Graph Alignment-Based Protein Comparison

Tanisha Saxena and Daniel Xu

January 23, 2021

Abstract

Inspired by the question of identifying mechanisms of viral infection, we are interested in the problem of comparing pairs of proteins, given by amino acid sequences and traces of their 3-dimensional structure. While it is true that the problem of predicting and comparing protein function is one of the most famous unsolved problems in computational biology, we propose a heuristic which poses it as a simple alignment problem, which - after some linear-algebraic pre-processing - is amenable to a dynamic programming solution.

Contents

1 Introduction

Sequence alignment is an important problem in biology that seeks to compare sequences of information such as DNA or proteins. Traditional algorithms like Needleman-Wunsch [NW70] and Smith-Waterman [SW+81] align sequences using dynamic programming. However, a key idea in biology is that function is related to structure; thus, ideal comparisons of protein function should take structural information into account. Contact maps are 2D representations of the 3D structures of sequences. They come in the form of symmetrical binary matrix M where each value $M_{i,j}$, is marked 1 if the ith and jth amino acids of the protein are in contact with each other. By creating a binary matrix to store the physical contacts of the protein, contact maps are able to store the structural information of a protein within a 2D matrix, and thus can be more easily used by computers to computes scores based on physical characteristics.

There exist previous works which perform on alignments of self-contact maps of pairs of proteins. For example, CMAPi [PBB08] uses 4D dynamic programming to match up the 1's within one contact map to the 1's within the contact map of another protein, as a way to compare the physical structure of a protein via the comparison of pairs of 2D matrices. However, the diagonals of the matrix are not necessarily aligned together and therefore it does not explicitly utilize sequence information.

An important idea that we use in our work is to partially pose protein comparison as an alignment of graphs. The graph alignment problem seeks to find common substructures between pairs of graphs. As in sequence alignment, the graph alignment problem can be posed either locally or globally. Global graph alignment seeks to align the entire graph to the other graph such that edges overlap as much as possible. Given graphs G and H, vertices of G are mapped to vertices of H such that as many edges overlap as possible. The GRAAL algorithm [KMM+10] starts with matching a pair of vertices and then building spheres of matchings around the seed pair. Since this algorithm explicitly aligns the graphs in real time, it is not possible to apply this algorithm to contact map alignment where sequences structure must be preserved. The Isorank graph alignment algorithm [LLB+09] calculates structural similarity between vertices of graphs before explicit alignment. This allows us to take advantage of this structure in sequence alignment as well.

We develop a contact map alignment algorithm that incorporates strategies from graph alignment and sequence alignment. Unlike previous approaches to this problem, our alignment uses that fact that the diagonal substructures of the contact maps should be aligned to each other, which corresponds to contiguous subsequences of the input amino acid sequences. Thus, we are able to extend our algorithm such that it can incorporate information regarding the actual sequence information and their alignments.

1.1 Background

Proteins can be represented as a sequence of three letter codes, each representing an amino acid. The binding sites of proteins are sections of the proteins that can come into contact with binding sites on other proteins. When two proteins bind at their binding sites after a series of biochemical events, this is called a protein-protein interaction (PPI). Under regular circumstances, a protein is able to interact freely with all relevant proteins and carry out human functions. However, virus proteins can also interact with human proteins which allows them to hijack the proteins and use them to replicate themselves. Thus, PPIs between virus and human proteins are an important area of research. With more complete knowledge of these interactions, vaccines can be made that target those proteins in particular and train them to efficiently identify and eliminate intruding molecules.

1.2 Motivation

We initially began with the problem of identifying new virus-host protein interactions, inspired by the study of the novel COVID-19 genomes. The data was collected as explained in Section 2.1 and a logistic regression model was created as shown in Section 2.2. Unfortunately, this model failed to extrapolate to novel proteins, as these manually curated databases were not useful in producing new, useful knowledge using such a simple model.

To solve this issue, we created our own algorithm to determine the similarity between two proteins, using amino acid sequences and contact maps as input. By doing so, we can have the ability to infer interactions between *novel* pairs of proteins and their motifs, which is information that is not available in the Eukaryotic Linear Motif database [KGM+20] or ELM.

1.3 Problem Statement

The goal of aligning contact maps is to determine protein similarity. The Contact Map Overlap (CMO) problem is a heuristic for scoring contact map alignments. We present a modified version of the CMO problem that includes sequence information as well. We hypothesize that combining these two modalities of data better captures evolutionarily-conserved regions of proteins, and thus allows better prediction of novel PPIs.

