EFFICIENT ALGORITHM FOR SELECTING PROTEIN RESIDUE-RESIDUE CONTACTS

BY
QING YE

THESIS
Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science in the Graduate College of the University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Adviser:
Assistant Professor Jian Peng
ABSTRACT

The functions of proteins are largely determined by their structures. Determination of the protein three-dimensional structure is experimentally and computationally challenging.

Since amino acids residues that are spatially close often co-evolve, the correlation allows us to predict the contacts from multiple sequence alignments. The predicted contacts can then be used as spatial constraints and offer guidance in protein structure prediction. The constraints can be used as inputs to a protein structure prediction algorithm to produce “decoy” models as tentative 3D structures for proteins. However, the computation power required for structural prediction grows exponentially with respect to the number of contacts selected. Thus selecting few and yet informative contacts are essential for producing high-quality models quickly.

Existing contact prediction methods aim for improving precision and recall. However, not all contacts offer the same level of structural information in terms of structure prediction. Therefore, the strategy to select contacts of highest confidence may not be ideal for structure prediction. Here we present an efficient algorithm, ContactSel, to select contacts for assisting contact-guided ab inito folding. We take the key idea that contacts that involve residues far apart (long-ranged) and collections of contacts that are most diverse contains more information than contacts that are shorter ranged and closed by. We formulate the contact selection problem into an integer programming algorithm to select structurally diverse contacts.

For evaluation, we generated decoy models using $L/2$ contacts selected by ContactSel and a naive selection baseline. We show that we achieved significant improvement on the CASP 12 domain set.
To my parents, for their love and support.
# TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION ......................................................... 1
  1.1 Background ................................................................. 1
  1.2 Structure Prediction ..................................................... 2
  1.3 Contacts Prediction ...................................................... 2
  1.4 Motivation ................................................................. 3
  1.5 Contribution of this Thesis ............................................ 4

CHAPTER 2 RELATED WORKS ..................................................... 7
  2.1 Jackhmmer ................................................................. 7
  2.2 CCMPred ................................................................. 7
  2.3 DeepContact ............................................................ 8
  2.4 Confold ................................................................. 9
  2.5 Integer Programming ................................................... 10

CHAPTER 3 RANGE AND DIVERSITY BASED CONTACT SELECTION ........ 12
  3.1 Problem Definition ..................................................... 12
  3.2 Pre-processing .......................................................... 12
  3.3 The Integer Programming Problem .................................. 13
  3.4 Algorithm ............................................................... 16

CHAPTER 4 EXPERIMENTS ........................................................... 19
  4.1 Overview of the Pipeline .............................................. 19
  4.2 Dataset ................................................................. 20
  4.3 Protein Contact Prediction .......................................... 20
  4.4 ContactSel Implementation and Settings .......................... 22
  4.5 Protein Folding ....................................................... 22
  4.6 Baseline and Evaluation ............................................. 23

CHAPTER 5 RESULT ................................................................. 24
  5.1 Visualization of the Contact Selection ............................ 24
  5.2 Performance Comparison ............................................ 25
  5.3 Computational Efficiency ............................................ 30

CHAPTER 6 CONCLUSION AND FUTURE WORK ............................... 32

REFERENCES ........................................................................... 33
CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

Proteins are complex molecular devices formed by amino acid polymer chains. These chains fold into specific conformation for proteins to perform their biological functions. Determining the three-dimensional structure of a protein is the first step of understanding such functions and gaining insights into many biomedical problems such as drug design and enzyme design. Traditionally, four levels of protein structures are considered: the primary structure refers to a 1-D sequence of the amino acid polypeptide chain; the secondary structure refers to local sub-structure on the backbone chain with the two main type, α-helix and β-strands; the tertiary structure refers to the 3-D structure folded by a single polypeptide chain; the Quaternary structure refers to the structure folded by multiple chains.

