Algorithmic Aspects of Protein Structure Similarity (Extended Abstract)

Deborah Goldman* Sorin Istrail† Christos H. Papadimitriou‡

Abstract

We show that calculating contact map overlap (a measure of similarity of protein structures) is NP-hard, but can be solved in polynomial time for several interesting and relevant special cases. We identify an important special case of this problem corresponding to self-avoiding walks, and prove a decomposition theorem and a corollary approximation result for this special case. These are the first approximation algorithms with guaranteed error bounds, and NP-completeness results in the literature in the area of protein structure alignment/fold recognition for measures of structure similarity of practical interest.

A Introduction

Protein structure prediction is going through an important paradigm shift. The Ab initio prediction method aims to derive from the protein sequence, using first principles, the 3D structure of the protein. Despite over 30 years of research generating an immense literature, and considerable progress in understanding the physics, chemistry and biology of the folding process, the success of the method on naturally occurring proteins is very limited. A new exciting research direction of proven practical success is emerging: Protein Fold Assignment[23].

*Department of Mathematics, University of California at Berkeley, Berkeley, CA 94720. Research supported by Phi Beta Kappa Graduate Scholarship and by the DOE, MICS at Sandia National Laboratories, Program DE-AC04-94AL85000. E-mail: dgoldman@cs.berkeley.edu

†Sandia National Laboratories, Applied Mathematics Department, MS 1110, Albuquerque, NM 87185-1110. Research supported by the DOE, MICS Program DE-AC04-94AL85000. E-mail: scistra@cs.sandia.gov

‡Computer Science Division, University of California at Berkeley, Berkeley, CA 94720. Research supported by NSF grant CCR96226361, and JSEP grant FDF 49620-97-1-0220-03-98. E-mail: christos@cs.berkeley.edu

A.1 A Paradigm-shift

The Brookhaven Protein Database (PDB) records several thousands of protein structures. Each such structure gives (basically) the 3D coordinates of all the atoms of the structure. It has hundreds to thousands such coordinates. Two trends were observed as the PDB grows every year incorporating new protein structures. First of all, the structures cluster naturally into "fold families" based on their topological similarity. Second, the number of "new folds" incorporated in the PDB is decreasing "exponentially" from year to year. As the PDB has today (based on several classifications) about 700 fold-families, it is conjectured that there will be a total of about 1000 fold-families for the naturally occuring proteins.

These developments induced a paradigm-shift in structure prediction methods. Instead of basing prediction on first principles, the prediction of the structure of a newly discovered protein can be adequately "solved" by computationally assigning one of the 1000 folds to it. This new direction, *Protein Fold Assignment*, turns out to be the most successful approach to structure prediction as reported in the recent CASP3: International Protein Folding Competition, ASILOMAR 98 [23]. Protein fold assignment methodology is a rich area for combinatorial problems and algorithms necessary to finetune and maturely and adequately solve its computational challenges. It requires computational support for several basic areas of protein structure analysis including: (1) pairwise structure alignment based on various measures of structure similarity, (2) fold clustering/classification and multiple structure alignment, (3) fold recognition/threading.

A.2 Protein Structure Similarity

Due to the paradigm-shift described above, and as the sequence and the three-dimensional structure of

more and more proteins are discovered in the laboratory and catalogued in databases, the comparative study of protein structure has emerged as an important and urgent problem in Computational Biology. Several *measures of protein structure similarity* have been proposed and used over the past few years, attempting to assign to each pair of proteins a distance, presumably capturing the extent to which the two proteins "resemble" each other in structure, origin, and function [17, 7, 34, 8, 28, 9, 16, 25, 27, 30, 29, 22, 17, 4, 36, 5, 24, 35, 37, 38, 32, 21, 32, 19, 17, 18]. The most important and popular such measures used in the Protein Science literature are these:

- The root-mean-square distance (RMSD) of the two proteins —the two three-dimensional structures are superposed in such a way that their L_2 metric is minimized [28, 9, 11, 12, 13, 16, 25, 27, 30, 29]
- A related measure is the difference of the distance matrices [21, 32, 22].
- In this paper we examine an emerging important distance measure called *contact map overlap*. To compute this distance between two proteins, the monotonic one-to-one mapping between two subsets of the two sets of monomers is found that maximizes the number of "nearby" or "contact" pairs that are mapped to each other [19, 17, 18].
- Various other *ad hoc* scores based on local secondary structure, hydrogen bonding pattern, burial status, or interaction environment [22, 17, 4, 7, 8, 36, 34, 5, 24, 35, 37, 38, 32]

There are many conceptual difficulties associated with these measures. First, some of them are notoriously non-robust [28, 9, 29, 17, 22, 4]. It is well-known that the mapping from sequence to structure ("the protein folding problem") is very complex and non-local [10, 3]; this means that there is very little relationship between the edit distance of two proteins and their three-dimensional similarity. Unfortunately, many alignment algorithms introduce significant biases by disregarding this point. Also, the hydrophobic/hydrophilic character of the residues (believed by many to be the single most important predictor of structure [14]) is often not reflected in the distance calculation (this is especially true in the RMSD distance, and even more serious in the so-called C-alpha alignments [17]). Further, most models fail to take into account the "excluded volume" aspect of protein structure —that is to say, the fact that protein backbones are self-avoiding walks.

