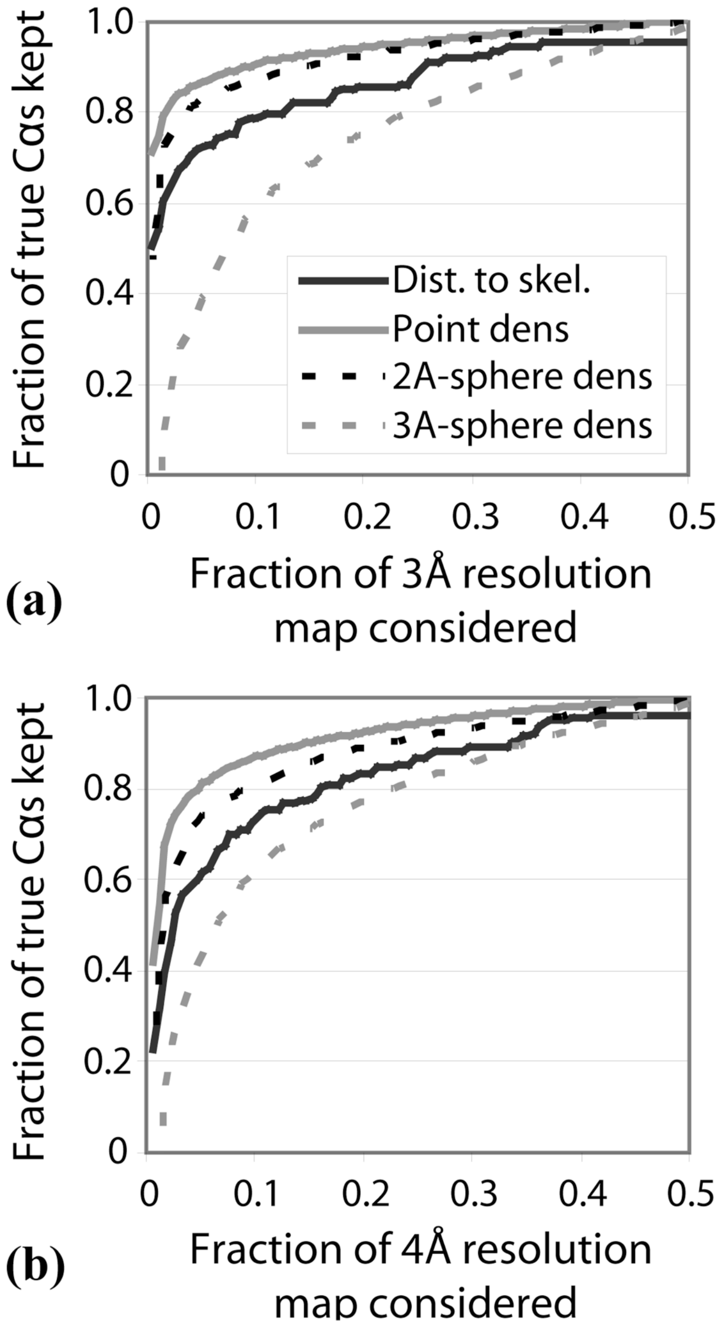


Figure 6.

A comparison of four different filters for quickly eliminating some portion of points in the density map. Filter performance is compared on (a) 3Å and (b) 4Å resolution density maps.

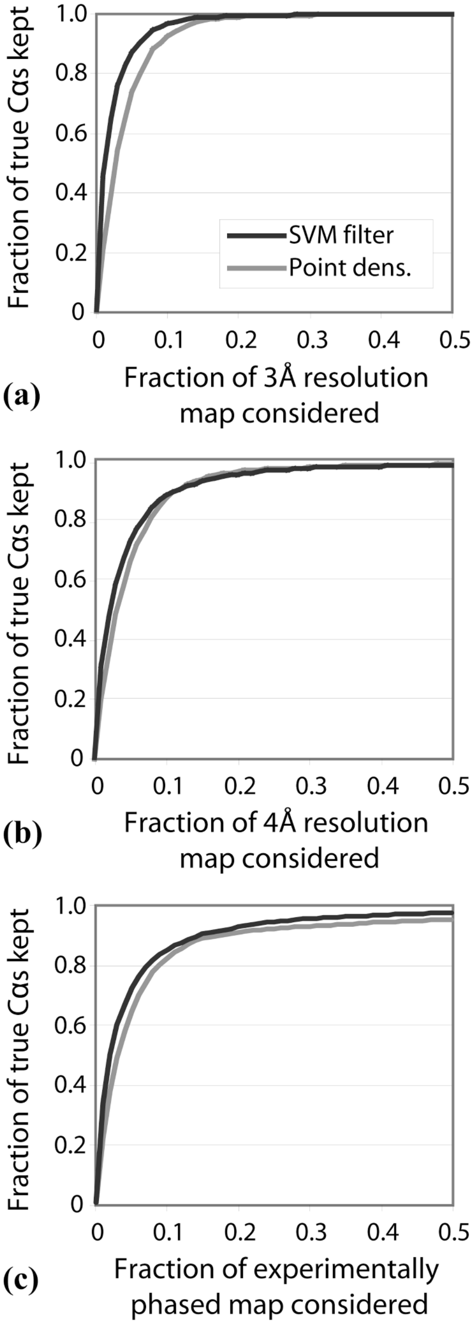


Figure 7.

