Pictorial Structures for Molecular Modeling

Interpreting Electron Density Maps

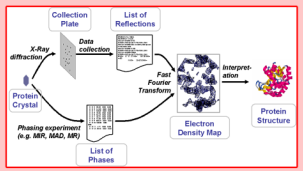
THE UNIVERSITY of WISCONSIN MADISON

Frank DiMaio
University of Wisconsin-Madison
Computer Sciences Department
dimaio@cs.wisc.edu

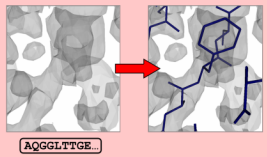
Jude Shavlik
University of Wisconsin-Madison
Computer Sciences Department
shavlik@cs.wisc.edu

George Phillips
University of Wisconsin-Madison
Biochemistry Department
phillips@biochem.wisc.edu

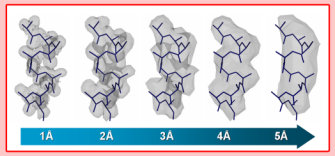
X-ray Crystallography



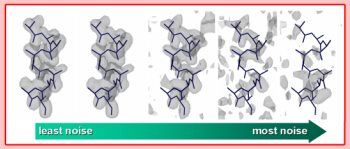
Beginning with a protein crystal, a beam of x-rays is fired through the crystal, and a pattern of spots appears on a collection plate. Data from this plate is collected, yielding a list of relections. Simultaneously, experiments on the protein crystal (or on the protein itself) give a list of phases. **Once phases and reflections are computed**, a fast Fourier transform gives an electron density map. Finally, interpreting this density map gives the final product: **a listing of the 3D coordinates of each of the protein's (non-hydrogen) atoms.**



Given the electron-density map on the left, we want to computationally find the blue lines on the right: **the arrangement of atoms that generated the observed density**. In this figure (and all density map illustrations in this poster), the density map is a three-dimensional grid of densities (analogous to a 3D discretized image). It is challenging to meaningfully convey this data format in 2D. One common method, used in this figure, represents the density map as an isocontoured surface corresponding to some density-value thresholds.



Confounding the interpretation task are several sources of error that make automated interpretation extremely difficult. The primary source of difficulty is due to the crystal only diffracting to a certain extent, eliminating higher frequency components of the density map. This produces an overall blurring effect evident in the density map. The extent to which a crystal diffracts is known as the ***resolution of the density map***. This is illustrated above, which shows a short protein's density map at a variety of resolutions (lower values of resolution mean a higher-quality density map). Given minimal noise, accurate phasing, and sufficiently good resolution – about 2.3Å or less – automated density map interpretation is essentially a solved problem [PSW97].



In addition to resolution, another issue complicating interpretation is the ***phase problem***. Phases need to be experimentally computed before an electron density map can be constructed. Initial phases are often inaccurate, leading to significant errors. As the model is built, these phases are iteratively improved [AD98], producing a better quality map, which may require resolving large portions of the map. The figure above illustrates the effect poor phasing has on

Pictorial Structures

Pictorial structures allow a single flexible template to model a variety of shapes. They represent an object as an undirected graph.

Edges model the spatial relationship between the parts they connect. Each edge has an associated cost function $\mathbf{config}_{ij}(l_i,l_j)$

Vertices in this graph model the appearance of their corresponding parts. Each vertex and has cost function $\mathbf{match}_i(l_i)$

The figure above illustrates a pictorial structure for recognizing a red car in an image. The parts ***front bumper*** and ***car body*** have an edge connecting them. This edge's **config** function is defined to enforce the constraint that the front bumper must be ***adjacent*** to the car body.

Felzenszwalb and Huttenlocher [FH00] have devised a fast matching algorithm for tree-structured pictorial structures. They treat the pictorial structure graph analogous to a **Markov random field**, placing each part independently conditioned on the position of its immediate neighbors. Their implementation uses dynamic programming, and they use a distance transform to compute the optimal position for each part in linear time. Their algorithm requires that each edges **config** cost function take the form

$$\mathbf{config}_{ij} = ||\mathbf{T}_{ij}(l_i)-\mathbf{T}_{ji}(l_j)||$$

where $||\cdot||$ is some norm (e.g. 1-norm), and $\mathbf{T}_{ij}$, $\mathbf{T}_{ji}$ are arbitrary functions.

The matching algorithm finds the **globally optimal** position and orientation of each part in the image. That is, given a pictorial structure model - the graph topology and the definition of each edge's **config** and each part's **match** function - the algorithm returns the position and orientation of each part minimizing

$$\sum_{v_i \in V} \mathbf{match}_i(l_i) + \sum_{(v_i,v_j)\in E} \mathbf{config}_{ij}(l_i,l_j)$$

How well do individual parts in the structure ***match the image***?

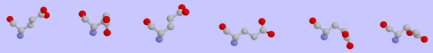
How close to ideal is the ***shape*** of parts?

Abstract

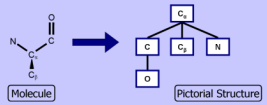
X-ray crystallography is currently the most common way protein structures are elucidated. One of the most time-consuming steps in the crystallographic process is interpretation of the electron density map, a task analogous to pattern matching in a three-dimensional picture of a protein. This paper describes DEFT (DEFormable Template), an algorithm using pictorial structures to build a flexible protein model from the protein's amino-acid sequence. Matching this pictorial structure into the density map using Felzenszwalb and Huttenlocher's fast matching algorithm is a way of automating density map interpretation. This paper describes several extensions to the pictorial structure matching algorithm necessary for this automated interpretation. DEFT is tested on a set of density maps ranging from 2 to 4Å resolution. The root-mean-squared error in DEFT's placement of protein atoms in electron-density maps ranges from 1.38 to 1.84Å.

