Shape modeling and matching in identifying 3D protein structures

Sasakthi Abeysinghe [a,*] Tao Ju [a] Matthew L. Baker [b]
Wah Chiu [b]

[a] *Washington University, St. Louis, USA*

[b] *Baylor College of Medicine, Houston, USA*

Abstract

In this paper, we describe a novel geometric approach in the process of recovering 3D protein structures from scalar volumes. The input to our method is a sequence of α-helices that make up a protein, and a low-resolution protein density volume where possible locations of α-helices have been detected. Our task is to identify the correspondence between the two sets of helices, which will shed light on how the protein folds in space. The central theme of our approach is to cast the correspondence problem as that of shape matching between the 3D volume and the 1D sequence. We model both shapes as attributed relational graphs, and formulate a constrained inexact graph matching problem. To compute the matching, we developed an optimal algorithm based on the A*-search with several choices of heuristic functions. As demonstrated in a suite of synthetic and authentic inputs, the shape-modeling approach is capable of identifying helix correspondences in noise-abundant volumes at high accuracy with minimal or no user intervention.

Key words: shape matching, graph matching, protein structure, electron microscopy

1 Introduction

Proteins are the fundamental building blocks of all life forms. Consisting of a linear sequence of amino acids, each protein "folds" up in space into a specific 3D shape in order to interact with other molecules. As a result, determining the 3D protein structure has critical importance in biomedical research [32]. In

[*] sasakthi.abeysinghe@wustl.edu

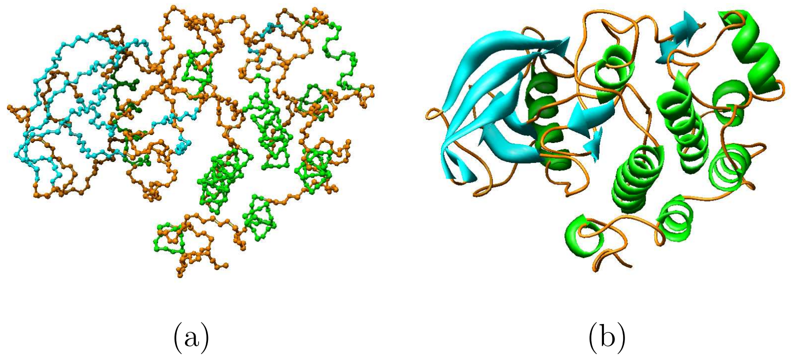


Fig. 1. The 3D protein structure of Human Insulin Receptor - Tyrosine Kinase Domain (1IRK): the folded sequence of amino acids (a) and a ribbon diagram (b) showing α-helices (green spirals) and β-sheets (blue arrows). The amino acids in these secondary structure elements are colored accordingly in (a).

an on-going project involving the co-authors, volumetric images of proteins obtained using advanced imaging techniques are utilized to decipher the protein structure. While the long-term aspiration is to determine locations of every amino-acid of the protein in such an image, we have formulated an intermediate step towards this goal as a novel shape matching task. Such formulation allows a complex feature correspondence problem in a noise-abundant environment to be solved effectively using graph matching.

1.1 Background

Proteins are large organic compounds made of amino acids arranged in a linear sequence and joined together with peptide bonds. This linear sequence folds up in space into a unique 3D structure, which largely determines the function of the protein. An example of this 3D structure is shown in Figure 1 (a). While appearing complex, this structure can be represented by high-level building blocks known as *secondary structure elements*, which are groups of continuous segments in the protein sequence stabilized by hydrogen bonds. Common secondary structure elements include α-helices and β-sheets (referred to as helices and sheets hereafter). These elements are illustrated in Figure 1 (b), and the amino acids in these elements are colored accordingly in (a).

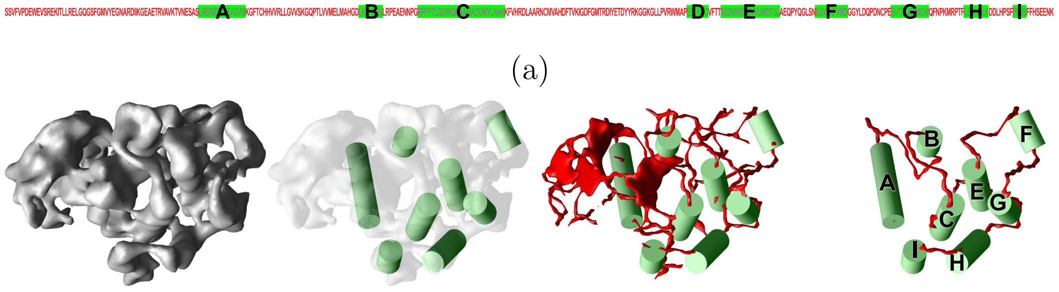
To experimentally determine the 3D structure of a protein, traditional protein imaging either uses X-Ray crystallography or NMR spectroscopy. However, both methods are limited to determining 3D structures mostly of isolated, small protein complexes. To overcome these difficulties, computational approaches such as Homology modeling and Ab-initio modeling were introduced in the past. *Homology modeling* is based on the assumption that proteins which have a reasonably similar sequence, will in turn have a similar structure. Sequence alignment is performed to determine the relationship between the template sequences (of which the structure is already known) and the target protein sequence. Thereafter, the structure of the target is derived by combining the structures of the templates [38,36]. Note that this approach is limited due to the inherent problems that occur while performing the sequence alignment step [33,50], and is severely limited by the availability of the templates [44]. *Ab-Initio modeling* on the other hand attempts to build up the 3D structure by considering the physical interactions between the atoms that form the protein [39,26,31]. This technique requires an immense amount of computational resources as it is a search for the global free energy minimum state in a significantly large search space. Therefore, this method is restricted to relatively small proteins (less than 150 amino acid residues), and is not regarded to be as accurate as homology modeling.

Recently, electron cryomicroscopy (cryoEM) emerged as a promising alternative for imaging proteins within large complexes such as viruses [9]. A typical CryoEM experiment produces a large collection of micrographs showing the projection of a single virus particle in many random directions. These are then put together into a single 3D density volume using a process known as *single particle reconstruction* [30]. Within this volume, higher density regions indicate a higher probability of the presence of atoms, and the shape of the protein can be conveniently visualized by extracting iso-surfaces at appropriate density levels. Figure 2 (b) shows the iso-surface of a simulated cryoEM density volume of 1IRK. For ease of discussion, we will refer to cryoEM density volume as simply *volume* or *density volume* hereafter.

1.2 Problem statement

The ultimate biological goal of our project is to find, in the density volume, the locations of atoms for each of the amino acids that make up the protein. Unfortunately, unlike X-ray crystallography and NMR spectroscopy, the resolution of cryoEM reconstructions is often far from sufficient to directly obtain an accurate atomic model of the imaged protein (for example, compare Figure 2 (b) with Figure 1 (a)). Instead, we first consider an intermediate step towards this goal; which is the locating of secondary structures, helices in particular, in the density volume.

Progress has already been made in the biology community for detecting positions, orientations and lengths of possible helices in a density volume [21,1] based on their cylindrical density distributions. An example is shown in Figure 2 (c) where identified helices are drawn as cylinders. Additionally, reliable methods exist for determining the location of helix segments in the amino-acid sequence [2]. An example is shown in Figure 2 (a). What is missing however, is the knowledge of which helix detected in the volume corresponds to a given



(a)

(b) (c) (d) (e)

Fig. 2. Helix correspondence as shape matching: the inputs are the 1D amino-acid sequence of the protein (a), where α-helices are highlighted in green, and the 3D volume obtained by cryo-EM (b), where possible locations of α-helices have been detected (c). Our method computes the correspondence between the two sets of helices (e) by matching the 1D sequence with a skeleton representation of the volume (d).

helix in the sequence.

As a result, the computational problem that we will address here is the *correspondence* between the helices in the sequence and the helices in the density volume. Such correspondence would establish a coarse 3D protein structure consisting of a chain of helices that sheds light on how the protein folds in 3D, as illustrated in Figure 2 (e). It is important to note that this correspondence may not be a bijection. Due to noise in a typical density volume, a helix detection algorithm may fail to find the locations of all the helices within that volume and may also identify false helices. For example, the number of helices detected in the volume in Figure 2 (c) is one less than that in the sequence in Figure 2 (a).