1.3.1 The sequence-informed contact map overlap problem (SCMO)

Given protein amino acid sequences $a = a_1, ..., a_m \in \Sigma^m, b = b_1, ..., b_n \in \Sigma^n$ where Σ is the set of amino acids and contact maps $A \in \{0,1\}^{m \times m}, B \in \{0,1\}^{n \times n}$ of a, b respectively, then $A_{u,v} = 1$ if and only if amino acids a_u and a_v are in contact with each other in the 3D structure of the protein. An alignment of a, b is a pair of monotonic injective functions $f_a : [m] \to \mathbb{Z}$ and $f_b : [n] \to \mathbb{Z}$. From f_a and f_b, a pair of functions $g_a, g_b[l] \to \mathbb{Z} \cup -$ are constructed where $g_a(f_a(i)) = i$ for all $i \in [n]$ and $g_a(i) = -$ if i is not in the codomain of f_a. g_b is constructed similarly. The $-$ elements represent gaps within the sequence. l is the largest value of the codomains of f_a, f_b. Amino acids a_i at position i is aligned to b_j at position j are aligned if $g_a(k) = i$ and $g_b(k) = j$ for some k. The entries $A_{u,x}$ and $B_{v,y}$ are aligned to each other in the contact map alignment if a_u, b_v are aligned and a_x, b_y are aligned. We wish to find the alignment that maximizes:

$$\begin{aligned} &\sum_{i=1}^{l}\sum_{j=1}^{l} A_{a_{g_a(i)},a_{g_a(j)}} B_{b_{g_b(i)},b_{g_b(j)}} - \sum_{i=2}^{m} K(f_a(i) - f_a(i-1) - 1) \\ &- \sum_{j=2}^{n} K(f_b(j) - f_b(j-1) - 1) + c\sum_{i=1}^{l} \text{BLOSUM}(a_{g_a(i)}, b_{g_b(i)}) \end{aligned} \tag{1}$$

where δ denotes the Kronecker delta function, as a function over Σ^2 or over $\mathbb{Z}^2$:

$$\delta(\sigma, \tau) = \begin{cases} 1 & \text{if } \sigma = \tau \\ 0 & \text{else.} \end{cases}$$

The first part of the sum counts the number of aligned contacts. The function f_a can be visualized as adding gaps of length $f_b(j) - f_b(j-1) - 1$ between amino acids a_j, a_{j-1}. The next 2 terms of the sum penalize gaps. The affine gap penalties are calculated by the function $K : \mathbb{Z}_{\geq 0} \to \mathbb{R}$. On the positive integers, K is a linear function $a(x-1) + b$ that outputs the penalty of a gap of size x. $K(0) = 0$ since gaps of size 0 are not penalized. b is the cost of "opening" a gap and a is the cost of "extending" the gap. The idea is that multiple small gaps are penalized more than one large gap. The final term iterates through aligned amino acids and adds their similarity scores. We use the BLOSUM matrix blosum,

explained in Section 2.5, as a heuristic for amino acid similarity. The constant c adjusts how much sequence information should be used in relation to the structural information in the contact map.

1.4 Our Contribution and Related Works

Standard protein comparison in generally done by comparing and scoring protein sequences by following the Needleman-Wunsch [NW70] or Smith-Waterman [SW+81] algorithm. This approach efficiently and accurately finds the character difference between two strings and the alignment that best matches them to each other. These methods have also been shown to have a nice probabilistic interpretation, via statistics coming from the theory of random sequences [SW+81].

To account for the added complexity of comparing both strings and exponentially growing contact maps, our algorithm identifies sections of the protein that is most relevant in creating the shape of the binding site and pulls out a 20 by 20 matrix to represent its contact map (the whole contact map could not be used for the sake of efficiency and runtime), along with the amino acid sequence paired with that segment. The contact map is then converted into a graph form as explained in Section 2.3 and, using dynamic programming and global alignment, the optimal alignment is found. The algorithm then outputs that alignment in the form of the amino acid sequences with gaps inserted via dashed lines.

The protein alignment algorithm iWRAP [HXBB11] classifies proteins and creates templates from said proteins to use as guidance for alignment. Subsequent proteins are then matched to templates and use the algorithm from [PBB08] to create the final output. The Biopython library [CAC+09] allows us to interpret information from the Protein Data Bank [KXdlC+06] and get the distances between each of the amino acids in the physical structure of any protein. This information can then be used to create contact maps for each of the proteins, and also get the amino acids mapped to each index. IsoRank is an alignment algorithm between global protein protein interaction (PPI) networks. The algorithm converts these networks into graphs and uses equations to score them in regards to similarity. Our algorithm used this idea of graph alignment and turns it into a contact map problem by converting contact maps into graphs as will be described in 2.3.

The SCMO problem is built on two previously studied problems:

1.4.1 The Contact Map Overlap (CMO)

The CMO problem is usually formalized as a comparison between two matrices. Just as sequence alignment is stated as an optimization problem to compute a distance metric between two strings a and b, one defines a distance metric between pairs of binary matrices A and B. By thinking of matrix-to-matrix alignments as pairs of matrices (A', B') containing A, B respectively as submatrices (and placeholder gap characters everywhere else), the corresponding score of (A', B') is

$$\sum_{i,j} \sum_{k,\ell} A_{i,j} B_{k,\ell}$$

where any numbers multiplied by the placeholder character is 0. The CMAPi [PBB08] algorithm is a specific realization of the above model. Unlike in our algorithm, CMAPi directly aligns the Contact Maps with a 4D dynamic programming algorithm. The matrices A', B' are no longer symmetric which disrupts the relation between contact map and sequence. Thus, the CMAPi algorithm is unable to consider sequence information.