There are also many *computational difficulties* associated with such measures, as many of these measures require the solution of intractable optimization problems. The optimization problems draw their complexity from the non-locality of the scoring function, and the handling of insertions and deletions. As a consequence all existing structural alignment algorithms use *ad hoc* simplifications either of the scoring function or of the search procedure [17]. They attempt to reduce the dimensionality of the problem by performing at least part of the search at the level of secondary structure elements. Other methods employ dynamic programming [8] and Monte Carlo simulations [17] or heuristics [17, 28, 9, 22, 4, 36, 5, 24, 35, 37, 32, 21, 19], without, however, a rigorous analysis.

There is a natural list of desiderata for a structural similarity measure:

- it should not penalize too heavily insertions and deletions
- it should be reasonably robust, in that small perturbations of the definition should not make too much difference in the measure
- it should be easy to compute (or at least rigorously approximated)
- it should be able to discover both local and global alignments
- it should be able to discover hydrophilic-hydrophobic alignments
- it should take into account the self-avoiding nature of a protein
- it should be subject to empirical studies on Protein Data Base (PDB) data to validate its success in capturing structural similarity
- even if one comes up, from a theoretical standpoint, with a "perfect" measure, it will be difficult to displace entrenched measures, used for years by protein scientists. Acceptance in the field is thus a further desideratum.

For all of these reasons, we chose to concentrate on *contact map overlap*, explained next. In our view, no other measure comes even close to satisfying the above list of desiderata.

A fundamental concept in protein structure analysis is that of a "contact" —an instance in which two amino acids of the protein come very close to each other, presumably forming some kind of bond. Understanding the mathematical structure of the set of contacts of a self-avoiding walk is a long-standing open problem (in this paper we make significant progress in this problem, proving a decomposition theorem for the contact graph of *two*-dimensional walks, *Theorem* 8). The study of two-dimensional self-avoiding walks on the square lattice, including the structure of their contact maps, is of basic importance in the statistical mechanics studies of lattice models of protein folding. A *contact map* is a useful graph-theoretic abstraction (and two-dimensional depiction) of the structure of a protein. For a protein of size n, and a given threshold $\mu > 0$, the contact map M_μ is an $n \times n$ 0-1 matrix, whose entry $M_\mu(i, j)$ equals 1, if the distance between amino acids i and j is less than or equal to μ, and 0 otherwise. The contact map can be also viewed as a Hamilton path (usually depicted horizontally), with nodes representing the amino acids, and with edges added that join pairs of nodes whose centers of gravity have been found to be closer to each other than the fixed threshold μ (see Figure 1 for an example). The center of gravity is one of the possible choices for represnting the amino acid by a central point. Contact maps are used for secondary structure prediction, fold identification, fold classification, fold assignment, protein structure alignment, and threading [34, 8, 7]. They are also used extensively for the calculation of *statistical potentials*, a most popular example being the *Miyazawa-Jernigan matrix* [31], a 20×20 matrix, whose entries reflect the frequency of contact between pairs of amino acids in a protein database. These potentials are in turn used for simulating protein folding, judging the quality of proposed protein models, as well as in protein design.

Contact maps are also used extensively in the study of *RNA structure*. The three-dimensional structure of RNA is also the object of current intense study, and contact maps have been employed in it [2, 26, 33]. Calculating the contact map overlap distance of two RNA structures is another fundamental problem. It had been known that the three-dimensional structure of RNA is more dominated by its two-dimensional structure. Our results are powerful enough to give the first rigorous approximation algorithms for the RNA case in full generality.

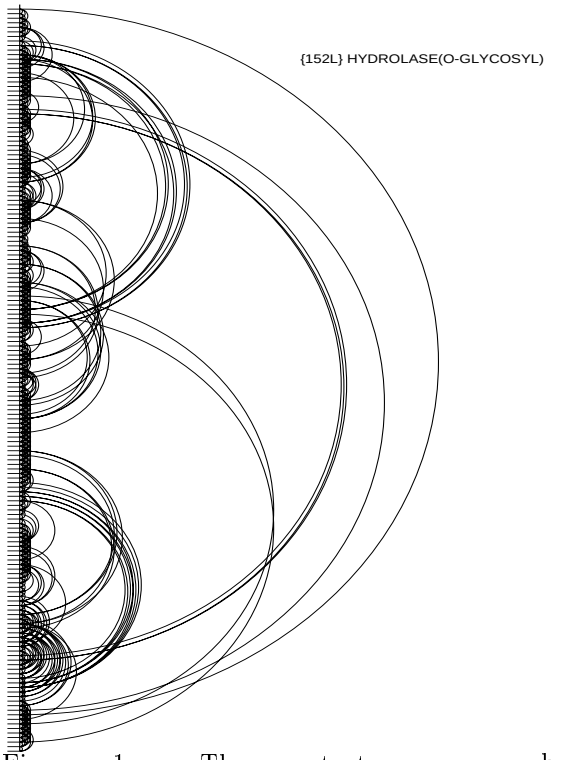


Figure 1: The contact map graph of HYDROLASE(O-GLYCOSYL)

Our Results

In this paper we embark on a theoretical and algorithmic study of protein structure similarity, focussing on the measure of contact map overlap, which, as we have argued, is the one that appears the most susceptible and ultimately useful (especially in view of our positive results explained below).