In the past, the helix correspondence problem has only been studied in the work of Wu *et al.* [47]. Wu employed an exhaustive combinatoric search to find, amongst all permutations of helices in the density volume, an ordering that best matches the protein sequence. Note that this brute-force algorithm has a factorial time complexity. According to their experiments, this method is only practical for very small inputs, taking 1.5 hours and 16 hours respectively to find the correspondence of a 3-helix and a 8-helix protein.

1.3 Shape modeling and matching

The central theme of our approach is to cast the helix correspondence problem as that of *shape matching* between the 1D sequence and the 3D volume. The key observation is that the search space of possible helix correspondences can be much reduced if the shape of the density volume and the sequence are taken into consideration. That is, the successive helices in the density volume must be connected by paths through high-density regions, and the lengths of these helices and paths must match those in the sequence.

The key that makes such a matching possible is the modeling of both the 1D and 3D shapes as graphs that encode the lengths of helices as well as their connectivity. In particular, the graph representing the density volume is obtained by computing a *skeleton* that encodes the topology of the high-density regions (Figure 2 (d)). Using the shape representations, helix correspondence reduces to a constrained error-correcting graph-matching problem, which seeks the best-matching simple paths among two graphs. Using a heuristic search algorithm, the optimal match can be found in an efficient manner.

When applied to an extensive suite of test data, our method was shown to be capable of identifying the correct helix correspondence with no or minimal user-intervention for small and medium size proteins. For example, Figure 2 (e) shows the correspondence computed by our method. Our shape-matching approach improves the efficiency of an otherwise exhaustive search [47] by several orders of magnitude, obtaining the helix correspondence of a 20-helix protein within 4 seconds. In addition, the availability of the skeleton allows us to plot a path on the skeleton that connects successive helices, suggesting a possible 3D trace of the amino acid sequence.

In summary, we see our work making the following contributions to shape modeling, matching and computational biology:

- We introduce a common shape representation for both protein sequences and density volumes as attributed relational graphs, which are suitable for structural matching.
- We formulate a constrained error-correcting matching problem between attributed graphs, which differs from previously known exact and inexact matching problems. In addition, we develop an optimal solution based on the A*-search, and we explore several novel heuristic functions for pruning the search space.
- We present a novel and efficient computational approach for solving an open problem in structural biology, which achieves orders of magnitude speedup over the best available method and makes structure identification from cryo-EM volumes much easier for medium-size proteins.

2 Previous work

2.1 Shape representation for matching

Shape representations, or *descriptors*, have been widely employed in graphics and computer vision for matching purposes. Generally, such representations can be classified into two classes. *Global* shape representations, often used in shape retrieval from a large repository of models, aim at computing a compact set of feature vectors of an entire object for fast comparison between objects [8,37,49]. We would refer interested readers to the survey [37] for descriptions and comparisons of these descriptors. Note that these global descriptors seldom provide local feature information and are thus generally unsuitable for partial matching; that is, finding a portion of an input object that matches a model object.

In contrast, *local* shape representations describe geometric features of an object (possibly at multiple scales) and are designed for partial matching and object alignment. Some examples of local descriptors include SIFT features [24], local spherical harmonics [16], salient surface features [17], curvature maps [18], and skeletons [40]. In this paper, we utilize the skeleton descriptor to translate the shape of an iso-surface in the density volume into a graph structure that can be used to identify connectivity among helices. Such a skeleton can be efficiently generated from a discrete volume by iterative thinning [3,4,28,41,22].

2.2 Graph matching

In pattern recognition and machine vision, graphs have long been used to represent object models, such that object recognition reduces to graph matching. Here we only give a brief review of graph matching problems and methodologies and refer the reader to the excellent surveys [7,11] for the rich volume of matching techniques.

In general, graph matching problems can be divided into exact matching and inexact matching. Exact matching aims at identifying a correspondence between a model graph and (a part of) an input graph, which can be solved using sub-graph isomorphism [43,12] or graph monomorphism [46]. However, since real-world data is seldom perfect and noise-free, inexact or error-correcting matching is desired in a large number of applications. As in [5], error-correcting matching can be formulated as finding the bijection between two subgraphs from the model and input graph that minimizes some error function. This error typically consists of the cost of deforming the original graphs to their subgraphs and the error of matching the attributes of corresponding elements in the two subgraphs. Note that, in most applications, the topology of the optimally matching subgraphs (e.g., whether it is connected, a tree, a path, etc.) is generally unknown. Such matching is said to be *un-constrained*, since the minimization of the error function is the only goal.

The most popular algorithms for error-correcting graph matching are based on the A*-search [27]. These algorithms are optimal in the sense that they are guaranteed to find the global optimal match. However, since the graph matching problem itself is NP-complete, the actual computational cost can be prohibitive for large graphs. To this end, various types of heuristic functions have been developed to prune the A* search space [42,35,6,34,46]. Other methods such as simulated annealing [19], neural networks [15], probabilistic relaxation [10], genetic algorithms [45], and graph decomposition [25] can also be used to reduce the computational cost. Observe that all of these optimization methods are developed for un-constrained matching where the matched subgraphs can assume any topology.

3 Shape representation

To solve the helix correspondence problem as stated in Section 1.2, we first seek a common shape representation of both the 1D protein sequence and the 3D density volume that is suitable for matching. In particular, such representation should encode the lengths of each helix as well as their connectivity. Here we introduce such a representation using attributed relational graphs (ARG).

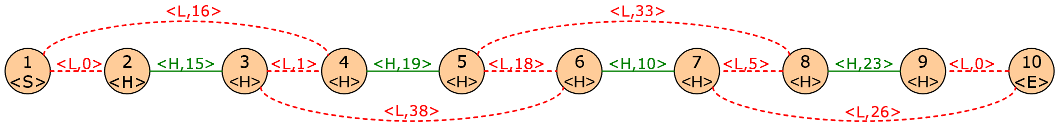
In general, an ARG G consists of a 4-tuple $< V_G, E_G, \alpha_G, \beta_G >$, where V_G is a non-empty set of nodes ($|V_G|$ denotes the number of nodes), $E_G \subseteq V_G \times V_G$ is a set of edges between pairs of nodes, and α_G, β_G are attribute functions respectively on nodes and edges. Below we detail the meaning of these graph components when describing a protein sequence or a density volume, and conclude this section with a brief summary. Note that the graphs are specifically designed to tolerate the low-resolution and noise in a density volume.

3.1 Protein sequence graph

To represent helices in the sequence, the protein sequence graph S consists of a collection of node-pairs, each denoting the two ends of a helix. These nodes are augmented by two additional terminal nodes denoting the two ends of the protein. To reflect the linearity of the sequence, we index the nodes in V_S in ascending order $\{1, \ldots, 2r + 2\}$ where r is the total number of helices, 1 and

FDVSDFLTTFLSQLRACGQYEIFSDAMDQLTNSLITNYMDPPAIPAGLAFTSPWFRFSERARTILALQNVDLNIRKLIVRHLWVITSLIAVFGRYYRPN

(a)



(b)

Fig. 3. Protein sequence graph: the amino acid sequence of a portion of the Rice Dwarf Virus (1UF2) (a), and the corresponding attributed relational graph (b).

$2r+2$ are the two terminals of the protein, and $2k$ and $2k+1$ are the two ends of the kth helix in the protein sequence. For matching purposes, the different types of nodes are also distinguished by their attributes: $\alpha_S(x)$ for each $x \in V_S$ assumes H, S or E if x represents an end of a helix, the start or the end of the protein. An example of nodes and attributes is shown in Figure 3 (b) for the sequence in (a).

To encode the lengths of helices and their connectivity, a *helix edge* is formed between every two successive nodes $2k$ and $2k+1$ for $k \in [1, r]$, and a *link edge* is formed between nodes $2k-1$ and $2k$ for $k \in [1, r+1]$, as shown in Figure 3 (b). Note that these edges form a simple path with alternating edge types. The attribute function $\beta_S(x, y)$ for each edge $\{x, y\}$ returns a 2-tuple: $\beta_{S,1}(x, y)$ indicates the edge type, being H or L when $\{x, y\}$ is a helix edge or link edge, and $\beta_{S,2}(x, y)$ maintains the length of that helix or link as the number of amino acids in the sequence. Note that the graph is undirected, that is, $\beta_{S,k}(x, y) = \beta_{S,k}(y, x)$ for $k = 1, 2$.