1.4.2 Graph Alignment Problem

The CMO problem is similar to the graph alignment problem in graph theory [Döp13].

Given graphs G, H with vertices $V = v_1, ..., v_m$ and $W = w_1, ..., w_n$ respectively, let $w^G_{i,j}$ represent the edge weight between v_i, v_j and $w^H_{i,j}$ be the edge weight between u_i, u_j.

The alignment of these graphs is an injective function $f : V \to W$. We wish to find the alignment that maximizes:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} w^G_{i,j} w^H_{f(i),f(j)}$$

This problem has been studied in works such as GRAAL [KMM+10] and the Isorank Graph Alignment algorithm [LLB+09].

2 Methods

2.1 Our Data

The data came from four main databases. The first of which is the the National Center for Biotechnology Information (NCBI) [SAB+19]. This database gave us access to lab-submitted protein sequences which we were then able to enter into the Eukaryotic Linear Motif or ELM [KGM+20] database.

ELM compares the given sequence to a list of well-documented proteins and returns the most similar ones in terms of direct sequence motif matches. Similarity scores of the entered protein sequence to the returned proteins are also given. Under each returned protein is a list of direct motif matches. Each motif interacts with a list of other proteins; these proteins can be found from the STRING database.

STRING [SMC+16] uses the name of the protein motif as input and returns the most interactive proteins with that motif, organized by score. The score is calculated independently by STRING and is meant to represent the likelihood of interaction between the given protein and the returned one. Each interactive protein may occur multiple times under the same ELM-returned protein. Data regarding the number of times each interactive protein appeared under the same ELM protein was also kept track of within our models. The order of hierarchy within the data set is further explained in Fig. 1 to clarify the relationship between all data points used.

Further protein information was found on the Protein Data Bank (PDB) [KXdlC+06] website. Here, we were able to organize the structural information of the proteins. This database has a large collection of useful information. But, for our purposes, we collected the distances between amino acids in the protein sequence as reference to create the contact maps for each protein. Each protein has a sequence of at most 200 amino acids so the complete contact map was too large for the scale of our algorithm. For this reason, we only used the center 20×20 matrix of the protein's contact map over the D and M chain. This section was specifically chosen because it marks the portion of the protein that is in closest contact while also being in the center of the protein, meaning it stores the most relevant information in regards to the binding site of the protein as well.

2.2 Logistic Regression

Our goal was to use a model that to predict yes/no binary answers indicating whether an input protein interacts with another input protein. We used logistic regression, because it is the simplest model that one can try to fit for such a scenario. Each protein corresponds to a point $\mathbf{x} \in \mathbb{R}^n$; in our case, $n = 4$ because we used four different features. The method tries to fit the model

$$f(\mathbf{x}; \mathbf{b}) = \frac{1}{1 + \exp(b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + b_4 x_4)} \tag{2}$$

by minimizing a certain loss function $\mathcal{L}$ evaluated on the training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$

$$\mathcal{L}(\mathbf{b}) = \frac{1}{N}\sum_{i=1}^{N} y_i f(\mathbf{x}_i; \mathbf{b}_i) + (1 - y_i)(1 - f(\mathbf{x}_i; \mathbf{b}_i))$$