- We formulate the calculation of the contact map overlap of two protein structures as a graph-theoretic optimization problem.
- We prove that this optimization problem is in fact MAXSNP-hard to solve, and it is NP-hard even if the underlying contact maps are the contact maps of two-dimensional self-avoiding walks.
- We identify two important special cases of contact map graphs, the *queue* and the *stack* (previously studied in the context of VLSI [6, 20]), as well as the *staircase* (a special case of the queue) and the *augmented staircase* (a staircase with a stack embedded in it in a restricted way). We develop polynomial-time dynamic programming

algorithms that solve the contact map overlap problem for some of these graphs. We point out that using these algorithms we can approximately compute the contact map overlap of two RNA structures.

- We study the contact map graphs of self-avoiding walks in the two-dimensional grid, and we show a structural theorem establishing that any such graph is the union of two augmented staircases and a stack. (We also show that such graphs are NP-hard to recognize, and therefore there is little hope for a more restrictive if-and-only-if characterization.) This is an important first step in identifying positive properties of protein contact maps by exploiting their self-avoiding nature.

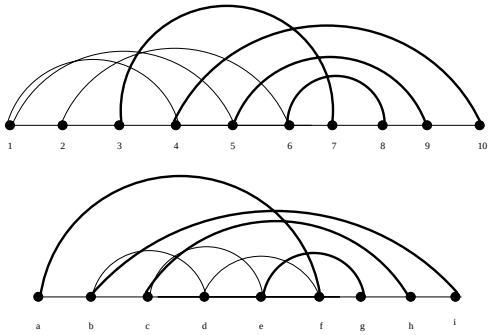
- As a corollary of the results in the two topics above, we develop a polynomial-time algorithm that approximates the contact map overlap of two self-avoiding two-dimensional walks within a factor of 3.

To our knowledge, this is the first theoretical study of protein structure similarity. An independent formulation of the problem and NP-completeness proof for a different measure involving RNA can be found in [15]. Also, a more general measure was used by [1] to provide hardness results and algorithms for threading.

B Problem Formulation and NP-completeness

A *contact map* (n, E) is an undirected graph $G = (V, E)$ such that the set of vertices $V = \{1, 2, \ldots, n\}$ is linearly ordered (see Figure 2 for an example). Contact maps are useful representations of proteins, where the vertices are the amino acids of the protein, and the edges are pairs of amino acids whose centroids are closer than a fixed threshold value (typically a few Angstroms).

The CONTACT MAP OVERLAP problem is the following optimization problem: Given two contact maps (n, E) and (m, E'), find two subsets $S \subseteq \{1, \ldots, n\}$ and $S' \subseteq \{1, \ldots, m\}$ with $|S| = |S'|$ such that the cardinality $|\{[u, v] \in E : u, v \in S, [f(u), f(v)] \in E'\}|$ is as large as possible, where f is an order-preserving bijection between S and S'. For example, the maximum overlap between the two contact maps shown in Figure 2 is 4, obtained by taking $S = \{3, 4, 5, 6, 7, 8, 9, 10\}$ and $S' = \{a, b, c, e, f, g, h, i\}$.



Alignment: (3,a), (4,b), (5,c), (6,e), (7,f), (8,g), (9,h), (10,i)

Figure 2: Contact overlap example

We next show that this problem is hard to solve or approximate (proof omitted; an independent and related NP-completeness proof can be found in [**?**]):

Theorem 1 *The* CONTACT MAP OVERLAP *problem is MAXSNP-complete even if both contact maps have maximum degree one.*

It is intuitively clear and widely accepted that contact maps of real proteins are far from arbitrary collections of edges, since they have a specialized structure reflecting the geometry of proteins. We next introduce a special class of contact maps that seem to go a long way towards capturing this structure. A *self-avoiding walk on the two-dimensional grid* is a one-to-one mapping f from $\{1, 2, \ldots, n\}$ to Z^2 such that $\|f(i) - f(i+1)\|_2 = 1$ for $i = 1, \ldots, n-1$. Associate with each such walk f its contact map $G_f = (\{1, 2, \ldots, n\}, E)$, where $E = \{[i, j] : |i - j| > 1, \|f(i) - f(j)\|_2 = 1\}$. That is, G_f is the contact map that represents all distance-one contacts of the walk excluding consecutive neighbors. We shall informally refer to G_f itself as a "self-avoiding walk;" notice that such graphs have maximum degree two (with the exception of nodes 1 and n, which may have degree three). Unfortunately, the problem is hard here as well:

Theorem 2 *The* CONTACT MAP OVERLAP *problem is NP-complete even if both contact maps are self-avoiding walks.*

As we shall see, however, this special case of the

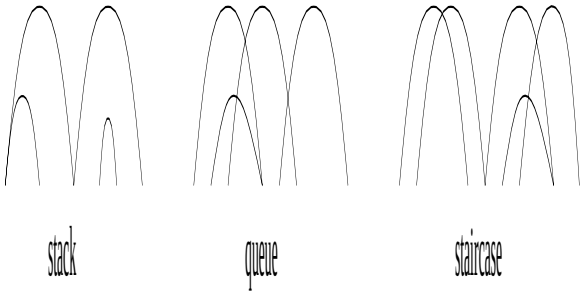


Figure 3: Stack, queue, and staircase examples

problem can be approximated within a factor of three (see Corollary 9 below).

C Polynomial Algorithms for Special Cases

Define a *stack* to be a contact map (n, E) such that if $[i,j],[k,\ell] \in E$ then the intervals $[i,j]$ and $[k,\ell]$ either contain one another, are disjoint, or overlap at an endpoint. Define a *queue* to be a contact map (n, E) such that if $[i,j],[k,\ell] \in E$ then the intervals $[i,j]$ and $[k,\ell]$ do not contain one another unless they share an endpoint. Stacks and queues have been studied extensively in [6, 20]. A *staircase* is a queue which contains sets of mutually overlapping intervals such that either no two intervals in differing sets meet, or at most two intervals overlap at an endpoint. (See Figure 3 for examples.)

Theorem 3 *There is an $O(n^6)$ algorithm for finding the maximum overlap of two degree-2 contact maps, one of which is either a stack or a staircase.*

Sketch: All algorithms are based on dynamic programming. In the stack case, for example, let S be a degree-2 stack containing n vertices labelled 1 through n, and let G be an arbitrary degree-2 graph containing m vertices labelled 1 through m. We compute the contact overlap of S and G, $co(S,G)$, using dynamic programming. The subproblems, or table entries, have the following form. We compute the contact overlap of subgraphs of the original two graphs which consist, given two pairs of positive integers $1 \le a < b \le n$ and $1 \le c < d \le m$, of the set of all edges (i,j) in S with $a \le i < j \le b$ and the set of all edges (k,l) in G with $c \le k < l \le d$. We denote such graphs $S_{(a,b)}$ and $G_{(c,d)}$. For technical reasons, in the case when there are two edges which meet b (respectively, d), we may omit the edge with the lower second coordinate, in which case we denote the graphs $S_{(a,\hat{b})}$ (respectively, $G_{(c,\hat{d})}$). As well, to compute the answers recursively, we may require that the contact overlap be computed with the restriction that the edge with the lowest endpoint in one set be mapped to the edge with the lowest endpoint in the other set (it will never be the case that we make this restriction when there are two edges with the lowest endpoint), denoted $co(S_{(a,b)}, G_{(c,d)})^l$, the edges with the highest endpoints be mapped to each other (in this case if there are two edges to consider we choose the edge with the lower second coordinate), denoted $co(S_{(a,b)}, G_{(c,d)})^h$, or both (in this case we may assume one edge does not include the other), denoted $co(S_{(a,b)}, G_{(c,d)})^{lh}$. The number of table entries is $O(n^4)$. The optimum is given by

$$\begin{aligned} co(S,G) = \max_{2\le i\le n, 2\le j\le m} \{\max\{ & co(S_{(1,i)}, G_{(1,j)})^h, \\ & co(S_{(1,\hat{i})}, G_{(1,j)})^h, co(S_{(1,i)}, G_{(1,\hat{j})})^h, \\ & co(S_{(1,\hat{i})}, G_{(1,\hat{j})})^h\}\}. \end{aligned}$$

Suppose we would like to compute the contact overlap, $co(S_{(a,b)}, G_{(c,d)})^{lh}$, of the two graphs pictured in Figure 4. Then, recursively, the contact overlap is given by

$$\begin{aligned} co(S_{(a,b)}, G_{(c,d)})^{lh} &= 2 + co(S_{(a+1,e-1)}, G_{(c+1,g-1)}) \\ &+ \max\{co(S_{(e,f)}, G_{(g,h)})^l, co(S_{(e+1,f)}, G_{(g+1,h)})\} \\ &+ \max\{co(S_{(f,b-1)}, G_{(h,d-1)})^l, co(S_{(f+1,b)}, G_{(h+1,d)})^h, \\ &co(S_{(f+1,b-1)}, G_{(h+1,d-1)})\}. \end{aligned}$$

The remaining recursions are similar to those presented above and are based on a case analysis of the existence of edges which share an endpoint with restricted edges. The optimum and table entries can be computed in $O(n^2)$ time, completing the proof of the theorem. ■

In fact, if the contact maps in the previous theorem are derived from self-avoiding walks, then the runtime

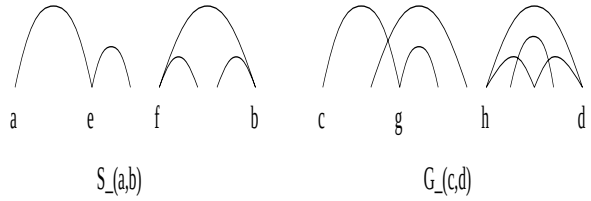


Figure 4: Stack algorithm example.

can be improved to $O(n^4)$. We also note that all our algorithms generalize to constant degree-bounded graphs, which is a reasonable assumption for protein contact maps.