Due to the noisiness and the low resolution of the density volume, helix detection in the volume may not be able to find all helices of that protein (as in the example of Figure 2). To be able to establish an error-correcting matching in the presence of missing helices, we augment the graph with link edges connecting nodes $\{2k-1, 2k+2l\}$ for every $k \in [1, r]$ and $l \in [1, \min(m, r-k+1)]$ where m is a user-specified maximum number of helices that are possibly missing in the volume. The attribute $\beta_{S,2}(x, y)$ for each new link edge is set to be the total number of amino acids in the sequence bypassed by the edge. Figure 3 (b) shows an example with $m = 1$. Note that after such an addition, any simple path in the graph connecting nodes with ascending indices still consists of alternating edge types, which represents an ordered subset of helices in the protein sequence.

As in the sequence graph, the volume graph C consists of two nodes for each detected helix and two terminal nodes for the entire protein. The different types of nodes are distinguished using the node attribute function α_C, which assumes H, S or E for the helix nodes, start node or end node of the protein. Unlike the sequence graph, where there is an explicit ordering of nodes, the indices of nodes in V_C do not imply any ordering.

To encode helix information, nodes representing the two ends of a helix are connected by a helix edge. As in the sequence graph, the edge attribute function β_C returns a 2-tuple, where $\beta_{C,1}$ assumes H or L indicating a helix or link edge, and $\beta_{C,2}$ returns the length information. For a helix edge $\{x, y\} \in E_C$, $\beta_{C,2}(x, y)$ is the Euclidean length of the detected helix in the density volume, which can be normalized by the resolution of the volume to approximate the number of amino acids in the helix [1]. An example of such edges are shown in green in Figure 4 (c) representing the helices detected in the density volume in (a).

Unlike the sequence graph, the density volume does not explicitly provide the needed connectivity among detected helices. However, as stated earlier, two helices at successive positions in the sequence are more likely to be connected in 3D through regions in the volume with high density. As a result, we seek a representation that depicts the topology of such high-density regions. To this end, we extract an iso-surface from the volume at a user-specified density level and compute a morphological *skeleton* of the solid enclosed by the iso-surface. Using a recently developed erosion-based skeletonization technique [22], such skeletons can be robustly generated even from noisy surfaces while preserving the solid topology. An example of the skeleton is shown in Figure 4 (b).

Given the skeleton, we form link edges as shown in Figure 4 (c). First, we link every two nodes in the graph that represent ends of two helices connected by a path on the skeleton. When multiple paths exist between two helix ends, the shortest is taken. Note that, due to noise present in the volume, these skeleton paths may not capture all the necessary connectivity among helices. To this end, we additionally create a link edge between ends of every two helices whose Euclidean distance is within a user-specified value ϵ. Finally, to complete the graph, a link edge is created between each terminal node and every non-terminal node. The edge attribute $\beta_{C,2}$ for the above three classes of link edges are set to the length of the skeleton path, the Euclidean distance, and zero respectively (normalized by the resolution of the volume as in [1]).

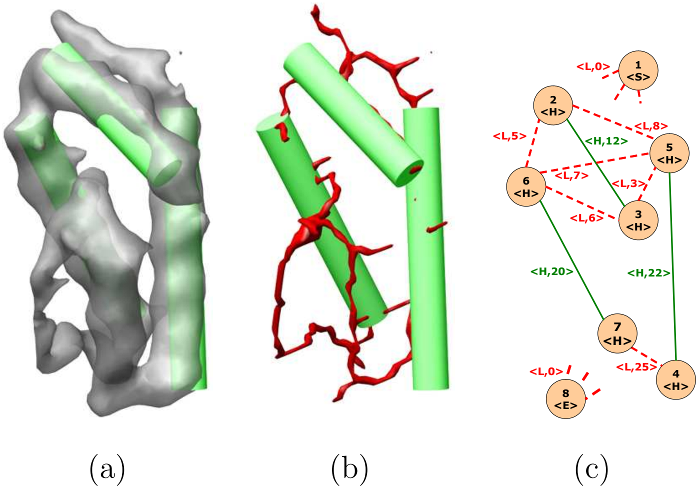


Fig. 4. Density volume graph: iso-surface of the density volume (a), the skeleton created from the iso-surface with detected helices (b), and the corresponding attributed relational graph (c) where the two terminal nodes 1,8 are connected to every other node via loop edges.

3.3 Summary

Here we briefly summarize the common meanings of the graph components $< V_G, E_G, \alpha_G, \beta_G >$ in a sequence graph ($G = S$) and in a volume graph ($G = C$).

- Nodes V_G: A helix node represents one of the two ends of a helix. A terminal node represents one of the two terminals of the protein.
- Edges E_G: A helix edge connects two ends of a helix. A link edge connects ends of two helices or between a helix end and a protein terminal.
- Node attribute $\alpha_G(x)$: Returns H, S or E if x is a helix node, the start terminal or the end terminal.
- Edge attribute
  - · $\beta_{G,1}(x, y)$: Returns H or L if $\{x, y\}$ is a helix edge or a link edge.
  - · $\beta_{G,2}(x, y)$: Returns the length of the edge $\{x, y\}$, measured as the number of amino-acids (in the sequence graph) or as the normalized length of the skeleton path (in the volume graph).

4 Constrained graph matching

Given two graphs representing the helices in the sequence and the volume, here we show that finding the correspondence between the two sets of helices reduces to a constrained graph matching problem. We first define:

Definition 1 *A* chain *of an ARG G is a sequence of nodes $\{v_1, \ldots, v_n\} \subseteq V_G$ that form a simple path in G. A chain is* ordered *if $v_1 = 1, v_n = |V_G|$, and $v_i < v_{i+1}$ for all $i \in [1, n-1]$.*

For example, an ordered chain in the sequence graph consists of edges with alternating types (e.g., helix or link), depicting a linked sequence of helices. A correspondence between helices in the sequence and the volume is therefore a bijection between an ordered chain in the sequence graph and a chain in the volume graph. Note that the definition of chain allows establishing partial correspondence between a subset of the helices in both the sequence and the volume. More generally, the problem can be defined for any attributed relational graphs:

Problem 1 *Let S, C be two ARGs. Find an ordered chain $\{p_1, \ldots, p_n\} \subseteq V_S$ and chain $\{q_1, \ldots, q_n\} \subseteq V_C$ that minimizes the matching cost:*

$$\sum_{i}^{n} c_v(p_i, q_i) + \sum_{i}^{n-1} c_e(p_i, p_{i+1}, q_i, q_{i+1}) \tag{1}$$

where c_v, c_e are any given functions evaluating the cost of matching node p_i with q_i or edge $\{p_i, p_{i+1}\}$ with $\{q_i, q_{i+1}\}$.

Comparing to previously studied graph matching problems such as exact graph (or subgraph) isomorphisms, inexact graph matching and maximum common subgraph problems [20], Problem 1 is unique in that it seeks best-matching subgraphs from two graphs that have a particular shape. Given such constraints, previous graph matching algorithms that are guided only by error-minimization can not be directly applied.

4.1 Cost functions

Here we explain our choice for the two cost functions c_v, c_e in Equation 1 when matching the sequence graph and the volume graph. Note that, the algorithm we present in the next section works for any non-negative cost function.

