APPEARS IN THE PROCEEDINGS OF THE IEEE CONFERENCE ON BIOINFORMATICS AND BIOMEDICINE (BIBM), 2007

Improved Methods for Template-Matching in Electron-Density Maps Using Spherical Harmonics

Frank DiMaio[a,b], Ameet Soni[a,b], George N. Phillips, Jr.[c,a], Jude W. Shavlik[a,b]
[a]Computer Sciences Dept., [b]Biostatistics and Medical Informatics Dept., [c]Biochemistry Dept.
University of Wisconsin – Madison
Madison, WI, 53706

Abstract

An important problem in high-throughput protein crystallography is constructing a protein model from an electron-density map. Previous work by some of this paper's authors [1] describes an automated approach to this otherwise time-consuming process. An important step in the previous method requires searching the density map for many small template protein-fragments. This previous approach uses Fourier convolution to quickly match some rotation *of the template to the entire density map. This paper proposes an alternate approach that makes use of the spherical-harmonic decomposition of the template and of some region in the density map. This new framework allows us to mask specific regions of the map, enabling a first-pass filter to eliminate a majority of the density map without requiring an expensive rotational search. 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.*

1 Introduction

There has been significant research interest in high-throughput protein crystallography [2], 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 of the protein it contains, 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.

We previously developed a method, ACMI, which automatically produces a backbone trace in poor-quality electron-density maps [1]. 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 [3] 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 [4] 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 filtering method is effective, allowing ACMI-SH to eliminate 80% of the density map from its search while only eliminating 2-3% of the true matches. Using this filtering, we are able to produce improved protein models (compared to ground truth) 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.

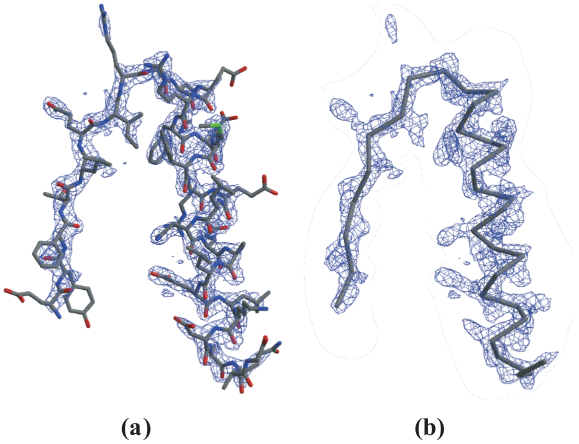

Figure 1. An overview of density map interpretation: (a) A density map with the solved structure indicated as connected sticks, and (b) a backbone trace, where one central atom (Cα) in each amino acid is located.

2 Automatic Density Map Interpretation

2.1 Protein Crystallography Background

Interpreting an electron-density map produces an all-atom protein model from this 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 the beam better than others. In general, the more the crystal diffracts the beam, the higher the *resolution*, and the better quality the map.

At excellent resolutions (2Å or lower) individual atoms are visible, and automated interpretation is usually straightforward, primarily with the atom-based method ARP/wARP [5]. 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 – 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.

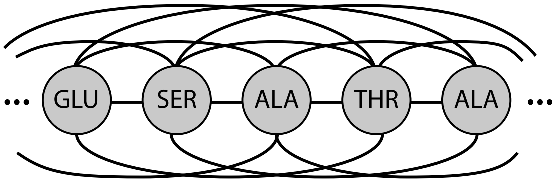

Figure 2. The undirected graph corresponding to the protein Markov field model. The probability of some backbone model is proportional to the product of potential functions: one associated with each vertex, and one with each edge in the fully-connected graph.

Given a protein's linear amino-acid sequence, ACMI constructs a pairwise Markov-field model [6]. 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

$$P(\mathbf{U} = \{u_i\}) \propto \prod_{\text{amino-acid } i} \psi_i(u_i) \times \prod_{\substack{\text{amino-acids } i,j \\ i \neq j}} \psi_{ij}(u_i, u_j)$$

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, builds a set of small template pentapeptides (5-amino-acid structures). ACMI clusters these pentapeptides, and searches the map for a single representative from each cluster.

Matching a template to the map uses Fourier convolution (like that of Cowtan's FFFEAR [3]) 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, Ioerger's TEXTAL and Terwilliger's RESOLVE both aim to automatically interpret maps around 3Å resolution.

Ioerger's TEXTAL [7] 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 [9] 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 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 Spherical Harmonics and 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 [10, 11], as well as for shape matching in other domains [12, 13].