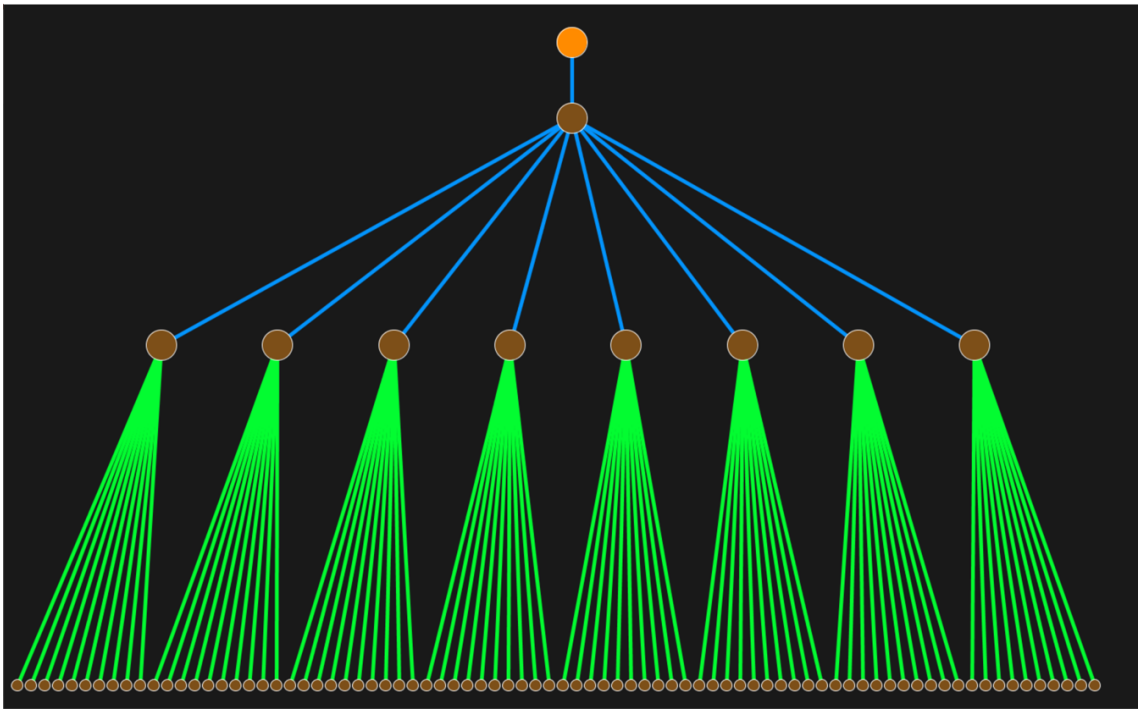


Figure 1: This is a graph visualization to clarify the order of hierarchy within the data we collected from all the databases. The top orange node represents the NCBI protein data. Once that sequence is entered into ELM, multiple similar proteins are given. In the case of this graph only one of those similar proteins is shown (the brown node on the second level from the top) such that the data displayed is at a manageable size. In reality, we collected data for the top five most similar proteins returned according to ELM. Beneath the ELM protein are the direct motif matches within it. Each brown node on the third level from the top represents one such motif. Lastly, the bottom most layer of brown nodes represents the top ten most interactive proteins with each motif according to the STRING database. The blue edges further clarity by denoting proteins that are similar, while green edges connect proteins that interact with each other.

over all possible choices of coefficients $\mathbf{b} = (b_0, b_1, b_2, b_3, b_4)$. This is typically solved using gradient descent, which we did as well using standard packages built into scikit-learn.

The features we chose for x_1 through x_4 were as follows. The value x_1 is the similarity score between the virus protein and the current protein taken from ELM. The value x_2 is the number of direct motif matches between the virus sequence and the current protein sequence, also calculated based off of the output of ELM. The interaction score of each of the ten proteins under the current motif is represented by x_3. The number of occurrences of each interactive protein under the overall ELM protein is saved in x_4.

As mentioned earlier in Section 1.2, the results of the logistic regression model based on our compiled data was not informative. Some data points were inconsistent, having all equivalent independent variable, yet lab-tested results from [SMC+16] showed that only part of these data points had confirmed interactions with the goal protein, while others did not. For our data specifically, 5289 of 45211 data points (11.7 percent) were the same proteins that had been lab-tested as confirmed interactions. The resulting model was able to predict the correct form of interaction 89.58 percent of the time, which is near equivalent to the percentage of non-interactions within the data. This similarity between the model's ability to predict the correct interaction and the percentage of non-interacting proteins in the data lead us to believe that the model was near equivalent to random guessing when it came to predicting the interaction. Had the model returned 0 as in non-interactive for all proteins, the percentage of correct predictions would still be about 88.3 percent. This

proves that the equation generated that was used to create logistic regression model had determined that all independent variables given as input were not informative in terms of determining the interaction probability of two proteins.

2.3 Contact Map Alignment

The goal of this algorithm is to find an alignment that optimizes the score defined in section 1.3.1. The first part of our algorithm draws from the Isorank Graph Alignment algorithm [LLB+09]. The Isorank algorithm aligns pairs of graphs G and H by calculating structural similarity scores between vertices of graphs. This information is encoded in a particular matrix R, which is then passed onto a simple greedy heuristic for alignment.

We adapted this algorithm for use on protein contact maps. First, we interpret protein contact maps as graphs. From an $m \times m$ contact map named M, we can construct a graph called G. The graph contains m vertices, each labelled $v_1, ..., v_m$. v_i and v_j on the graph have an edge between them if the space in $M_(i, j)$ contains a 1. If $M_(i, j)$ contains a 0, there is no edge between v_i and v_j. The matrix M is the adjacency matrix of G. Similarly, for an $n \times n$ contact map N, we construct the graph H such that the N is the adjacency matrix of H. The graphs are then aligned using a combined version of the Isorank and Needleman-Wunsch algorithm (Eq. (1)).

The main idea of our algorithm is to compute structural similarity scores between vertices of the graph, and instead of directly matching vertices, transform the R matrix into a payoff matrix containing sequence information and then aligning the sequences using Needleman-Wunsch.