Each cost function measures the similarity of the attributes associated with two nodes or two edges. To enforce matching of terminal nodes in the two graphs, the node cost function is defined as

$$c_v(x, y) = \begin{cases} 0, & \text{if } \alpha_S(x) = \alpha_C(y) \\ \infty, & \text{otherwise} \end{cases} \tag{2}$$

The edge cost function computes the length difference between two helix edges or two link edges, and is defined as

$$c_e(x, y, u, v) =$$

$$\begin{cases} |\beta_{S,2}(x,y) - \beta_{C,2}(u,v)|, & \text{if } \beta_{S,1}(x,y) = \beta_{C,1}(u,v), \\ & \text{and } y = x+1. \\ |\beta_{S,2}(x,y) - \beta_{C,2}(u,v)| + \gamma_S(x,y), & \text{if } \beta_{S,1}(x,y) = \beta_{C,1}(u,v), \\ & \text{and } y > x+1. \\ \infty, & \text{otherwise} \end{cases} \tag{3}$$

Here, the γ_S term penalizes missing helices in the volume graph and is set to be the sum of lengths of the helix edges in the sequence graph bypassed by a link edge. Given a protein sequence with r helices and m possible missing helices in the density volume, and let $x = 2k - 1$ and $y = 2k + 2l$ where $k \in [1, r]$ and $l \in [1, \min(m, r - k + 1)]$, we compute

$$\gamma_S(x, y) = \omega \sum_{i=1}^{l} \beta_{S,2}(2k + 2i - 2, 2k + 2i - 1) \tag{4}$$

where ω is a user-specified weight that adjusts the influence of this penalty term.

4.2 An optimal algorithm

In this section, we present a heuristic search algorithm for solving Problem 1. Our method extends the tree-search paradigm popularized in computing unconstrained error-correcting graph matching, and is guaranteed to find the optimal match.

To find a match between two graphs, a tree-search algorithm starts out from an initial, incomplete match and incrementally builds more complete matches. To find matching chains in graphs S, C, we first consider a partial match as a sequence of node-pairs

$$M_k = \{\{p_1, q_1\}, \ldots, \{p_k, q_k\}\}$$

where $\{p_1, \ldots, p_k\}$ and $\{q_1, \ldots, q_k\}$ are the initial portion of some ordered chain in S and some chain in C. Based on the definition of chains and our matching goal of minimizing cost functions, elements of M_k must satisfy the following requirements:

- **Node requirement**: $p_1 = 1$, $q_i \neq q_j (\forall j \neq i \in [1, k])$, and for all $i \in [1, k]$:

  $$p_i \in V_S, \quad q_i \in V_C, \quad \text{and} \quad c_v(p_i, q_i) \neq \infty$$

  In words, M_k cannot contain any repeated nodes, and nodes in each pair must be of a same type.
- **Edge requirement**: For all $i \in [1, k-1]$:

  $$p_i < p_{i+1}, \quad \{p_i, p_{i+1}\} \in E_S, \quad \{q_i, q_{i+1}\} \in E_C, \\ \text{and} \quad c_e(p_i, p_{i+1}, q_i, q_{i+1}) \neq \infty$$

  In words, $\{p_1, \ldots, p_k\}$ must form an ordered chain, and the two edges connecting the two nodes in neighboring pairs in M_k must be of a same type.

Starting with an empty match $M_0 = \emptyset$, the search algorithm incrementally builds longer matching chains. Specifically, we define an *expansion* of a partial match M_k as a new partial match $M_{k+1} = M_k \cup \{\{p_{k+1}, q_{k+1}\}\}$ such that the added nodes p_{k+1}, q_{k+1} satisfy the node requirement and the added edges $\{p_k, p_{k+1}\}, \{q_k, q_{k+1}\}$ (for $k > 0$) satisfy the edge requirement. Note that usually a M_k can be expanded into multiple M_{k+1}. A match M_k is *complete* (i.e., no more expansion can be done) if $p_k = |V_S|$.

Observe that the search procedure essentially builds a tree structure with M_0 at the root of the tree, expanded partial matches M_k at the kth level of the tree, and complete matches at the tree leaves. Our goal is therefore to find the complete match that minimizes the matching error defined in Equation 1.

4.2.1 A-search*

To avoid a breadth-first tree search to find the optimal complete match, we adopt the A* search algorithm which prioritizes the expansion of incomplete matches using a fitness function. This function, $f(M_k)$, assesses the likelihood of a partial match M_k to be a part of the optimal complete match. The function has two parts:

$$f(M_k) = g(M_k) + h(M_k) \tag{5}$$

where $g(M_k)$ returns the matching cost as defined in Equation 1, and $h(M_k)$ estimates the remaining cost to be added in future expansions from M_k.

```
// Finding the optimal common chain in $S, C$

$ChainMatch(S, C)$
// $Q$ is a min heap
// The key of each element $M \in Q$ is $f(M)$
    $Q \leftarrow \{M_0\}$
    Repeat
        $M_k \leftarrow \text{Pop}(Q)$
        // $M_k$ has the form $\{\{p_1, q_1\}, \ldots, \{p_k, q_k\}\}$
        If $p_k = |V_S|$
            Return $M_k$
        Repeat for each expansion $M_{k+1}$ from $M_k$
            Insert$(Q, M_{k+1})$
```

Fig. 5. A* algorithm for Problem 1.

Given a fitness function, the A*-search algorithm works by maintaining all unexpanded partial matches in a priority queue and only expanding the partial match with the best (smallest) fitness function value. Figure 5 outlines the pseudo-code of the algorithm.

Observe from Figure 5 that the algorithm returns the first complete match that it finds. Based on A* theory, such match is guaranteed to be the *optimal* match as long as the $h(M_k)$ portion of the fitness function is a lower-bound of the actual remaining matching cost of any complete match M_n that contains M_k. That is, our algorithm works correctly if

$$h(M_k) \leq h^*(M_k) = \min_{M_n : M_k \subset M_n} (g(M_n) - g(M_k)) \tag{6}$$

where M_n are complete matches expanded from M_k, and $h^*(M_k)$ is the minimum remaining cost among all M_n.

Assuming that cost functions c_e, c_v in Equation 1 are non-negative, $h^*(M_k)$ in Equation 6 is also non-negative. Hence an obvious choice is $h(M_k) = 0$, which is a guaranteed lower-bound of $h^*(M_k)$. However, the better the approximation of $h(M_k)$ to the actual minimum remaining cost $h^*(M_k)$, the fewer nodes that have to be explored during the search. Next we present three variations of $h(M_k)$ that are all lower-bounds of $h^*(M_k)$ with different levels of tightness.

4.2.2 Heuristic fitness function

Given a partial match M_k, we denote the set of all nodes in V_S and V_C that can be added to M_k in an expansion as $R_S(M_k)$ and $R_C(M_k)$. Let $x \in R_S(M_k)$, we define:

$$h_a(M_k, x) = \min_{y \in R_C(M_k)} c_e(p_k, x, q_k, y) \tag{7}$$

and

$$h_b(M_k, x) = \sum_{y=x}^{|V_S|-1} \min_{\{u,v\} \in E_C, u \notin M_k, v \notin M_k} c'(y, u, v) \tag{8}$$

In essence, h_a computes the minimum cost of appending a pair $\{x, y\}$ into M_k for any candidate nodes y, and h_b computes the minimum cost of appending the remaining pairs to form a complete match. Here, c' is an amortized minimum cost of matching an edge $\{u, v\} \in E_C$ to any edge $\{u', v'\}$ in E_S such that $u' \leq y$ and $v' \geq y + 1$, defined as

$$c'(y, u, v) = \min_{j \in [0, y-1]} \min_{k \in [j+1, j+|V_S|-y]} \frac{c(y-j, y-j+k, u, v)}{k} \tag{9}$$

Now we define three choices of $h(M_k)$ and prove that they are all lower-bounds of $h^*(M_k)$:

$$\begin{aligned} h_0(M_k) &= 0 \\ h_1(M_k) &= \min\nolimits_{x \in R_S(M_k)} h_a(M_k, x) \\ h_2(M_k) &= \min\nolimits_{x \in R_S(M_k)} (h_a(M_k, x) + h_b(M_k, x)) \end{aligned}$$

Proposition 1 *$h_i(M_k) \leq h^*(M_k)$ for $i = 0, 1, 2$.*

Proof:

(1) Trivially we see that $h_0(M_k) = 0 \leq h^*(M_k)$

(2) Observe that h_1 computes the minimum cost of appending any pair $\{x, y\}$ into M_k, hence we have

$$\begin{aligned} h_1(M_k) &= \min\nolimits_{M_{k+1}: M_k \subset M_{k+1}} (g(M_{k+1}) - g(M_k)) \\ &\leq \min\nolimits_{M_n: M_k \subset M_n} (g(M_n) - g(M_k)) \\ &\leq h^*(M_k) \end{aligned}$$

where M_n is a complete match.