Automating Density Map Interpretation

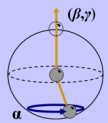
Even though the amino-acid sequence of the protein is known in advance, the large conformational space of proteins makes the use of rigid templates matching in interpreting density maps impractical. The protein is free to rotate about the majority of its bonds. A single residue alone may take dozens of residues, as illustrated below for the amino acid *glutamine*. Typical proteins may contain anywhere from about a hundred to to thousands of residues.



Because of this, we developed ***DEFT***, which models a protein using pictorial structures. In DEFT's model,
- Each **atom** (or ring) in a molecule is a part (**vertex**) in the pictorial structure graph
- Each **covalent bond** in a molecule is an **edge** in the pictorial structure graph

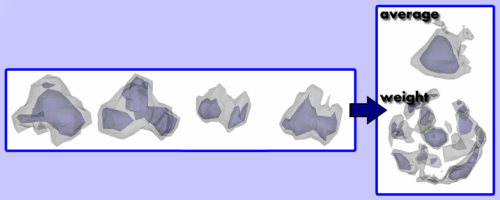


Each edge's **config** function is defined by creating a new joint type: ***the screw joint***. The screw joint allows free (no cost) rotation about the bond axis but steeply penalizes (high cost) any other rotation or translation. Ideal bond parameters are computed from a set of previously-determined structures. The figure below illustrates the screw joint: given the atom in the center, an adjacent atom can take (with no cost) any position along the blue arrow. The corresponding equation is shown on the right,



$$\begin{aligned}\mathbf{config}_{ij}(l_i,l_j) = w_{ij}^{rotate}\ & \left|(x_i-o_i)\right| \\ + w_{ij}^{orient}\ & \left(\left|(\beta_i-\beta_j)+\operatorname{atan2}\left(\sqrt{x'^2+y'^2},-z'\right)\right| + \left|(\gamma_i-\gamma_j)-\pi/2+\operatorname{atan2}(y',x')\right|\right) \\ + w_{ij}^{translate}\ & \left(\left|(x_i-x_j)-x'\right| + \left|(y_i-y_j)-y'\right| + \left|(z_i-z_j)-z'\right|\right)\end{aligned}$$

Each part's match function is defined as the similarity between the density map and a weighted template. The weighted templates were learned for each atom type from a set of crystallographer-solved structures. To learn a template for some atom type (say **leucine's C-alpha**), we take a neighborhood of density around each atom of that type in our training set. We align these examples, and - for each grid point in each example's neighborhood - compute the average and the weight (=1/std. dev.). This process is illustrated below.

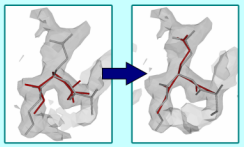


Improving the Model

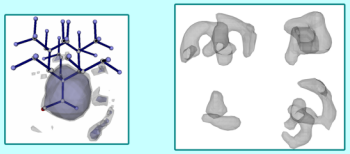
Several improvements were made to DEFT's pictorial structure matching algorithm to improve interpretation of electron density maps. The first of these had to do with **collisions** in our proposed solution. Because DEFT assumes conditional independence of each part's position, the solution our algorithm returned **often placed non-bonded atoms impossibly close together**. To handle these collisions we devised a algorithm to resolve collisions.

```
for each colliding branch Y
  x_i ← root of Y
  L_i ← optimal position of subtree rooted at x_i fixing remainder of tree
  score_i ← score(L_i) − score(subtree of L rooted at x_i)
i_min ← arg min (score_i)
L' ← replace subtree rooted at x_i in L with L_i_min
```

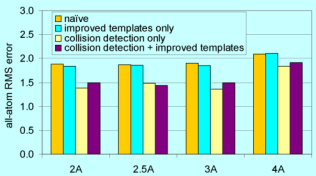
One collision in which our algorithm finds the correct structure is illustrated below (our solution is in red).

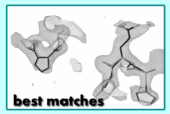


Another problem had to do with the way DEFT learned templates. **Averaging templates let to blurring out density contributions of atoms more than one bond away from the target atom.** The figure on the left shows 16 different conformations of the alpha carbon of leucine, overlayed with the learned template. We improved this approach by using k-means clustering to learn the optimal grouping of atoms. The figure on the right shows four templates (of twenty-four) learned in this fashion. Clearly these templates reflect density contributions of distant atoms.



Finally, we tested our algorithm against a set of crystallographer-solved structures. **We assumed we had an accurate backbone trace** (we knew the C-alpha locations) and used pictorial-structure matching to **lay down the sidechains**. Our training set consisted of ~1000 residues from four proteins, and the test set was ~100 residues from a protein not in the training set. The testset RMS error is illustrated in the plot below. Also plotted are a selection of the best and worst matches reported (DEFT's solution is in **black**).





References

[PSW97] A. Perrakis, T. Sixma, K. Wilson, & V. Lamzin (1997). wARP: Improvement and extension of crystallographic phases by weighted averaging of multiple refined dummy atomic models. Acta Crystallographica. D53.

[AD98] J. Abrahams, R. De Graaff (1998). New developments in phase refinement, Curr Opin Struct Biol. 8(6).

[FH00] P. Felzenszwalb & D. Huttenlocher (2000). Efficient matching of pictorial structures. IEEE Conference on Computer Vision and Pattern Recognition.

Presented at NIPS Conference, Dec. 2004