A comparison of two different filters: the best simple filter from Figure 6, the point-density filter, and a filter based on a support vector machine (SVM). Filter performance is compared on (a) 3Å, (b) 4Å resolution and (c) varied resolution, experimentally phased density maps.

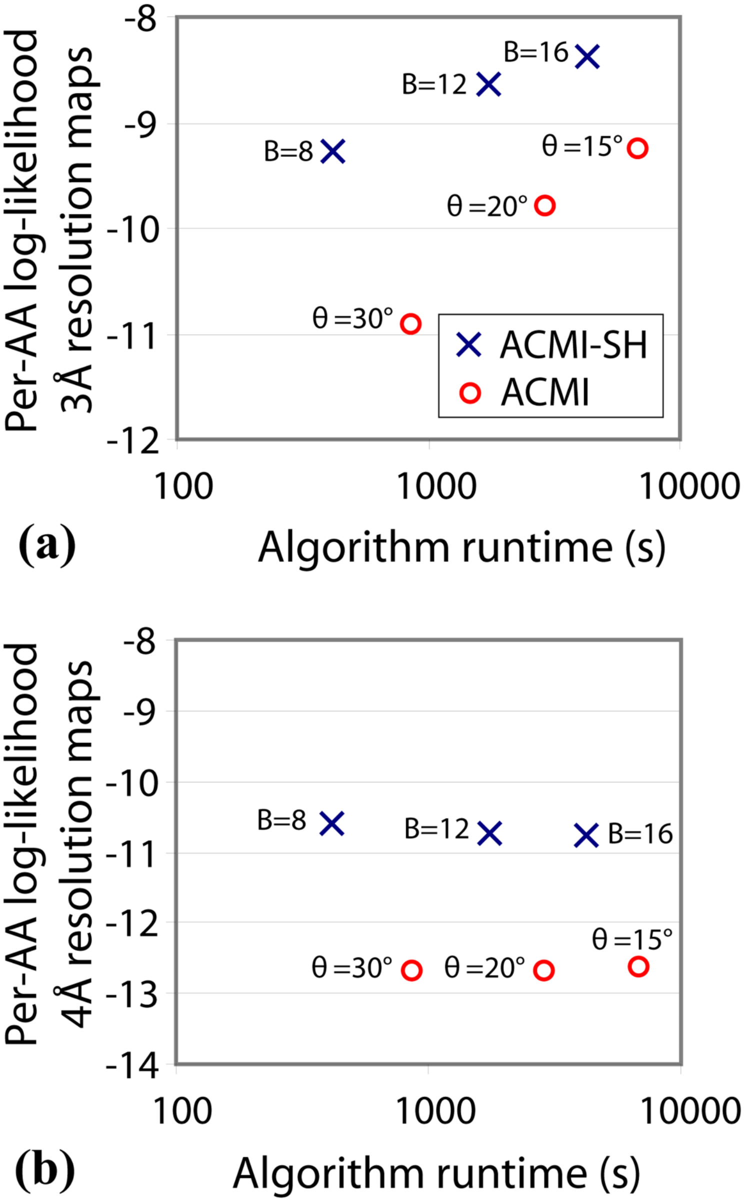


Figure 8.

A comparison of ACMI-SH's and ACMI's template matching on (a) 3Å and (b) 4Å resolution maps, in terms of average per-amino-acid log-likelihood of the true trace (higher values are better).

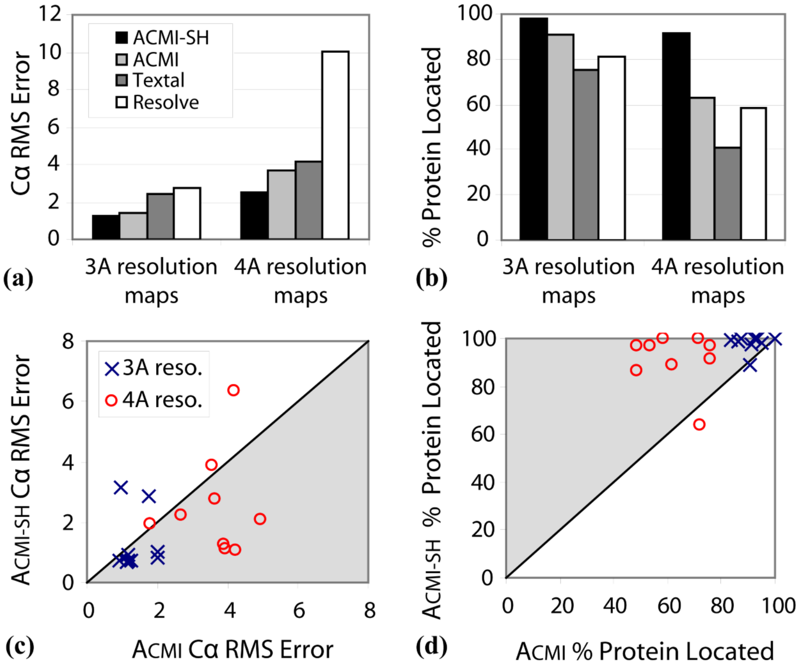


Figure 9.