(3) We examine the minimum-cost complete match $M_n = \{\{p_1, q_1\}, \ldots, \{p_n, q_n\}\}$

such that $M_k \subset M_n$. Hence $h^*(M_k) = g(M_n) - g(M_k) = g_a + g_b$, where

$$\begin{aligned} g_a &= c_e(p_k, p_{k+1}, q_k, q_{k+1}) \\ g_b &= \textstyle\sum_{j=k+1}^{n-1} c(p_j, p_{j+1}, q_j, q_{j+1}) \end{aligned}$$

Note that $h_a(M_k, p_{k+1}) \leq g_a$. In addition, the lower-bound cost function c' ensures that

$$c'(i, q_j, q_{j+1}) \leq \frac{c_e(p_j, p_{j+1}, q_j, q_{j+1})}{p_{j+1} - p_j}$$

for any $p_j \leq i < p_{j+1}$. Hence we have

$$\begin{aligned} h_b(M_k, p_k) &\leq \textstyle\sum_{j=k+1}^{n-1} \sum_{i=p_j}^{p_{j+1}-1} \frac{c_e(p_j, p_{j+1}, q_j, q_{j+1})}{p_{j+1}-p_j} \\ &\leq \textstyle\sum_{j=k+1}^{n-1} c_e(p_j, p_{j+1}, q_j, q_{j+1}) \\ &\leq g_b \end{aligned}$$

Finally,

$$\begin{aligned} h_2(M_k) &\leq h_a(M_k, p_k) + h_b(M_k, p_k) \\ &\leq g_a + g_b \\ &\leq h^*(M_k) \end{aligned}$$

□

Based on the proposition, using either of the three functions in the fitness function 5 will result in an optimal solution in the A*-search. We can further show that the three functions h_i achieve increasingly better approximation of the actual minimum remaining cost:

Corollary 1 $0 = h_0(M_k) \leq h_1(M_k) \leq h_2(M_k) \leq h^*(M_k)$.

Proof: It is not hard to see that $h_1(M_k) \leq h_2(M_k)$ from their definitions, given that h_b is non-negative. The rest of the inequality is a direct result of Proposition 1 and the fact that both h_1 and h_2 are non-negative. □

Since a better approximation of the remaining cost helps to prune the search space, fewer nodes need to be expanded during the search using h_1 or h_2 over using h_0 in the fitness function. However, we have to note that the computation of h_1, h_2 is much more expensive than h_0. In particular, evaluating the h_b portion of h_1 or h_2 involves nested minimality queries. In our implementation, we accelerated the brute-force calculation of h_b by pre-computing a look-up table indexed by a node $y \in V_S$, which maintains a sorted list of edges $\{u, v\} \in E_C$ in the ascending order of $c'(y, u, v)$.

5 Results

In this section, we discuss the performance of our method on an extensive suite of protein data. For a significant fraction of these test data sets, we observed that our method was capable of finding the correct helix correspondences without any user intervention. However, for density volumes with poor quality, the optimal graph matching may not represent the actual helix correspondence, and domain knowledge has to be incorporated to yield the correct result.

5.1 Setup

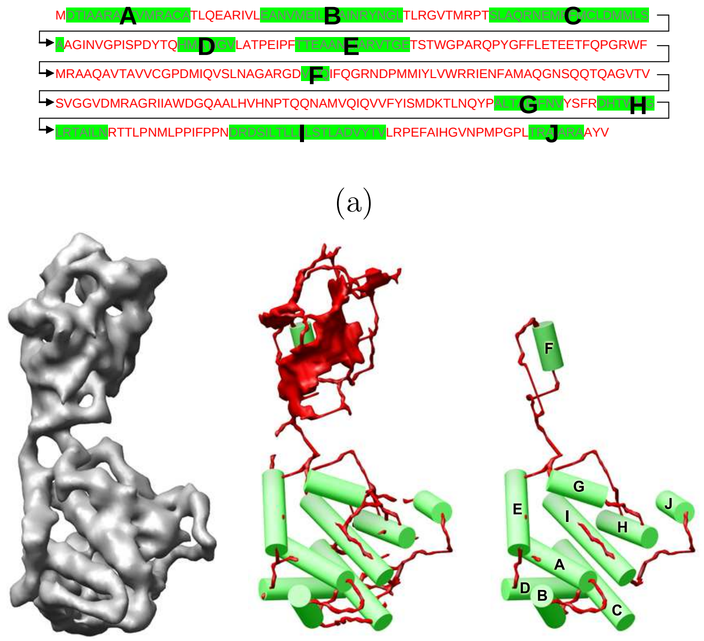
Our experiment consists of 11 cryoEM volumes at 6Å-10Å resolution, 8 of which are simulated from the actual atomic model obtained from the Protein Data Bank [14] and 3 which are authentic cryoEM reconstructions (P22 GP5, RDV P8 and a GroEL monomer [1]). These structures, while not an exhaustive representation of those found in the Protein Data Bank, do represent commonly occurring folds of the major families of protein structure. In addition, only three authentic cryoEM reconstructions are reported as there are only a small number of structures in the public domain with resolutions beyond 7Å-8Å.

In each example, we utilize the protein sequence data from the Protein Data Bank, the helices in density volumes detected using the SSEhunter software [1], and the skeleton created using the method of [22]. The matching result is presented as a correspondence between helices in the sequence with those in the density volume. In all the experiments, an Euclidean distance threshold of $\epsilon = 0.15d$ is used for creating extra edges in the volume graph where d is the size of the volume (d for the data is shown in Table 1), and $\omega = 5$ is used in weighting the missing helix penalty term in the cost function. Experiments were performed on a PC with a 3GHz Pentium D CPU and 2GB of memory (our implementation runs on a single thread, thus utilizes only one of the cores of the CPU).

5.2 Evaluation method

To evaluate the accuracy of matching, we compare the helix correspondence computed by our method with a manual labeling of the helices in the density volume based on the known atomic structure (for simulated data) or a

[1] EMDB number for these authentic reconstructions are 1060 (RDV P8), 1101 (P22) and 1081 (GroEL)



(a)

(b)

(c)

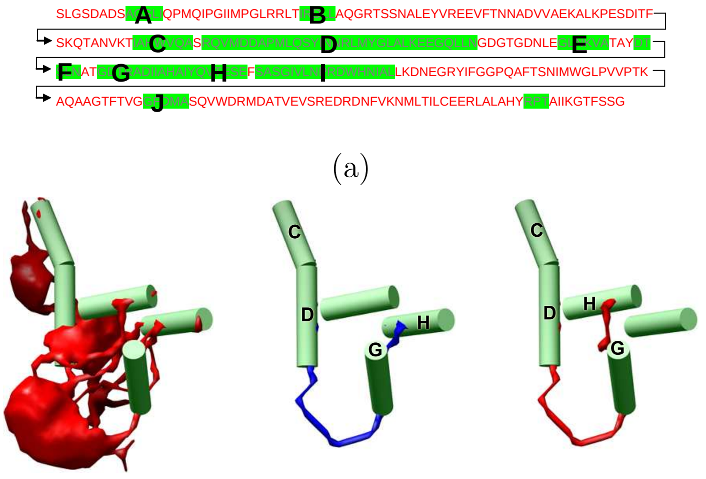
(d)

Fig. 6. Bluetongue Virus (2BTV): the amino acid sequence with detected helices (a), the density volume (b), the detected helices with the skeleton (c), and the correspondence between two sets of helices computed as the optimal match between the sequence and volume graphs (d).

structural homologue (for authentic data). To improve on a binary evaluation of whether or not our method finds the correct correspondence, we compute a list of candidate correspondences between the helices in the sequence graph and the volume graph, ranked by their matching costs. This can be done easily in the A*-search framework by terminating the search only after a number of complete matches (e.g., 100) have been found. In our experiments, the accuracy of our method is reported as the ranking of the manual-labeled correspondence in this candidate list.

5.3 Unsupervised matching

Figure 2 and 6 show two examples (1IRK and 2BTV) where our method is able to identify the correct full or partial correspondence as the top-ranked candidate. Note that our algorithm is robust to noise in the data such as the one missing helix in the density volume of 1IRK. As a by-product of our matching algorithm, a "trace" of the protein sequence in the density can be visualized by rendering the skeleton paths represented by the graph edges in the optimally matching chain. Such a trace could serve as a starting point to determine finer-scale protein components such as amino acids.