2.4 R Matrix

In this section, we describe the algorithm to compute the R matrix. Let G be a graph with vertices $v_1, ..., v_m$ while H is a graph with vertices $w_1, ..., w_n$ where $w^G_{i,j}$ is the edge weight between vertices v_i and v_j in graph G. Similarly, $w^H_{i,j}$ denotes the edge weight between vertices w_i and w_j in graph H. G and H are the graphs corresponding to 2 inputted contact maps. Denote $G_{i,i}$ and $H_{i,i}$ to be 0. Let $deg(v_i)$ to be the sum of edge weights adjacent to v_i. In the style of [LLB+09], the values of R are indexed by pairs a, x where v_a and w_x are vertices of G and H respectively. The values of R satisfy the following equations:

$$R_{a,x} = \sum_{b=1}^{m} \sum_{y=1}^{n} R_{b,y} \frac{w^G_{a,b} w^H_{x,y}}{deg(v_b) deg(w_y)}$$

Intuitively, the value of $R_{a,x}$ is a heuristic for the similarity between v_a and w_x. Thus, $R_{a,x}$ increases the more likely v_a and w_x should be aligned. Note that $R_{a,x}$ and $R_{b,y}$ are positively correlated in proportion to $w^G_{a,b} w^H_{x,y}$. The intuition is if $w^G_{a,b}$ and $w^H_{x,y}$ are non-zero, then if v_b, w_y are aligned, and $v_a v_b$ and $w_x w_y$ are edges, since the algorithm wants edges $v_a v_b$ and $w_x w_y$ to overlap, then v_a, w_x are also more likely to be aligned to each other.

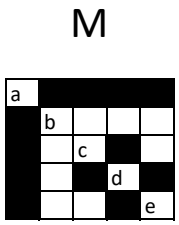
The way R is defined makes it easy to calculate. Define the matrix A to be as follows: The rows and columns of A are indexed by pairs a, x where v_a and w_x are in G and H respectively and

$$A_{(a,x),(b,y)} = \frac{w^G_{a,b} w^H_{x,y}}{deg(v_b) deg(w_y)},$$

Note that $R = AR$.

In [LLB+09] it is proven that R is the principle eigenvector of A. Thus, we can calculate R using the power method. We first initialize R_0 to be a length mn vector of all 1's. Then, we define

$$R_{i+1} = AR_i$$



R

	a	b	c	d	e
v	0.538	0.134	0.269	0.403	0.269
w	0.134	0.033	0.067	0.100	0.067
x	0.134	0.033	0.067	0.100	0.067
y	0.269	0.067	0.134	0.201	0.134
z	0.269	0.067	0.134	0.201	0.134

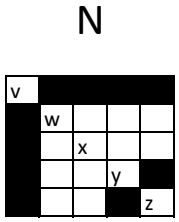


Figure 2: Example Contact Maps M and N with resulting R matrix. Black squares in the figure represent contacts. Example Equations:

$$R_{aw} = \frac{R_{bv}}{4} + \frac{R_{cv}}{8} + \frac{R_{dv}}{12} + \frac{R_{ev}}{8}$$

$$R_{ay} = \frac{R_{bv}}{4} + \frac{R_{cv}}{8} + \frac{R_{dv}}{12} + \frac{R_{ev}}{8} + \frac{R_{bz}}{2} + \frac{R_{cz}}{4} + \frac{R_{dz}}{6} + \frac{R_{ez}}{4}$$

$$R_{by} = \frac{R_{av}}{16} + \frac{R_{az}}{8}$$

$$R_{bw} = \frac{R_{av}}{16}$$

for all i. Now we calculate R_i in order until we reach a value of i where

$$|R_i - R_{i+1}| < 0.01$$

Thus, since $R_i \approx AR_i$, R_i approximates an eigenvector of A. It is a well known fact of linear algebra that R_i should approximate the principle eigenvector of A. Then we set R to be R_i. Finally, we reshape R into a $m \times n$ matrix where the rows are indexed by vertices in G and columns indexed by vertices in H. The intuition behind the R matrix is that $R_{i,j}$ represents the structural similarity score between $v_i \in G$ and $w_j \in H$. Fig. 2 shows an example of the R matrix for 2 contact maps.

2.5 Sequence Alignment

The BLOSUM Matrix [HH92] is a 20×20 symmetric matrix where rows and columns are indexed by the 20 amino acids. For amino acids x, y, $BLOSUM_{x,y}$ is the normalized log likelihood that a certain amino acids x could be randomly swapped with y. The BLOSUM matrix represents a heuristic for amino acids similarity.

The R' matrix is constructed as follows for all $v_a \in G$ and $w_b \in H$:

$$R'_{a,x} = R_{a,x} + c \cdot \text{BLOSUM}_{v_a, w_x}$$

for some constant c. Thus, R' contains both sequential information on the amino acid

types and structural information of the structures. The constant c controls how much of structural or sequence information should be used in the alignment.

Next, we performed Needleman-Wunsch [NW70] where instead of a binary 0/1 scoring scheme, the R' matrix defines the score. The Needleman-Wunsch algorithm is a modified dynamic programming algorithm with an affine gap penalty. Let $a = a_1, ..., a_m$ and $b = b_1, ..., b_n$ be sequences of two proteins. An alignment of a, b is a pair of injective functions where $f_a : a \to \mathbb{Z}$ and $f_b : b \to \mathbb{Z}$. The Needleman-Wunsch algorithm uses dynamic programing to find the alignment that maximizes:

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\delta(f_a(i), f_b(j))R'(i,j) - \sum_{i=2}^{m}K(f_a(i) - f_a(i-1) - 1) - \sum_{j=2}^{n}K(f_b(j) - f_b(j-1) - 1) \quad (3)$$

where R' is the payoff matrix. δ is again taken to be the Kronecker delta function. K is the Affine gap penalty function where the input is the length of a gap in the alignment. Consideration of the Affine gap penalties are built into the algorithm.