The contact maps of all known RNA structures except for one are known to be decomposable into two degree-1 stacks, hence their maximum overlap can be approximated within a factor of 2 (optimize the overlap of each stack contained in one contact map against the other contact map and take the larger value).

Unfortunately, one cannot extend this result much further, since we can show that the problem becomes NP-complete even in the case of two contact maps which are unions of two degree-1 stacks. We also conjecture that finding the maximum overlap of two *queues* is NP-complete. However, we have the following consolation result:

Theorem 4 *Every queue can be decomposed into two staircases.*

Consequently, the maximum overlap of two degree-2 queues can be approximated within a factor of 2 (decompose one queue into two staircases, optimize the overlap of each staircase with the second queue, and choose the larger value).

Define now an *augmented staircase* to be a contact map which can be decomposed into a staircase and a stack such that for every stack edge and for every staircase edge, the intervals formed by the edges are disjoint, overlap at an endpoint, or the interval formed by the staircase edge contains the interval formed by the stack edge. Our strongest polynomial-time result is the following:

Theorem 5 *There is an $O(n^6)$ algorithm for finding the maximum overlap of two degree-2 contact maps, one of which is an augmented staircase.*

Sketch: The algorithm is more complicated than the stack comparison algorithm, and we provide only a rough sketch here. Let A be a degree-2 augmented staircase containing n vertices and let G be an arbitrary degree-2 graph containing m vertices. We may assume we are given the decomposition of A into its staircase, T, and stack, S. Number the edges of A and G according to the ascending order of their right endpoints; if two edges share the same right endpoint, the edge with the lower left endpoint receives the higher number. As in the stack comparison algorithm, we compute the contact overlap using a dynamic programming algorithm, and we use that algorithm to compute a table of entries comparing subgraphs of S to subgraphs of G. In the present algorithm, table entries are indexed by four items: an edge, $e = (i, j)$, contained in T, an edge, $f = (k, l)$, contained in G, and either the next two entries are blank, denoted "-", or they contain a higher edge, g, of T whose interval overlaps that of e nontrivially (at more than one point) and a higher edge, h, of G whose interval interacts with f in a similar fashion. The table entry contains a constrained contact overlap of the subgraph of A, $A_{(e,g)}$, which consists of all edges $e' \leq e$, except for those edges of S contained in the interval formed by the left endpoints of e and g, and the subgraph of G, $G_{(f,h)}$, which consists of edges $f' \leq f$. The constraints are that e must map to f under the bijection, and that matched edges of $G_{(f,h)}$ which overlap f must also overlap h nontrivially. The constraints ensure consistency under recursion. The contact overlap of A and G is given by

$$co(A, G) = \max\{co(S, G), \max_{e \in T, f \in G}\{co(A_{(e,-)}, G_{(f,-)}) + s(j, l)\}\},$$

where

$$s(j, l) = \begin{cases} \max\{co(S_{(j,n)}, G_{(l,m)})^l, co(S_{(j+1,n)}, G_{(l+1,m)})\} \\ \text{edges of S, G meet j, l as left endpoints} \\ co(S_{(j+1,n)}, G_{(l+1,m)}) \\ \text{otherwise.} \end{cases}$$

The intuition is that in an optimal bijection either there is a highest edge of T which maps to an edge of G, or there is no such edge. The recursion for the subproblems is based on branching on the next lowest edge of T to be matched, and we may need to look up at most two entries of the table comparing subgraphs of S to subgraphs of G. The reason for recording the additional set of edges, g and h, is that they are the highest set of edges which have been matched previously which overlap e and f nontrivially; they ensure consistency in the recursion. Adding the preprocessing time, $O(n^6)$ to the product of the number of table entries, $O(n^4)$, times the time to compute an entry, $O(n^2)$, we derive an $O(n^6)$ algorithm. ■

D A Decomposition Theorem

The following result, interesting in its own right, is the basis of our approximation algorithms.