(a)

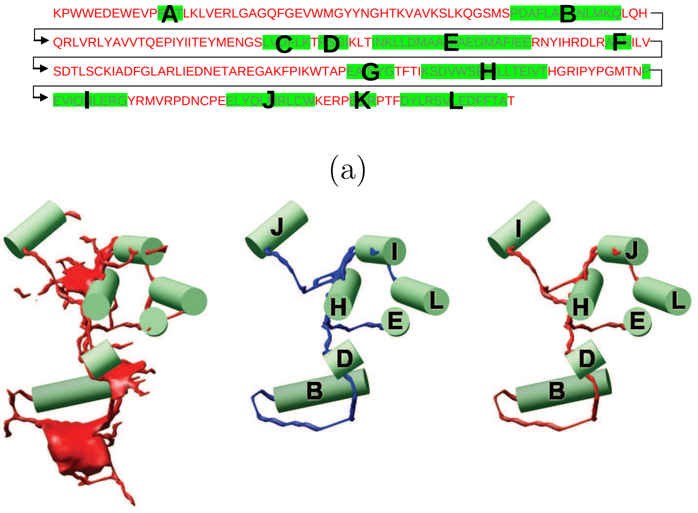
(b) (c) (d)

Fig. 7. Bacteriophage P22 capsid protein (P22 GP5): the amino acid sequence with detected helices (a), the detected helices in the density volume with the skeleton (b), the optimal match between the two sets of helices computed by our method (c), and the actual correspondence (d) which ranks 4th in the candidate matches.

Figure 7 shows an example (P22 GP5) where the correct correspondence (shown in (d)) is ranked 4th in the candidates list. Comparing with the top-ranked correspondence (shown in (c)), the two correspondences exhibit very similar helix lengths and connectivity, illustrating why graph matching alone can not distinguish the right from the wrong without further domain knowledge. A similar observation can be made in another example in Figure 8 for protein 3LCK, where the correct correspondence ranks 2nd in the candidate matches. Note that in both examples, the correct correspondence is found even in the presence of a severe amount of noise (see more discussion in Section 5.6), with the density volume missing 5 helices for protein 3LCK and 6 helices for protein P22.

5.4 Interactive matching

In the case where the resolution of the density volume does not provide sufficient shape or topology information of the embedded protein, our shape-matching based approach may not produce the correct correspondence in the candidate list. To battle data inaccuracy, We allow the user to manually assign matching constraints based on their biological knowledge of the spatial arrangement of helices. Specifically, the user may designate the correspondence between a small subset of helix edges in the sequence graph and the volume graph. Such information can be translated into additional edge attributes (e.g., $\beta_{S,1}(\{x,y\}) = \beta_{C,1}(\{u,v\}) = H_k$ if edge $\{x,y\}$ and $\{u,v\}$ are the kth corresponding pair) to enforce such explicit matching in the A*-search.


KPWWEDEWEVPRETLKLVERLGAGQFGEVWMGYYNGHTKVAVKSLKQGSMSPDAFLAEANLMKQLQH
QRLVRLYAVVTQEPIYIITEYMENGSLVDFLKTPSGIKLTINKLLDMAAQIAEGMAFIEERNYIHRDLRAANILV
SDTLSCKIADFGLARLIEDNETAREGAKFPIKWTAPEAINYGTFTIKSDVWSFGILLTEIVTHGRIPYPGMTNP
EVIQNLERGYRMVRPDNCPEELYQLMRLCWKERPEDRPTFDYLRSVLEDFFTAT
A
B
C
D
E
F
G
H
I
J
K
L
J
I
H
L
E
D
B
I
J
H
L
E
D
B

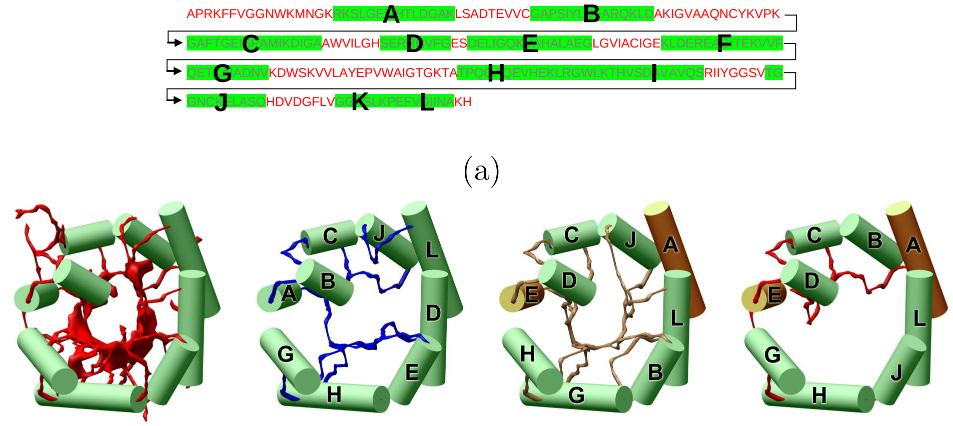
(a)

(b) (c) (d)

Fig. 8. Human Lymphocyte Protein Kinase (3LCK): the amino acid sequence with detected helices (a), the detected helices in the density volume with the skeleton (b), the optimal match between the two sets of helices computed by our method (c), and the actual correspondence (d) which ranks 2nd in the candidate matches.

Figure 9 shows an example of protein 1TIM where the correct helix correspondence was not found in the initial candidate list without any user constraints (we computed a list of 100 top matches, the optimal match is shown in (b)). After the user specified two constraints, however, the correct correspondence was found ranking 9th in the candidate list, as shown in (c,d). The reason that graph matching was not able to reproduce the correct correspondence without user constraints is that the helices in the protein exhibit similar lengths and are spatially close-by, hence pose challenges to matching which is primarily based on helix lengths and distances. Interactive constraints, which carry domain knowledge, can be used in these cases to provide anchor points to guide our method towards a more accurate correspondence. In this example, the constraints are picked by matching the two longest helixes in the sequence with the those in the density volume.

Due to the accumulation of error in the search process because of the inherent ambiguities present in low-resolution imaging, we observe that the amount of user constraints needed to obtain a high-ranked correct correspondence increases with the decreasing accuracy (in terms of resolution as well as noise) of the density map. The shape of the protein is also a factor, as proteins with rotational symmetry require a few user constraints to act as anchor points. As discussed later, the worst-case computational cost of our approach is exponentially proportional to the number of α-helices in the protein. Therefore, an increasing number of user constraints can to be used to reduce the computational cost when performance is a critical factor.



(a)

(b) (c) (d) (e)

Fig. 9. Triose Phosphate Isomerase from Chicken Muscle (1TIM): the amino acid sequence with detected helices (a), the detected helices in the density volume with the skeleton (b), the optimal match between the two sets of helices computed without any constraints (c), the optimal match after the user specified two constraints (colored brown) (d), and the correct correspondence (e) which ranks 9th in the candidate matches.

Although user constraints demand a time investment by a domain expert, we note that the time needed to specify these constraints is much smaller compared to the time needed if the user was to specify all the helix correspondences.

5.5 Performance

The result for all 11 proteins are presented in Table 1, showing the number of helices in the protein sequence, the number of missing helices in each data set (given as the parameter m in creating the sequence graph), the volume (d^3) representing the number of voxels in the cryo-EM density map, the number of user-specified constraints and the rank of the correct correspondence in the candidate list. Table 2 also contains the time taken by our method, and the number of nodes expanded when using the three cost functions ($h_0(M_k), h_1(M_k), h_2(M_k)$) in the A*-search.