This completes our alignment algorithm. Note that since R' contains structural information, the algorithm also incorporates structural information. The value of c and the affine gap penalties can be adjusted to optimal values.

Algorithm 1 Contact Map Alignment

for $i = 1$ to m **do**
    for $j = 1$ to m **do**
        $G[i][j] \leftarrow$ edge weight between v_i, v_j in G
    $deg(v_i) \leftarrow$ degree of v_i
for $i = 1$ to n **do**
    for $j = 1$ to n **do**
        $H[i][j] \leftarrow$ edge weight between w_i, w_j in H
    $deg(w_i) \leftarrow$ degree of w_i
for $a = 1$ to m **do**
    for $x = 1$ to n **do**
        for $b = 1$ to m **do**
            for $y = 1$ to n **do**
                $A[a * m + x][b * m + y] \leftarrow \frac{G_{a,b}H_{x,y}}{deg(v_b)deg(w_y)}$
$R[0] \leftarrow$ random vector length mn
$R[1] \leftarrow AR$
i=1
while $|R[i] - R[i-1] < 0|$ **do** $R[i+1] = AR[i]$
for $i = 1$ to m **do**
    for $j = 1$ to n **do**
        $R'[i][j] \leftarrow R[i+1][i * m + j]$
        $R'[i][j] += BLOSUM[v_i][w_j]$

R' is our payoff matrix. We then perform Needleman-Wunsch on this payoff matrix to align the contact maps.

3 Results

Each protein is converted into a binary matrix representing the contact map. To decrease the dimensions of the matrix, only the middle 20×20 section of the contact map is used to represent the overall physical structure of the protein. Two such contact maps are

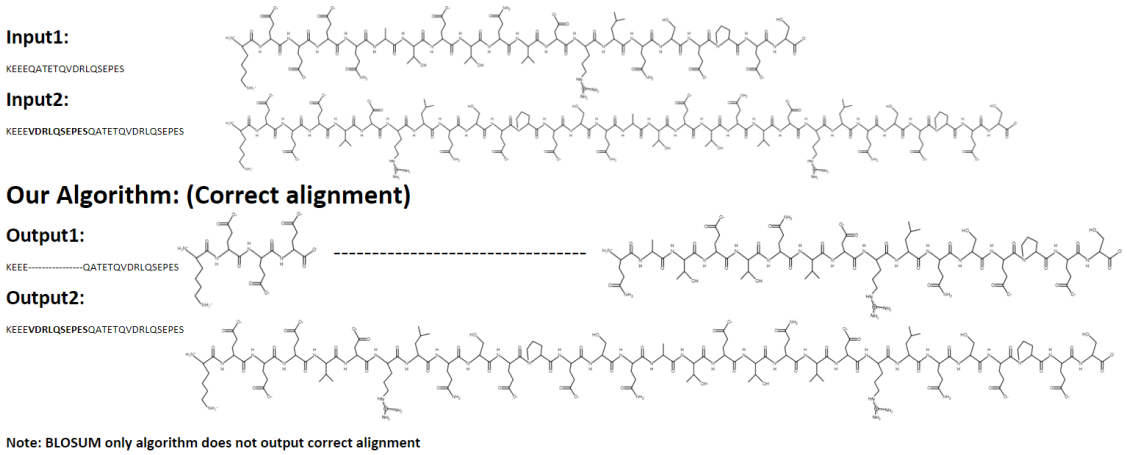


Figure 3: The top two input sequences are the proteins given to the algorithm as input. On the right are the peptide chains in their visual appearance. Input 1 is the sequence from the protein 1xi4 taken from PDB as described in section 2.1. Input 2 is the modified sequence, almost identical to Input 1 but with a group of 10 randomly chosen amino acids inserted after the fourth amino acid. This controlled test is used to see if our algorithm can output the correct alignment with a gap of 10 inserted after the fourth amino acid of Input 1, as aligned to Input 2. The output, seen underneath the inputs, is correct, with a gap of exactly 10 amino acids placed after the fourth amino acid of Input 1 to reach the optimal alignment.

given as input. The algorithm then aligns the two by finding the optimal orientation that maximizes the quantity of contact matched to each other. This algorithm compares two proteins to score the similarity between them while accounting for both physical and sequential patterns. The output of the algorithm is the sequence of a selected protein with gaps inserted to indicate the best alignment when compared to the other protein.

To test the algorithm, one controlled case was compared to the original sequence and contact map Fig. 4 of a protein. This altered version included a gap of 10 randomly selected amino acids in a random section of the sequence (the contact map Fig. 5 was also altered to match the new sequence, with random contacts inputted in the added rows and columns). When compared to the original protein with the original contact map, our algorithm correctly output the best alignment by inserting a gap of 10 in the original protein's sequence to perfectly match the amino acids with the altered sequence as shown in Fig. 3.