![Figure 1.1: Protein Secondary Structure](image)

The Primary Structure of a protein can be determined by a cheap and well-studied process, the Edman degradation [2] developed by Pehr Edman. The prediction of the secondary structure is also a largely solved problem by modern biological toolkit such as PSIPRED[3]. However, the determination of the structure still remains an important challenge in biology.

A protein structure is traditionally determined by experimental methods. However, the experimental determination of a protein structure requires extensive effort in sample preparation, data collection, and interpretation. Moreover, most methods often limit to narrow types of protein. For instance, X-ray crystallography, from which most structures in PDB[4] are determined, requires crystallization of protein samples, which has been generally unsuccessful among membrane proteins. On the other hand, nuclear magnetic resonance (NMR), another popular technique that can solve protein structure in atomic detail, restricts to relatively small proteins. The latter method also suffers from weeks of instrument time and laborious manual assignment of the NMR spectra.
The structure of a protein is almost solely decided by its 1-D amino acid sequence. And with the emergent computational power of modern hardware, we can hope to solve the problem computationally which leads into the domain of protein structure prediction.

1.2 STRUCTURE PREDICTION

Given a sequence, while in some cases a homologous sequence may be found in the PDB Databank [4]. The homologous sequence can then be used as a template, and confirmation of the target sequence can be found by template-based structure prediction. However, when the homologies cannot be found, template-free (or ab initio) modeling is the only options available. An ab initio protein structure prediction algorithm attempts to solve the structure from a single amino acid sequence. Although many algorithms are available, most have three essential components: An accurate energy function to identify thermodynamically stable folds, an efficient search method through the confirmation space to find the low-energy states and a strategy to select decoy models most similar to the native structure [6].

However, such algorithms suffer from two limitations: the amount of computation power required to solve for the native fold is often prohibitive and that the predicted structure often lacks accuracy.

1.3 CONTACTS PREDICTION

Recently, there are multiple advancements in template-free modeling. One of which is
the introduction of protein contacts. Multiple definitions has been given for protein contacts and here we define protein contacts as residues on the amino acid chain that are less than 8 Å apart. The availability of large sequence banks enables us to utilize evolutionary information to assist the computational approach. As spatially close amino acids in a structure tend to co-evolve, coupling analysis can reverse engineering the 3D structure from such correlation. In Figure 1.2, we show an example of correlated positions in the sequence, which often implies a physical contact in the amino acid chain. Direct coupling analysis such as plmDCA[7], CCMPred[8], GREMLIN[9], and EVFold[10] can take a multiple sequence alignment (MSA) as input and predict a residue-residue contact map by learning a graphical model representation of the underlying MSA. Recent advancement in machine learning further improves the co-evolutionary-based contact prediction. DeepContact[11], a convolution neural network based approach infers the contact probability accurately by utilizing the powerful automatic feature selection of deep learning to discover co-evolutionary motives. Some structure prediction pipelines, including CONFOLD[12] and pcons-fold[13] take advantage of the predicted contacts to generate 3D constraints to assist protein structure prediction. The resulted structure with the assisting constraints is often more accurate. One example is shown in Figure 1.3.

However, the contact prediction algorithms are ranking methods that rank the predicted contacts by their confidence. Since the prediction algorithms are tuned for accurate prediction, simply supply the Top-K contacts as input for the downstream structural prediction programs may not be ideal. Short-range contacts and contacts that are spatially close often gives little information for structural prediction due to their redundancy. For instance, the highlighted contacts in Figure 1.4 offers little structural information and should have been avoided. We will discuss the insights in detail in the motivation section.

1.4 MOTIVATION

Here, we provide two key insights into selecting contacts to improve structure prediction.

**Long range contacts are more informative** : β-strands often forms hydrogen bonds into a sheet-like structure. These hydrogen bonds cause the β-strands to form long-range
Figure 1.3: Decoy Models for CASP Target T0915[14]. The red models are decoy model and the blue models are the referencing ground-truth model determined experimentally. The decoy model that use contacts as constraint information has a much closer conformation to the reference model than the one without constraint information.

contacts. However, the information is not provided by the secondary structure prediction and therefore is very informative. Meanwhile, short-range contacts, such as those in α-helices, are not as informative, as the sub-structure is already rigid so the information duplicates with that provided by secondary structure prediction.