Observe from the tables that the graph matching approach in combination with the domain-specific strategies allow accurate identification of protein structure with no or a small amount of human input depending on the quality of the density volume. Also note that the time taken to perform a computation is almost negligible in human terms ($<$ 4 seconds for GroEL when using $h_0(M_k)$), which facilitates a much smoother user-interactive functionality. We would like to point out that using the heuristic functions h_1, h_2 dramatically reduces the number of expansions during A*-search compared to using the

Protein	Helix Count	Missing helices	Volume Size (d^3)	User constraints	Rank
1UF2	4	-	96^3	-	1
2ITG	6	-	64^3	2	4
1IRK	9	-	96^3	-	1
1WAB	9	2	64^3	-	1
1DAI	9	-	64^3	1	5
2BTV	10	-	128^3	-	1
P22 GP5	11	6	128^3	-	4
3LCK	12	5	64^3	-	2
1TIM	12	3	96^3	2	9
RDV P8	14	2	96^3	4	1
GroEL	20	4	128^3	4	1

Table 1
Experiment Results, Accuracy: The rankings of the correct correspondence in the final set of correspondences.

zero function h_0. However, since the time overhead of computing the functions h_1, h_2 is much larger than the zero function, the actual computation time is often slower. Nonetheless, we anticipate that h_1, h_2 can be useful in reducing the memory cost in large data sets.

6 Conclusion and future work

In this paper we reported a novel application of shape modeling and matching in biomedical research which aims at identifying protein structure from the results of an emerging imaging technique. We translated the biological problem into a computational one by representing the shapes of biological data (e.g., protein sequence and density volume) as attributed relational graphs. We solved the helix correspondence problem using graph matching, and we demonstrated the effectiveness of the method on authentic as well as simulated data sets. One of our main contributions is an optimal algorithm for constrained error-correcting graph matching, which will be useful in other shape-matching tasks where the sought match has a linear shape.

Below we discuss some limitations of our approach and directions for future work, starting with general topics in shape modeling and matching and following with topics specific to protein structure identification:

Protein	Helix	Time (seconds)				Nodes expanded		
	Count	S	$h_0(M_k)$	$h_1(M_k)$	$h_2(M_k)$	$h_0(M_k)$	$h_1(M_k)$	$h_2(M_k)$
1UF2	4	1.156	0.015	0.015	0.015	23	16	13
2ITG	6	0.609	0.015	0.015	0.015	65	51	41
1IRK	9	1.500	0.015	0.015	0.015	1813	1195	775
1WAB	9	0.687	0.015	0.015	0.015	2006	1199	644
1DAI	9	0.937	0.031	0.031	0.031	10791	8318	6884
2BTV	10	3.172	0.015	0.031	0.015	5735	3790	595
P22 GP5	11	2.515	0.015	0.015	0.015	514	378	314
3LCK	12	0.968	0.047	0.062	0.078	5685	4013	3001
1TIM	12	1.625	0.204	0.312	0.297	42357	25754	12861
RDV P8	14	1.203	0.203	0.297	0.641	74212	56770	56539
GroEL	20	3.937	3.688	7.859	14.375	774813	603378	564929

Table 2
Experiment Results, Performance: Results from the 11 experiments where the time taken (in seconds) to compute the skeleton S, the time taken to compute the best topology for each of the future cost functions, and the total number of nodes expanded in the A*-search are compared. Observe the significant reduction of nodes expanded when using the better approximations $h_1(M_k)$ and $h_2(M_k)$.

6.1 Shape modeling and matching

Efficiency in matching: One of the limitations of our graph matching algorithm, like other A*-based graph isomorphism techniques, is its high computational cost (both time and memory) for large graphs. In particular, our implementation of the method has difficulty in handling proteins with more than 20 helices without a fairly large number (> 4) of user-specified constraints. In the future we plan to explore variants of the A*-search, including iterative deepening A* and memory-bounded A*, that are better suited for solving the graph matching problem in Section 4 on large data sets.

Skeleton variation: Our application of graph matching relies on a skeleton generated from the iso-surface at a given iso-value. Not only different iso-values will yield different skeletons, oftentimes it is very difficult to find an appropriate iso-value so that the iso-surface accurately represents the *shape* of the volume (for example, when the volume exhibits different brightness levels in different regions). One of our current research agenda is to explore skeletonization techniques which apply directly to gray-scale volumes without the need for thresholding, such as those based on grayscale medial axes [29], topologically watersheds [13], and structure tensors [48,23]. These techniques will produce more robust skeletons and hence more reliable matching results.

6.2 Protein structure identification

User guidance: In our current method, user constraints are specified in the form of known helix correspondences. There are also situations where different forms of user interaction may be more convenient to specify and more effective. For example, proteins such as 1TIM (Figure 9) with rotational symmetry could be more accurately resolved using graph matching if the user provides the orientation of the helices and one helix constraint (so that graph matching can automatically identify the missing helices). We plan to investigate these different interaction modes based on different families of protein structures.

Handling β-sheets: While the current method only considers helix correspondences, we anticipate that a similar shape-matching formulation can be applied to other protein components such as β-sheets. Sheet-detection algorithms are already available for density volumes [1]. We envision that sheets can be represented in the same attributed relational graph abstraction where each sheet is maintained as a *re-visitable* node. The search and corresponding cost functions can thereafter be extended to incorporate these re-visitable nodes. The ability to simultaneously match helices and sheets is believed to result in much increased accuracy than matching helices alone due to added constraints.

Incorporating traditional computational approaches: We are actively working towards the long-term biological goal of recovering the atomic-resolution protein structure from density volumes. Due to the intrinsic resolution limit of cryo-EM imaging and reconstruction, it is unlikely that this goal can be achieved by studying the density volume alone. Only recent work has begun to merge density volumes with traditional computational methods for structure determination. For example, in an ongoing project involving the co-authors, techniques are being investigated where density volumes can provide a *folding space* used for the evaluation of potential structures and improved structure determination in Ab-initio modeling. In the future, we will be investigating the possibility of using Homology modeling to restrict the search space of the alignment of the protein sequence in the density volume. In particular, the skeleton of the density volume can be used to query for possible structure homologue in a database of known structures, such as the Protein Data Bank [14]. The alignment of such a 3D structure homologue with the density volume can then be used as a pre-processing step to remove unnecessary connectivity on the skeleton and therefore greatly increase the accuracy and speed of our matching algorithm.

Identifying the correct correspondence: Our evaluation method assumes the existence of a manual-labeled correspondence. This does beg the question: in practical application of this method, how to identify the correct correspondence within the candidate list when no "ground-truth" is available? Indeed, this is a common problem in structural biology. Many structure prediction algorithms produce a gallery of structures that range in accuracy. The end user is often required to evaluate the model in the context of other data. The ranking achieved by our program is at least on par with the best algorithms if not significantly better. However, we plan to investigate the use of pseudo-atomic models to automate this task further. A pseudo-atomic model of the protein can be built by placing a pseudo-atom for each amino acid in the density volume. Using the helix correspondences as anchors, the optimal placement of these pseudo-atoms can be determined by using previously established distance constraints of pseudo-atoms within secondary structure elements. The protein model with the lowest atomic energy value will be chosen as the correct correspondence for the given density volume.

A note on protein flexibility: Finally, we would like to note that while cryo-EM is well suited for imaging large macromolecular complexes in near-native solution conditions, the method ultimately reconstructs only a single snapshot of the assembly for a given set of images. In the event that there is some intrinsic flexibility in the molecule, the corresponding regions within the density map will appear less well resolved and have lower density values (such as Bacteriophage P22 in Figure 7). Based on empirical evidence, most flexibility on the order of helix or sheet shifts are not easily identifiable until sufficiently high resolutions are reached (typically better than 7Å-8Åresolution). We envision that, given density maps of higher resolution our technique could produce potential secondary structure topologies through regions of disorder that may not have been readily detectable by visual observation.

7 Acknowledgements

This research was supported in part by the National Science Foundation (EIA-0325004, IIS-0705538) and by the National Center for Research Resources (P41RR02250 and P20RR020647).

References

[1] Baker, M., Ju, T., and Chiu, W. Identification of secondary structure elements in intermediate resolution density maps. *Structure 15* (2007), 7–19.

[2] Baldi, P., Brunak, S., Frasconi, P., Soda, G., and Pollastri, G. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics 15* (1999), 937.

[3] Bertrand, G. A parallel thinning algorithm for medial surfaces. *Pattern Recogn. Lett. 16*, 9 (1995), 979–986.

[4] Borgefors, G., Nyström, I., and Sanniti di Baja, G. Computing skeletons in three dimensions. *Pattern Recognition 32*, 7 (1999), 1225–1236.