Spherical harmonics $Y_l^m(\theta, \phi)$, 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 $f(\theta, \phi)$ can be written

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

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 [11, 14]. That is, given (real) functions $f(\theta, \phi)$ and $g(\theta, \phi)$ on the sphere, we want to compute the *cross correlation* between them as a function of rotation angles $\vec{r}$,

$$C_{fg}(\vec{r}) = \int \int f(\theta, \phi) \cdot \mathbf{R}(\vec{r}) \cdot g(\theta, \phi) \cdot sin\, \theta\, d\theta\, d\phi \quad (1)$$

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)) [14, 15]. A full derivation is shown by Kostelec and Rockmore [14].

4 A Method for Fast Template Matching

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

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 $\theta - \phi$ 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 [16].

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*

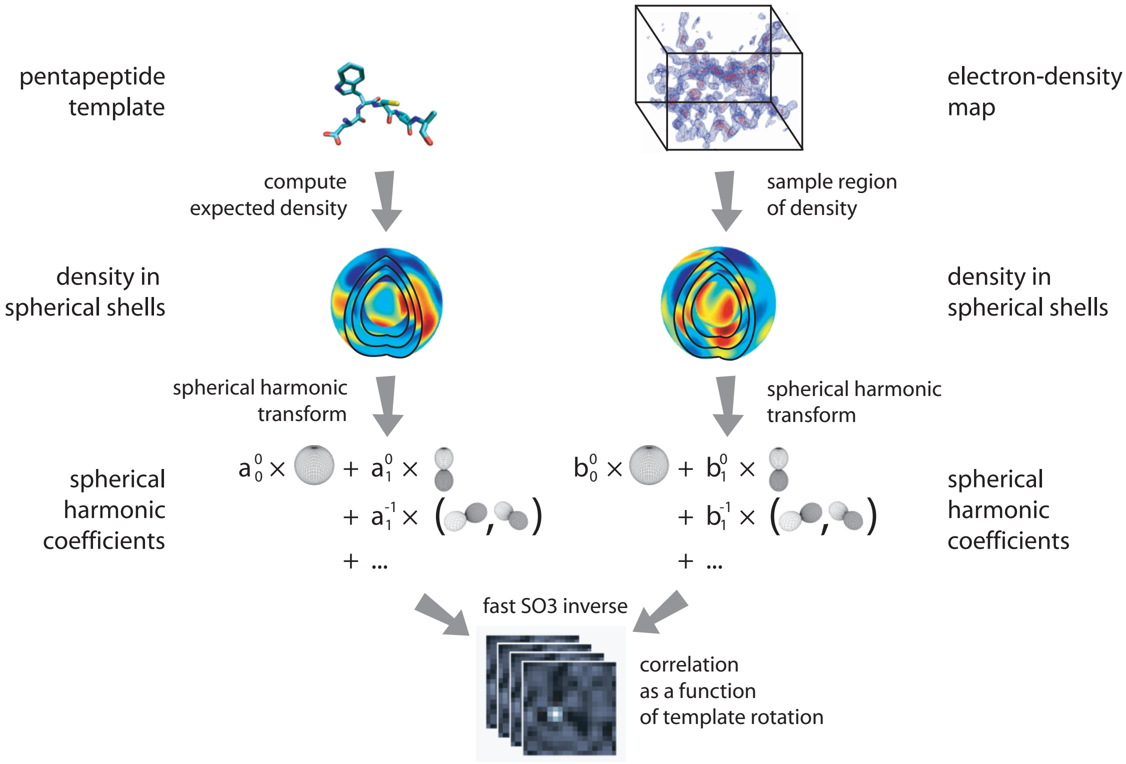
pentapeptide template
electron-density map
compute expected density
sample region of density
density in spherical shells
density in spherical shells
spherical harmonic transform
spherical harmonic transform
spherical harmonic coefficients
$a_0^0 \times$ + $a_1^0 \times$
+ $a_1^{-1} \times$ (,)
+ ...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

correlation over all rotations of the template pentapeptide. ACMI-SH uses the implementation of Rockmore and Kostelec [14].

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:

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

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,

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

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 $\mathbf{M}$, centering the template at every location $(x_i, y_i, z_i) \in \mathbf{M}$. 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 [3]) 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 alone may offers six to eight-fold runtime savings.

Additionally, convoluting in rotational space allows the use of a "first-pass filter" that only considers some small portion of the density map over which we perform a rotational search. Because ACMI searches the entire map at once using a Fourier convolution, it requires that we search over the entire density map for each fragment, even if we know a template lies in some small area. Using ACMI-SH, we instead search all of rotational space at once, allowing us to limit the (x, y, z) locations at which we search. A comparison of several such filters is presented in the next section.

There are other changes between ACMI and ACMI-SH as

Algorithm 1: ACMI-SH's template matching.

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

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 sphere, 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]

5 Results

This section evaluates ACMI-SH using four 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. Our third test compares ACMI to ACMI-SH, in terms of matching accuracy and running time as rotational sampling (the resolution of the θ–ϕ grid) varies. Finally, we use both ACMI and ACMI-SH's matching routines as input to ACMI's inference engine, and compare the resulting protein models to other approaches. The experimental setup in this final section is the same as we used in [1].