Although the alignment step of our algorithm uses Needleman-Wunsch, which only considers local information, since the payoff matrix already incorporates global structure information from the contact map, the dynamic programming alignment also accounts for the global structure. After testing, we concluded that our algorithm does meet the goals we expect. The scoring and alignment are both more suitably representing the interaction style of the given proteins when compared to each other.

4 Conclusion and Future Work

The goal of this paper was to try to create an algorithm to compare protein similarity using structural information. We did this in order to address a larger challenging problem: it is intrinsically difficult to identify novel protein-protein interactions using only manually-curated interaction and motif databases such as ELM and STRING. Our method is one

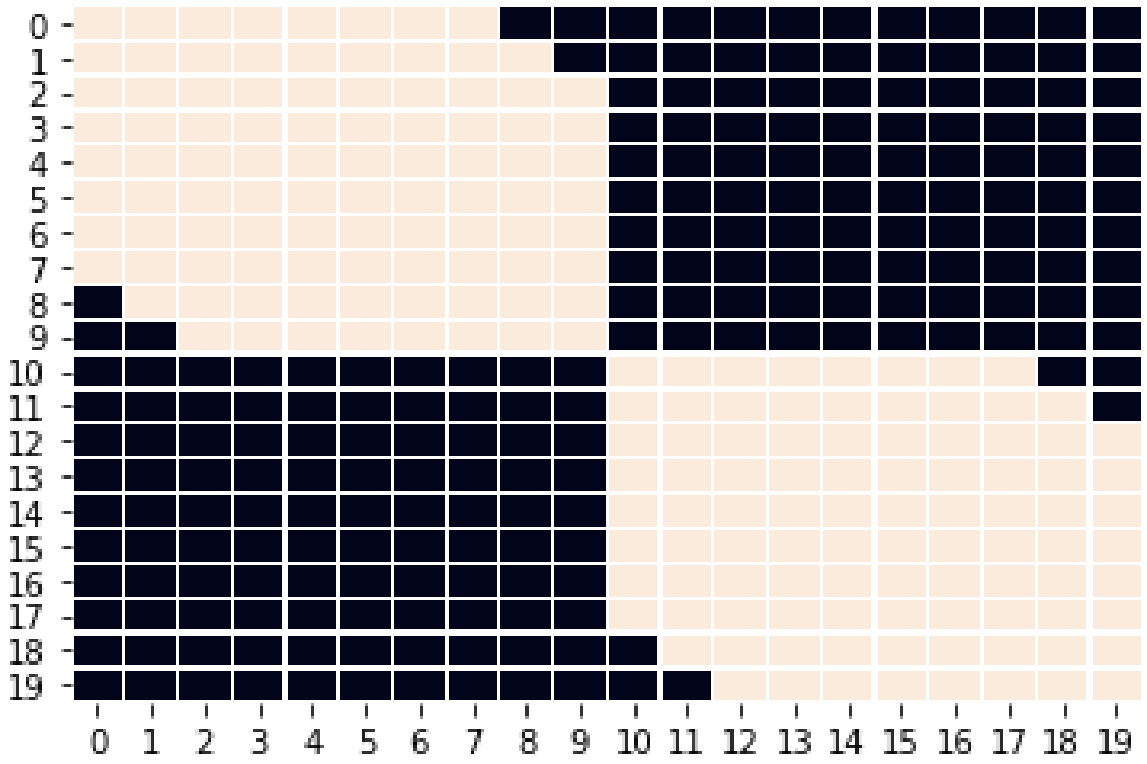


Figure 4: The contact map matching the sequence given for the protein 1xi4 in input 1.