Comparing ACMI-SH's protein models with three other methods. (a) The average Cα RMS error and (b) percentage of amino acids located. Scatter plots compare ACMI's performance with ACMI-SH's on (c) RMS error and (d) percentage of amino acids located. For (c) and (d), the shaded region indicates superior performance by ACMI-SH.

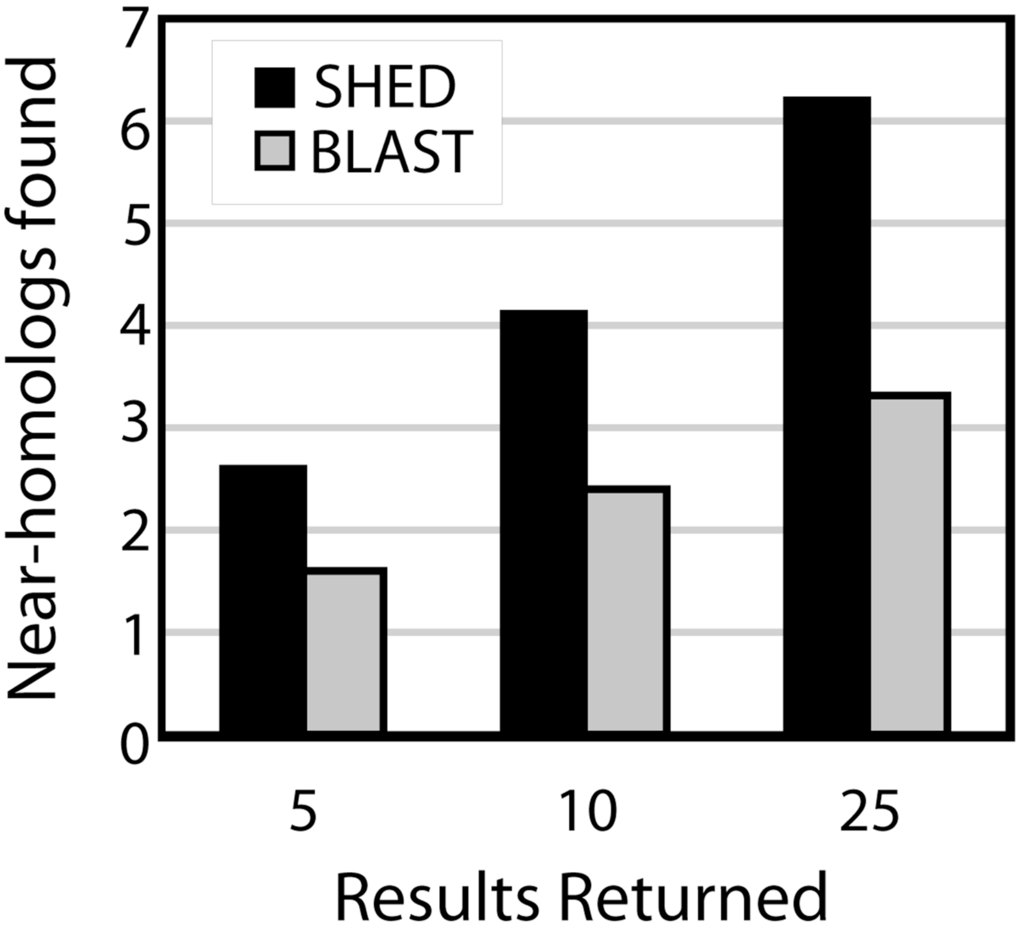


Figure 10.

The average number of structural homologs found by Shed and Blast when considering the top 5, 10, and 25 results returned. A near-homolog is a returned result that is also returned in the top 5, 10, or 25 results of DaliLite.

Table 1

Structural homology results broken down by map and number of results returned. The Wins are the number of maps over which one method outperformed the other, shown in bold.

	Top 5		Top 10		Top 25	
Map	**Shed**	**Blast**	**Shed**	**Blast**	**Shed**	**Blast**
1	4	4	**10**	9	**14**	11
2	**2**	1	**2**	1	**3**	2
3	1	1	1	1	1	1
4	**2**	1	**2**	1	**2**	1
5	3	3	4	**5**	4	**10**
6	**3**	1	3	1	3	1
7	**3**	1	**7**	1	**17**	1
8	1	1	**2**	1	**2**	1
9	**3**	2	3	3	3	**4**
10	**4**	1	**7**	1	**13**	1
Wins	**6**	**0**	**7**	**1**	**7**	**2**