Theorem 6 *Any self-avoiding walk can be decomposed into 2 stacks and 1 queue.*

Proof: For each vertex in the walk, we assign a label O (for over) or U (for under) to its adjacent edges in the lattice which are not edges in the walk. Edges in the lattice will then have multisets of labels consisting of 0, 1, or 2 members. Labels are assigned inductively as follows:

- Label non-walk edges adjacent to vertex 1 as follows:
  - Assign O to one of the edges perpendicular to edge $\{1,2\}$ of the walk and assign U to the other.
  - If the remaining edge adjacent to vertex 1 is contained in the contact map, assign it the same label as whichever of the 2 previously labelled edges is contained in the closed curve formed by walk edges and the edge we are considering, else label it arbitrarily.
- Label non-walk edges adjacent to vertex i, where $2 \leq i \leq n-1$, as follows:
  - If the walk edges adjacent to vertex i are parallel (i is "straight"), then at least one non-walk edge adjacent to i must lie in the same lattice square as an edge labelled by $i-1$, say with label L. Assign label L to the non-walk edge adjacent to i which shares the square and assign label $\{O,U\} \setminus \{L\}$ to the other non-walk edge adjacent to i.
  - If i is a corner, then its two adjacent non-walk edges will share the same label. If one of the two edges shares a lattice square with an edge labelled L by $i-1$, then we assign the two edges label L. If neither edge shares a lattice square with an edge labelled by $i-1$, then $i-1$ must be a corner; assign the edges adjacent to i the opposite of the label assigned by $i-1$.
- Label the non-walk edges adjacent to vertex n as follows:
  - At least one non-walk edge perpendicular to $\{n-1,n\}$ and adjacent to n must lie in the same lattice square as an edge labelled by $n-1$, say with label L. Assign label L to the edge adjacent to n which shares the square and assign label $\{O,U\} \setminus \{L\}$ to the other edge perpendicular to $\{n-1,n\}$ and adjacent to n.

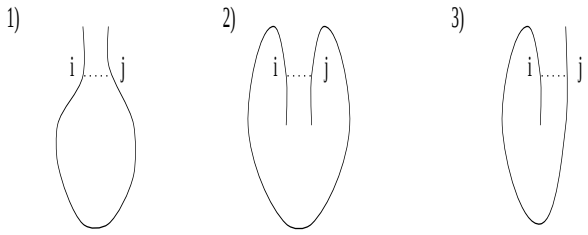


Figure 5: Three possible cycle configurations.

  - If the remaining edge adjacent to vertex n is contained in the contact map, assign it the same label as whichever of the 2 previously labelled edges is contained in the closed curve formed by walk edges and the edge we are considering, else label it arbitrarily.

One can check that the labelling is well-defined.

Now, the edges of the contact map are exactly those edges which have been assigned 2 labels. We prove in the next two claims that the two graphs consisting, respectively, of edges labelled by $\{O,O\}$ and $\{U,U\}$ are stacks and the graph consisting of edges labelled by $\{O,U\}$ is a queue.

Claim 1 *Let G_L, where $L = O$ or $L = U$, denote the graph induced by edges labelled by $\{L,L\}$. Then G_L is a stack.*

Proof: Let $\{i,j\}$ be an edge of G_L such that $i < j$. To show that G_L is a stack, it suffices to prove that there is no edge $\{k,l\}$ of G_L such that $i < k < j < l$ or $l < i < k < j$. For ease of exposition, assume $i \neq 1$ and $j \neq n$; these cases can be handled separately by an easy case analysis. Without loss of generality, there are 3 possible configurations of the graph consisting of walk edges and the edge $\{i,j\}$ depending on whether the ends of walk are inside or outside the cycle formed by walk edges between i and j and the edge $\{i,j\}$. The 3 configurations are pictured in Figure 5.

We analyze the 3 cases below.

Case 1: Neither vertex 1 nor vertex n is contained in the interior of the cycle.

Then inductively we can show that all edges assigned the label L by vertices k such that $i < k < j$ are in the interior of the cycle, whereas all edges assigned the label L by vertices l such that $l < i$ or

$j < l$ are outside the cycle. Thus there is no edge $\{k, l\}$ of G_L such that $i < k < j < l$ or $l < i < k < j$.

Case 2: Both vertices 1 and n are contained in the interior of the cycle.

Then inductively we can show that all edges assigned the label L by vertices k such that $i < k < j$ are outside the cycle, whereas all edges assigned the label L by vertices l such that $l < i$ or $j < l$ are in the interior of the cycle. Thus there is no edge $\{k, l\}$ of G_L such that $i < k < j < l$ or $l < i < k < j$.

Case 3: Exactly one of vertices 1 and n is contained in the interior of the cycle, say without loss of generality vertex 1.

This case is, in fact, impossible because inductively we can show that j must assign the label $\bar{L}$ to edge $\{i, j\}$.

Thus, G_L is, indeed, a stack.

Claim 2 *Let $G_{O,U}$ denote the graph induced by edges labelled by $\{O, U\}$. Then $G_{O,U}$ is a queue.*

Proof: Let $\{i, j\}$ be an edge of $G_{O,U}$ such that $i < j$. To show that G_L is a queue, it suffices to prove that there is no edge $\{k, l\}$ of $G_{O,U}$ such that $i < k < l < j$. Again, assume $i \neq 1$ and $j \neq n$; these cases can be checked separately. As in the previous claim, we must consider the 3 possible configurations involving $\{i, j\}$ and the walk. However, inductively we can show that cases 1 and 2 are impossible (in those cases i and j must label $\{i, j\}$ with the same label). Thus we consider the third case. Let L be the label vertex i assigns to edge $\{i, j\}$, so vertex j assigns $\bar{L}$. Then, by induction, a vertex k with $i < k < j$ may only assign L to edges outside the cycle and may only assign $\bar{L}$ to edges in the interior of the cycle. Thus, as desired, it is impossible for there to be an edge $\{k, l\}$ of $G_{O,U}$ such that $i < k < l < j$. Consequently, $G_{O,U}$ is a queue. ■