Diverse contacts are more informative: Pairs of contacts that are similar usually offer less information than those are different. Many contacts predicted by a contact selection algorithm are often immediately next to each other. However, such a group of contact offers little information more than one the contact since the constraints are too closed by causing a confirmation satisfying one constraint automatically satisfy all constraints in the group.

1.5 CONTRIBUTION OF THIS THESIS

Here we present a novel algorithm, ContactSel, for automatic selection of contacts to improve contact-guided structure prediction. Taking contact maps generated by existing
Figure 1.4: Example of redundant protein contacts: The contact in alpha-helix makes little contribution as a folding algorithm would already use the secondary prediction information to pose constraints on the local conformation; The two contacts in the beta sheet is also redundant, as one is already sufficient for keep a constraint on the local confirmation.

contact prediction algorithms as inputs, our algorithm poses the contact selection problem as an integer program. By using an objective function that incorporates the contact range and the contact probability and a set of constraints that prevent spatially similar contacts to be selected, our algorithm balance the qualities and structural importance of contacts. Using a modern integer linear programming solver, we can obtain a solution efficiently, with a very small computation cost.

The contribution of the thesis is summarized as below

- Transform the motivations into heuristics and provide a mathematical formulation to reduce the contact selection problem into an optimization problem.

- Provide an algorithm, ContactSel, to transform the input contact map into an integer program
• Create a pipeline to perform contact based ab initio protein structure prediction and evaluate the contacts selected by ContactSel.

The rest of the thesis will be organized as follows: In Chapter 2, we discuss related works. The problem and the formulation of our method will be stated in Chapter 3. Chapter 4 describes our pipeline and the evaluation method in detail. In Chapter 5, we will review the result and make conclusions in Chapter 6.
CHAPTER 2: RELATED WORKS

2.1 JACKHMMER

A multiple sequence alignment (MSA) is a 1-D sequence alignment of multiple biological sequences. A tool specialized in MS takes a query sequence and a large sequence database as inputs. The tool needs to quickly search through the database for homologous sequence and make an alignment of the query sequence and the homologies as an output. The MSA is critical for a contact-based protein structure prediction pipeline as it provides context information for predicting the secondary structure and direct coupling information for predicting the contacts. We are especially concerned about the sensitivity of the MSA as the sequence of interest in template-free modeling usually have very few homologies as template-based modeling could be used otherwise.

Jackhmmer[15] is a multiple sequence alignment algorithm to find and align homologous sequences. The algorithm first search the sequence database with a standard BLAST[16] search. From there, a hidden Markov model is built and used for a new search. At each iteration, when the new homologous sequence is identified, it is added and aligned to the existing profile-HMM. The model is then rebuilt and the iteration continues until new homologous can be found.

Thanks to the profile-HMM and the iterative search strategy, Jackhmmer is very sensitive for detecting remote homologous sequences. We choose Jackhmmer as our MSA tool as the sequence of our interest only has few and remote homologous sequences in the sequence database.

2.2 CCMPRED

CCMPred[8] is a fast and efficient implementation of the pseudo-likelihood maximization based direct coupling analysis. This work is important to our thesis for two reasons. First, CCMPred is a classical state-of-the-art direct coupling analysis method for generating protein contacts. Second, it provides the most important feature for DeepContact[11], an algorithm we use to perform contact prediction in our pipeline.
By generalizing the Ising model into the Potts model, CCMPred formulates the generative process of an amino acid sequence in the MSA as:

\[ P(\sigma) = \frac{1}{Z} \exp \left( \sum_{i=1}^{N} h_i(\sigma_i) + \sum_{1 \leq i < j \leq N} J_{ij}(\sigma_i, \sigma_j) \right), \]  

(2.1)