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 ($\geq$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 4 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.

Trepani and Navaza provide [11] a rule of thumb for the bandwidth limit in Patterson maps (an experimental map

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

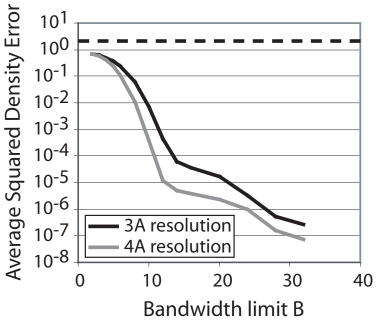

Figure 4. The average squared density difference between a region of sampled density and the bandwidth-limited region. The dotted line shows the error between two random regions.

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 Cα (the center of each template corresponds to a Cα), without needing to perform a computationally expensive rotational alignment. This section compares four different first-pass filters, all which are quickly computed.

The first three 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 [17]. The first of these filters considers filtering points based on the point's density, while the other two consider the density sum in a 2 or 3Å radius around each point.

The fourth and final filter we test is based upon the *skeletonization* of the density map [18]. Skeletonization, similar to the medial axis transformation in computer vision, gradually "erodes" the density map until it is a narrow ribbon approximately tracing the protein's backbone and sidechains. We consider filtering each point based upon its distance to the closest skeleton point. This is the first-pass filter used by CAPRA [8] to eliminate points from the density map.

Figure 5 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 Cα 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 Cα locations. Somewhat surprisingly, the simplest filter, the point density, performs the best. The remainder of our experiments consider using the point-density as a first-pass filter, eliminating a conservative 80% of the density map from rotational search.

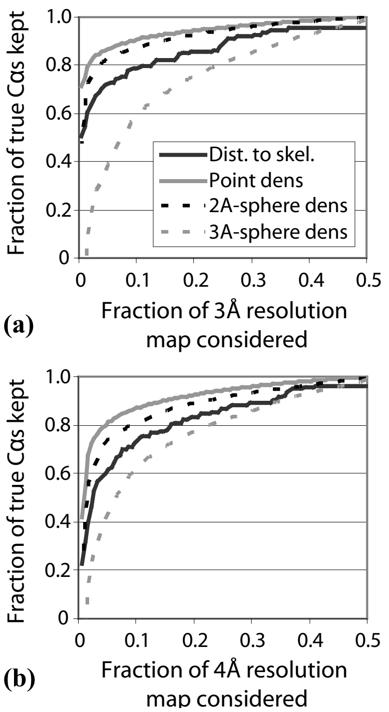

Figure 5. A comparison of four different filters for quickly eliminating some portion of points in the density map. Filter performance is compared on (a) 3Å and (b) 4Å resolution density maps.

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 testset, 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 [1]. 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.

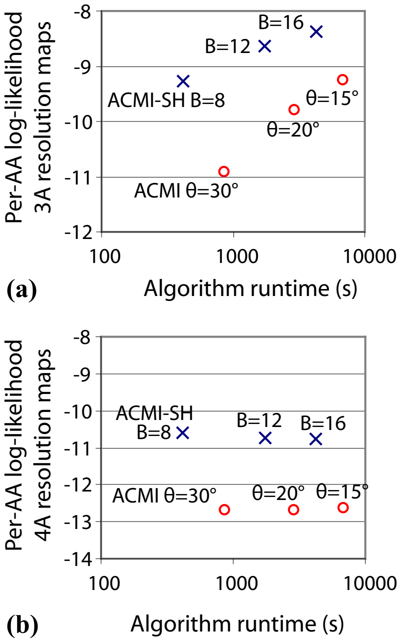

Figure 6. A comparison of ACMI-SH's and ACMI's template matching on (a) 3Å and (b) 4Å resolution maps, in terms of average per-amino-acid log-likelihood of the true trace (higher values are better).

Figure 6 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's TEXTAL and Terwilliger's RESOLVE, both described in Section 2.3.

We test ACMI-SH on these same maps, using the same experimental methodology. Figure 7 compares the accuracy of the Cα model predicted by ACMI-SH with that of ACMI, RESOLVE, and TEXTAL. Figures 7a and 7b show the average Cα RMS error and percentage of amino acids located over the ten structures. Figures 7c and 7d show scatterplots 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 6 and 7 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 6).

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.

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. 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.

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.