Since G_O, G_U, and $G_{O,U}$ form a decomposition of the walk into 2 stacks and 1 queue, this completes the proof of the theorem. ■

There is little hope of finding a much tighter characterization of self-avoiding walks, since we can show that it is NP-complete to tell whether a given contact map is a self-avoiding walk. Using the above decomposition, we immediately obtain a $\frac{1}{4}$-approximation algorithm for computing the contact map overlap between two self-avoiding walks: Decompose one walk into two stacks and two staircases (which have maximum degree 2) (by Theorems 6 and 4) and then compute the maximum overlaps with the second contact map (Theorem 3):

Corollary 7 *There is a $\frac{1}{4}$-approximation algorithm for computing the contact map overlap between two self-avoiding walks.*

A better approximation ratio, however, is obtained by a more sophisticated decomposition, based on a much more elaborate case analysis:

Theorem 8 *Any contact map of a self-avoiding walk in the two-dimensional square lattice can be decomposed into one stack and two augmented staircases.*

Sketch: We provide a rough sketch. Begin by decomposing the walk into 2 stacks and a queue as in the previous theorem. There are two important observations which follow from the proof of the previous theorem which we make use of here:

1. No queue edge labelled by (O, U) (here order is important, that is, O is assigned to the lower numbered coordinate) meets an edge labelled by (U, O) except possibly at their left and right endpoints (or viceversa).

2. The interval formed by a stack edge labelled by (L, L) does not intersect the L endpoint of a queue edge labelled by $(L, \bar{L})$ or $(\bar{L}, L)$ except possibly at one of its endpoints.

Decompose the queue into 2 staircases using the following algorithm. First, as in the proof of Theorem 5, number the edges in the queue according to the ascending order of their right endpoints; if two edges share an endpoint, the lower number is assigned to the edge with the lower left endpoint. Repeat the following step, alternating staircases, until no edges remain:

Let (i, j) be the lowest numbered unassigned edge. Assign all edges (k, l) with $i < k \leq j - 1$ to the same staircase. If the new lowest unassigned edge is labelled differently (that is, the order of the labels is different) from the edge which was assigned last, then repeat this step using the same staircase.

It follows from observation 1 noted above and by induction that this algorithm correctly decomposes the queue into two staircases. Due to observation 1, the staircases contain sets of mutually intersecting edges which are all labelled in the same way, (O, U)

or (U, O), and only at most two intervals in these sets intersect at an endpoint.

Next, using observation 2, we can add stack edges from the $\{O, O\}$ and $\{U, U\}$ stacks to the staircases to form augmented staircases, and the remaining stack edges are collected in a final stack. ■

We can now state our strongest approximation result:

Corollary 9 *There is a polynomial-time $\frac{1}{3}$-approximation algorithm for computing the contact map overlap between a self-avoiding walk and any other contact map.*

There are examples establishing that this factor is the best possible without major modifications of this algorithm.

E Discussion and Open Problems

One immediate open problem is the development of a better approximation for the contact map overlap of self-avoiding walks. We do not expect more favorable decompositions for such graphs, and so a different approach seems to be required. On lower bounds, we do not even know whether this special case is MAXSNP-complete; the planarity of the problem inhibits lower bounds, while its subgraph isomorhism character seems to rule out polynomial approximation schemes. Finally, we know of no satisfactory approximation algorithm for the general problem.

It would be interesting to discover favorable properties of three-dimensional self-avoiding walks. Our decomposition technique seems inherently two-dimensional, in that it exploits topological properties of the plane, such as the dichotomy between "folds" and "spirals." Finally, we conjecture that it is NP-hard to maximize the overlap of two queues.

The second author, together with Brian Walenz, are developing the Tortilla Protein Folding software system at Sandia National Laboratories. It includes computational tools related to contact maps and their use in protein structure analysis and prediction. We would like to thank Brian Walenz for useful discussions related to this research.

References

[1] T. Akutsu and S. Miyano. On the approximation of protein threading. *Proceedings of the First Annual International Conference on Computational Molecular Biology*, 1997:3–8, 1997.

[2] V. Bafna, S. Muthukrishnan, and R. Ravi. Computing similarity between rna strings. *DIMACS TR-96-30*, 1996.

[3] B. Berger and T. Leighton. Protein folding in the hydrophobic-hydrophilic (*hp*) model is np-complete. *Journal of Computational Biology*, 5, 1998.

[4] T. L. Blundell and M. S. Johnson. Catching a common fold. *Protein Science*, 2:877–883, 1993.

[5] J. U. Bowie, R. Luthy, and D. Eisenberg. A method to identify protein sequences that fold into known three-dimensional structures. *Science*, 253:164–170, 1991.