[5] Bunke, H. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Trans. Pattern Anal. Mach. Intell. 21*, 9 (1999), 917–922.

[6] Bunke, H., and Allermann, G. Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters 1* (1983), 245–253.

[7] Bunke, H., and Messmer, B. T. Recent advances in graph matching. *IJPRAI 11*, 1 (1997), 169–203.

[8] Chen, D., Tian, X., Shen, Y., and Ouhyoung, M. On visual similarity based 3d model retrieval. *Computer Graphics Forum 22*, 3 (2003), 223–232. Eurographics 2003 Conference Proceedings.

[9] Chiu, W., Baker, M., Jiang, W., Dougherty, M., and Schmid, M. Electron cryomicroscopy of biological machines at subnanometer resolution. *Structure (Camb) 13* (2005), 363–372.

[10] Christmas, W. J., Kittler, J., and Petrou, M. Structural matching in computer vision using probabilistic relaxation. *IEEE Trans. Pattern Anal. Mach. Intell. 17*, 8 (1995), 749–764.

[11] Conte, D., Foggia, P., Sansone, C., and Vento, M. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence* (2004).

[12] Cordella, L. P., Foggia, P., Sansone, C., and Vento, M. Performance evaluation of the vf graph matching algorithm. In *International Conference on Image Analysis and Processing* (1999).

[13] Couprie, M., Bezerra, F. N., and Bertrand, G. Topological operators for grayscale image processing. *Journal of Electronic Imaging 10*, 4 (October 2001), 1003–1015.

[14] Dutta, S., and Berman, H. Large macromolecular complexes in the protein data bank: a status report. *Structure 13* (2005), 381–388.

[15] Feng, J., Laumy, M., and Dhome, M. Inexact matching using neural networks. *Pattern Recognition in Practice IV: Multiple Paradigms, Comparative Studies, and Hybrid Systems* (1994), 177–184.

[16] Funkhouser, T., and Shilane, P. Partial matching of 3D shapes with priority-driven search. In *Symposium on Geometry Processing* (June 2006).

[17] Gal, R., and Cohen-Or, D. Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph. 25*, 1 (2006), 130–150.

[18] Gatzke, T., Zelinka, S., Grimm, C., and Garland, M. Curvature maps for local shape comparison. In *Shape Modeling International* (June 2005), pp. 244–256. A local shape comparison technique for meshes.

[19] Herault, L., Horaud, R., Veillon, F., and Niez, J. J. Symbolic image matching by simulated annealing. In *Proc. British Machine Vision Conference (BMVC90)* (1990), pp. 319–324.

[20] Horaud, R., and Skordas, T. Stereo correspondence through feature grouping and maximal cliques. *IEEE Trans. Pattern Anal. Mach. Intell. 11*, 11 (1989), 1168–1180.

[21] Jiang, W., Baker, M., Ludtke, S., and Chiu, W. Bridging the information gap: computational tools for intermediate resolution structure interpretation. *J Mol Biol 308* (2001), 1033–1044.

[22] Ju, T., Baker, M., and Chiu, W. Computing a family of skeletons of volumetric models for shape description. In *Geometric Modeling and Processing* (2006), pp. 235–247.

[23] Lôpez, A. M., Lloret, D., Serrat, J., and Villanueva, J. J. Multilocal creaseness based on the level-set extrinsic curvature. *Comput. Vis. Image Underst. 77*, 9 (2000), 111–144.

[24] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.

[25] Messmer, B. T., and Bunke, H. A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Trans. Pattern Anal. Mach. Intell. 20*, 5 (1998), 493–504.

[26] Moult, J. A decade of casp: progress, bottlenecks and prognosis in protein structure prediction. *Current Opinion in Structural Biology 15* (2005), 285–289.

[27] Nilsson, N. *Principles of Artificial Intelligence.* Morgan Kaufmann Publishers, 1980.

[28] Palágyi, K., and Kuba, A. A parallel 3d 12-subiteration thinning algorithm. *Graph. Models Image Process. 61*, 4 (1999), 199–221.

[29] Peleg, S., and Rosenfeld, A. A min-max medial axis transformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 3*, 2 (March 1981), 208–210.

[30] Penczek, P. A., Grassucci, R. A., and Frank, J. The ribosome at improved resolution: New techniques for merging and orientation refinement in 3d cryo-electron microscopy of biological particles. *Ultramicroscopy 53*, 3 (1994), 251–270.

[31] Rohl, C., Strauss, C., Misura, K., and Baker, D. Protein structure prediction using rosetta. *Methods Enzymol 383* (2005), 66–93.

[32] Sali, A. 100,000 protein structures for the biologist. *Nat Struct Biol 5* (1998), 1029–1032.

[33] Sali, A., and Overington, J. Derivation of rules for comparative protein modeling from a database of protein structure alignments. *Protein Sci. 3* (1994), 1582–1596.

[34] Sanfeliu, A., and Fu, K. A distance measure between attributed relational graphs for pattern recognition. *IEEE Trans. Systems, Man, and Cybernetics 13* (1983), 353–363.

[35] Shapiro, L. G., and Haralick, R. M. Structural descriptions and inexact matching. *IEEE Trans. Pattern Anal. Mach. Intell. 3*, 5 (1981), 504–519.

[36] Shen, M., and Sali, A. Statistical potential for assessment and prediction of protein structures. *Protein Sci. 15*, 11 (2006), 2507–2524.

[37] Shilane, P., Min, P., Kazhdan, M., and Funkhouser, T. The princeton shape benchmark. In *SMI '04: Proceedings of the Shape Modeling International 2004 (SMI'04)* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 167–178.

[38] Sippl, M. J. Boltzmann's principle, knowledge-based mean fields and protein folding. an approach to the computational determination of protein structures. *J Comput Aided Mol Des 7*, 4 (1993), 473–501.

[39] Sippl, M. J. Who solved the protein folding problem? *Structure 7*, 4 (1999), R81–R83.

[40] Sundar, H., Silver, D., Gagvani, N., and Dickinson, S. J. Skeleton based shape matching and retrieval. In *Shape Modeling International* (2003), pp. 130–142, 290.

[41] Svensson, S., Nyström, I., and Sanniti di Baja, G. Curve skeletonization of surface-like objects in 3d images guided by voxel classification. *Pattern Recognition Letters 23*, 12 (October 2002), 1419–1426.

[42] Tsai, W. H., and Fu, K. S. Error-correcting isomorphisms of attributed relational graphs for pattern recognition. *IEEE Trans. Systems, Man, and Cybernetics 9* (1979), 757–768.

[43] Ullmann, J. R. An algorithm for subgraph isomorphism. *J. ACM 23*, 1 (1976), 31–42.

[44] Venclovas, C., and Margelevicius, M. Comparative modeling in casp6 using consensus approach to template selection, sequence-structure alignment, and structure assessment. *Proteins: Structure, Function, and Bioinformatics 61*, S7 (2005), 99–105.

[45] Wang, Y., Fan, K., and Horng, J. Genetic-based search for error-correcting graph isomorphism. *IEEE Transactions on Systems, Man, and Cybernetics, Part B 27*, 4 (1997), 588–597.

[46] Wong, A., You, M., and Chan, A. An algorithm for graph optimal monomorphism. *IEEE Trans. Systems, Man, and Cybernetics 20*, 3 (1990), 628–636.

[47] Wu, Y., Chen, M., Lu, M., Wang, Q., and Ma, J. Determining protein topology from skeletons of secondary structures. *J. Mol. Biol. 350*, 3 (2005), 571–586.

[48] Yu, Z., and Bajaj, C. A structure tensor approach for 3d image skeletonization: Applications in protein secondary structural analysis. In *Proceedings of IEEE International Conference on Image Processing (ICIP'06)* (2006), pp. 2513–2516.

[49] Zhang, J., Siddiqi, K., Macrini, D., Shokoufandeh, A., and Dickinson, S. J. Retrieving articulated 3-d models using medial surfaces and their graph spectra. In *EMMCVPR* (2005), pp. 285–300.

[50] Zhang, Y., and Skolnick, J. The protein structure prediction problem could be solved using the current pdb library. *Proc Natl Acad Sci U S A 102*, 4 (2005), 1029–1034.