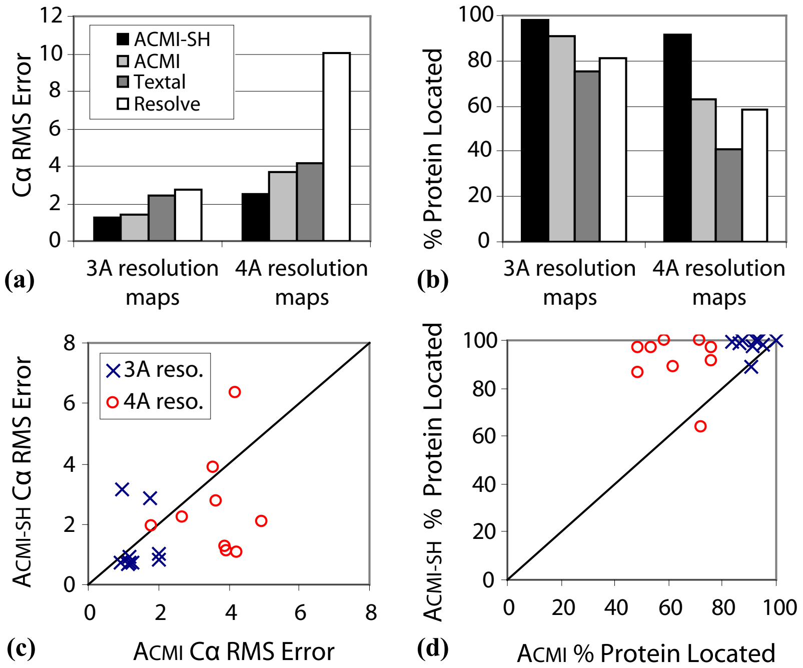

Figure 7. Comparing ACMI-SH's protein models with three other methods. (a) The average Cα RMS error and (b) percentage of amino acids located. Scatterplots compare ACMI's performance with ACMI-SH's on (c) RMS error and (d) percentage of amino acids located.

Acknowledgements

This work is supported by NLM grant 1R01 LM008796, NLM Grant 1T15 LM007359, and NIH U54 GM074901 to the Center for Eukaryotic Structural Genomics.

References

[1] F. DiMaio, J. Shavlik and G. Phillips (2006). A probabilistic approach to protein backbone tracing in electron-density maps. *Proc. ISMB*, e81-89.

[2] H. Berman and J. Westbrook (2004). The impact of structural genomics on the protein data bank. *Am. J. Pharmacogenomics* 4(4), 247-52.

[3] K. Cowtan (1998). Modified phased translation functions and their application to molecular-fragment location. *Acta Cryst.* D54, 750-756.

[4] A. Kirillov (1994). *Representation Theory and Noncommutative Harmonic Analysis. Encyclopaedia of Mathematical Sciences* 22. Springer.

[5] A. Perrakis, T. Sixma, K. Wilson and V. Lamzin (1997). wARP: Improvement and extension of crystallographic phases. *Acta Cryst.* D53, 448-55.

[6] S. Geman and D. Geman (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. of PAMI*, 721-41.

[7] T. Ioerger and J. Sacchettini (2003). The TEXTAL system: Artificial intelligence techniques for automated protein model building. *Methods in Enzymology* 374, 244-70.

[8] T. Ioerger and J. Sacchettini (2002). Automatic modeling of protein backbones in electron density maps via prediction of C-alpha coordinates. *Acta Cryst.* D58, 2043-54.

[9] T. Terwilliger (2003). Automated main-chain model-building by template-matching and iterative fragment extension. *Acta Cryst.* D59, 38-44.

[10] R. Crowther, *The Molecular Replacement Method. International Science Reviews Series* 13. Gordon and Breach.

[11] S. Trapani and J. Navaza (2006). Calculation of spherical harmonics and Wigner d functions by FFT. *Acta Cryst.* A62.

[12] D. Healy, H. Hendriks and P. Kim (1993). Spherical deconvolution with application to geometric quality assurance. *Technical Report, Department of Mathematics and Computer Science, Dartmouth College*.

[13] H. Huang, L. Shen, R. Zhang, F. Makedon, B. Hettleman and J. Pearlman (2005). Surface alignment of 3D spherical-harmonic models: Application to cardiac MRI analysis. *Proc. MICCAI* 8, 67-74.

[14] P. Kostelec and D. Rockmore (2003). FFTs on the rotation group. *Working Paper Series, Santa Fe Institute*.

[15] T. Risbo (1996) Fourier transform summation of Legendre series and D-functions. *Journal of Geodesy* 70, 383-96.

[16] D. Healy, D. Rockmore, P. Kostelec and S. Moore (2003). FFTs for the 2-sphere – improvements and variations. *J. Fourier Analysis and Applications* 9, 341-85.

[17] L. Leherte, J. Glasgow, K. Baxter, E. Steeg and S. Fortier (1997). Analysis of three-dimensional protein images. *J. AI research* 7, 125-59.

[18] J. Greer (1974). Three-dimensional pattern recognition. *J. Molecular Biology* 82, 279-301.