[6] F. R. Chung, F. T. Leighton, and A. Rosenberg. Embedding graphs in books: a layout problem with applications to vlsi design. *SIAM J. Algeb. and Discr. Meth*, 8, 1987.

[7] Protein Structure Classification. London: Uc bsm, http://www.biochem.ucl.ac.uk/bsm/cath/. 1995.

[8] Structural classification of proteins. London, http://scop.mrc-lmb.cam.ac.uk/scop/.

[9] F. E. Cohen and M. J. E. Sternberg. On the prediction of protein structure: the significance of the root-mean-square deviation. *J. Mol. Biol.*, 138:321–333, 1980.

[10] P. Crescenzi, D. Goldman, C. H. Papadimitriou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. *Journal of Computational Biology*, 5, 1998.

[11] R. Diamond. On the comparison of conformations using linear and quadratic transformations. *Acta Crystallogr. sect. A.*, 32:1–10, 1976.

[12] R. Diamond. A note on the rotational superposition problem. *Acta Crystallogr. sect A.*, 44:211–216, 1988.

[13] R. Diamond. On the multiple simultaneous superposition of molecular structures by rigid body transformations. *Protein Science*, 1:1279–1287, 1992.

[14] K. Dill, S. Bromberg, K. Yue, K. M. Fiebig, D. P. Yee, P. D. Thomas, and H. S. Chan. Principles of protein folding – a perspective from simple exact models.

[15] P. Evans. Finding the common subsequences with arcs and pseudoknots. *Proceedings of the 10th Annual Symposium on Combinatorial Pattern Matching (CPM99), Warwick, UK, July 22-24, 1999*, 1999.

[16] P. R. Gerber and K. Muller. Superimposing several sets of atomic coordinates. Acta Informatica.

[17] A. Godzik. The structural alignment between two proteins: Is there a unique answer ? *Protein Science*, 5:1325–1338, 1995.

[18] A. Godzik, J. Skolnick, and A. Kolinski. A topology fingerprint approach to inverse protein folding problem. *J. Mol. Biol.*, 227:227–238, 1992.

[19] A. Godzik, J. Skolnick, and A. Kolinski. Regularities in interaction patterns of globular proteins. *Protein Engineering*, 6:801–810, 1993.

[20] L. Heath and S. Istrail. The page number of genus g graphs is $o(g)$. *J. ACM*, 39, 1992.

[21] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, 233:123–138, 1993.

[22] L. Holm and C. Sander. Mapping the protein universe. *Science*, 273:595–602, 1996.

[23] http://PredictionCenter.llnl.gov/casp3/.

[24] D. T. Jones, W. R. Taylor, and J. M. Thornton. A new approach to protein fold recognition. *Nature*, 358:86–89, 1992.

[25] W Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallgr. sect A.*, 32:922–923, 1976.

[26] H. Lenhof, K. Reinert, and M Vingron. A polyhedral approach to rna sequence structure alignment. *Proceedings of the Second Annual International Conference on Computational Molecular Biology, S. Istrail, P. Pevzner, M. Waterman, Editors)*, 1998:153–159, 1998.

[27] A. M. Lesk. A toolkit for computational molecular biology. ii. on the optimal superposition of of two sets of molecules. *Acta Crystallogr. sect A.*, 42:110–113, 1986.

[28] V. N. Maiorov and G. M. Crippen. Significance of root-mean-square deviation in comparing three-dimensional structures of globular proteins. *J. Mol. Biol.*, 235:625–634, 1994.

[29] V. N. Maiorov and G. M. Crippen. Size-independent comparison of protein three-dimensional structures. *Protein: Structure, Function and Genetics*, 22:273–283, 1995.

[30] A. D. McLaghlan. A mathematical procedure for superimposing atomic coordinates of proteins. *Acta Crystallogr. sect A.*, 26:656–657, 1972.

[31] S. Miyazawa and R. L. Jernigan. Estimation of effective enterresidue contact energies from protein crystal structures: quasi-chemical approximation. *Macromolecules*, 18:534–552, 1985.

[32] C. Sander and R. Schneider. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins: Struct. Funct. Genet.*, 9:56–68, 1991.

[33] D. Sankoff. Simultaneous solution of the rna folding, alignment, and protosequence problems. *SIAM J. Appl. Math*, 45:810–825, 1985.

[34] Protein structure comparison by alignment of distance matrices. http://www2.ebi.ac.uk/dali/. 1995.

[35] W. R. Taylor and C. A. Orengo. A holistic approach to protein structure alignment. *Protein Eng.*, 2:505–519, 1989.

[36] W. R. Taylor and C. A. Orengo. Protein structure alignment. *J. Mol. Biol.*, 208:1–22, 1989.

[37] G. Vriend and C. Sander. Detection of common three-dimensional substructures in proteins. *Proteins: Struct. Funct. Genet.*, 11:52–58, 1991.

[38] M. Zuker and R. L. Somorjai. The alignment of proteins structures in three dimensions. *Bull. Math. Biol.*, 51:55–78, 1989.