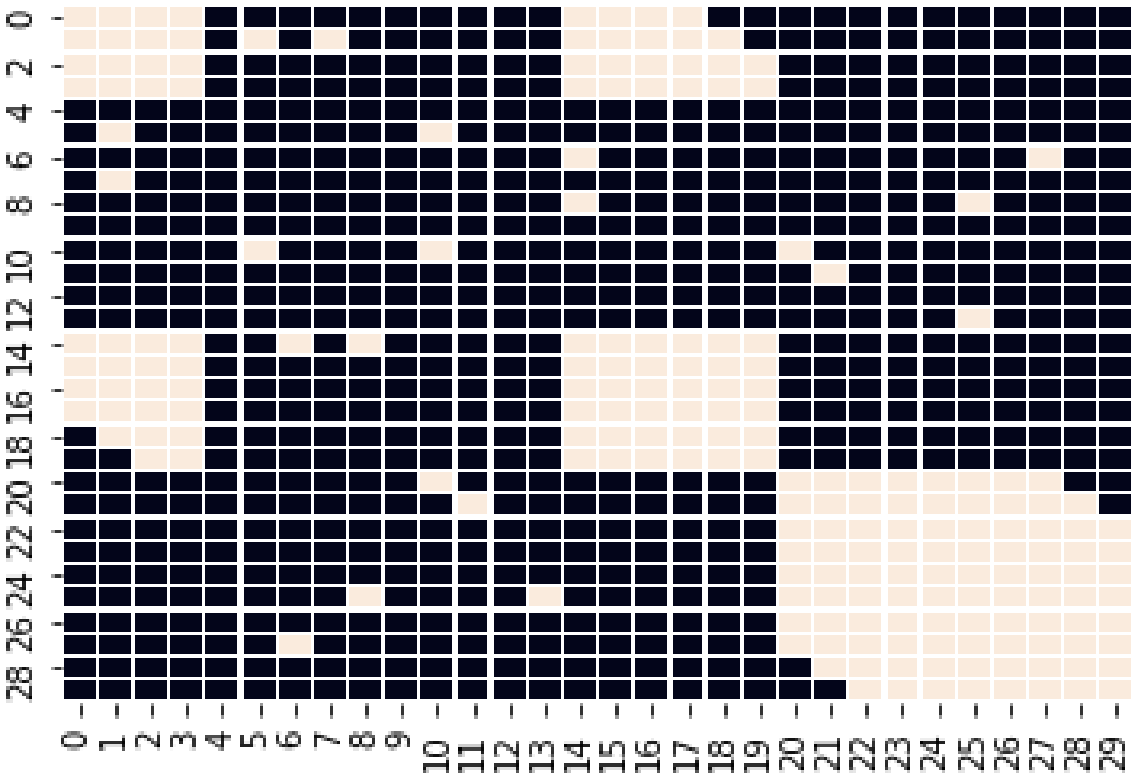


Figure 5: The contact map matching the sequence given in Input 2. Similar to that of Input 1 but with a stretch of 10 random amino acids inserted at index 4. The distances used to create the contact map for those new columns were also randomly generated.

attempt in incorporating these two modes of data. Currently, it faces some challenges: our algorithm does not scale well in the size of its inputs, because the computation of the R matrix necessarily computes an eigenvector of an $nm \times nm$ matrix, resulting in an $O(n^3m^3)$ algorithm. We leave it to further exploration to study different scoring schemes or alternative formulations of the problem which admit more efficient solutions, and to incorporate such an alignment scheme back into the original problem of detecting novel virus-host protein interactions.

5 Acknowledgements

We would like to thank the MIT PRIMES program for providing this research opportunity. We thank Younhun Kim, our mentor, for giving his time and guidance to help us throughout this project, and Dr. Tanya Khovanova and Dr. Alexander Vitanov for reviewing the paper draft.

References

[CAC+09] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009.

[Döp13] Christoph Döpmann. Survey on the graph alignment problem and a benchmark of suitable algorithms. 2013.

[HH92] Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

[HXBB11] Raghavendra Hosur, Jinbo Xu, Jadwiga Bienkowska, and Bonnie Berger. iwrap: an interface threading approach with application to prediction of cancer-related protein–protein interactions. *Journal of molecular biology*, 405(5):1295–1310, 2011.

[KGM+20] Manjeet Kumar, Marc Gouw, Sushama Michael, Hugo Sámano-Sánchez, Rita Pancsa, Juliana Glavina, Athina Diakogianni, Jesús Alvarado Valverde, Dayana Bukirova, Jelena Čalyševa, et al. Elm—the eukaryotic linear motif resource in 2020. *Nucleic Acids Research*, 48(D1):D296–D306, 2020.

[KMM+10] Oleksii Kuchaiev, Tijana Milenković, Vesna Memišević, Wayne Hayes, and Nataša Pržulj. Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*, 7(50):1341–1354, 2010.

[KXdlC+06] Andrei Kouranov, Lei Xie, Joanna de la Cruz, Li Chen, John Westbrook, Philip E Bourne, and Helen M Berman. The rcsb pdb information portal for structural genomics. *Nucleic acids research*, 34(suppl_1):D302–D305, 2006.

[LLB+09] Chung-Shou Liao, Kanghao Lu, Michael Baym, Rohit Singh, and Bonnie Berger. Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12):i253–i258, 2009.

[NW70] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[PBB08] Vinay Pulim, Bonnie Berger, and Jadwiga Bienkowska. Optimal contact map alignment of protein-protein interfaces. *Bioinformatics (Oxford, England)*, 24:2324–8, 09 2008.

[SAB+19] Eric W Sayers, Richa Agarwala, Evan E Bolton, J Rodney Brister, Kathi Canese, Karen Clark, Ryan Connor, Nicolas Fiorini, Kathryn Funk, Timothy Hefferon, et al. Database resources of the national center for biotechnology information. *Nucleic acids research*, 47(Database issue):D23, 2019.

[SMC+16] Damian Szklarczyk, John H Morris, Helen Cook, Michael Kuhn, Stefan Wyder, Milan Simonovic, Alberto Santos, Nadezhda T Doncheva, Alexander Roth, Peer Bork, et al. The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic acids research*, page gkw937, 2016.

[SW+81] Temple F Smith, Michael S Waterman, et al. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.