Here, \( h_i \) and \( J_{ij} \) are the modeling parameters. \( \theta_i \) corresponding to the amino acid at the \( i \)-th residue. \( Z \) is a normalization constant. \( L \) is the length of the sequence and \( N \) is the size of the MSA. In principle, the direct coupling analysis find the modeling parameter by maximizing the maximum likelihood

\[ L(\Sigma) = \frac{1}{Z} \prod_{\sigma \in \Sigma} P(\sigma) \]  

(2.2)

Due to the normalizing term, the running time of the underlying optimization problem is exponential. CCMPred thus instead optimize a pseudo-likelihood. That is

\[ pll(h, J|\Sigma) = \sum_{n=1}^{N} \sum_{i=1}^{L} \left( h_i(\sigma_i) + \sum_{1 \leq i < j \leq N} J_{ij}(\sigma_i, \sigma_j) - \log Z_i^n \right) \]  

(2.3)

and,

\[ Z_i^n = \sum_{c=1}^{20} \left( h_i(\sigma_i) + \sum_{1 \leq i < j \leq N} J_{ij}(\sigma_i, \sigma_j) \right) \]  

(2.4)

In addition, CCMPred adds a \( L2 \) regularization term to favor sparse solution. The final coupling of the residues is computed by the Frobenius norm of the parameters and the average product correction is used to suppress insufficient sampling of the sequences. CCMPred utilizes GPU for computing and is usually fast in practice.

Unfortunately, due to the lack of sequences for ab initio prediction task, the output of CCMPred can be noisy and error-prone as the maximum likelihood inference is prone to overfitting in such cases.

2.3 DEEPCONTACT

DeepContact [11] is a novel Deep-Learning based framework for contact prediction. This supervised learning method uses a deep, convolutional neural network as a powerful feature
Figure 2.1: Features and neural network used by DeepContact[11]

extractor to discover the co-evolutionary motifs. However, instead of depending on the sequence level information along, it combines the direct coupling features with other important features, such as 1-D features obtained by aggregating the MSA by residues, global features about the MSA and other 2-D contact maps as shown in 2.1. Thanks to the superb feature extractor and ample features, the framework can provide reasonable performance, even when the MSA has few homologous sequences.

Another reason we choose the framework for contacts is that it provides a consistent probability-based score between 0 and 1 as the final confidence score of each contact. This enables an easier formulation of the ContactSelection Algorithm as we will see in Chapter 3.

2.4 CONFOLD

Confold is a state-of-the-art contact based structure prediction algorithm. It takes a secondary structure prediction and a contact map as inputs to produce decoy models. The pipeline of Confold is shown in Figure 2.2 Confold works in two stages.

In the first stage, secondary structure and the contacts are converted into physical and spatial constraints. The constraints are used to assemble a set of decoy models. In the second stage, the contacts of the decoy models are used to filter out the noisy predicted contacts. In addition, \( \beta \)-strands are paired to attempt to form \( \beta \)-sheets. Because of the second stage, Confold is quite tolerable for false positives in predicted contacts.

However, the second stage is not a silver bullet for ignoring the contact quality altogether. When the constraints given by the contacts have low quality, the initial decoy models will
not be in a conformation close enough to the native conformation. In this situation, the second stage cannot fully utilize the contact filtering and β-sheets pairing strategy.

2.5 INTEGER PROGRAMMING

An integer program is an optimization problem in the following canonical form:

\[
\begin{align*}
\max_x & \quad c^T x \\
\text{subject to} & \quad A x \leq b \\
& \quad x \geq 0 \\
\text{and} & \quad x \in \mathbb{Z}^n
\end{align*}
\] (2.5)

In general, integer programming is NP-Hard as the minimum vertex cover problem can be reduced to an integer program. However, when the objective function and the constraints
are linear, heuristics can be used to obtain an approximate solution.

CPLEX\cite{17} is an optimization package and provides an excellent Mixed Integer Programming solver. When the program is convex, CPLEX can often result in a solution quickly.
CHAPTER 3: RANGE AND DIVERSITY BASED CONTACT SELECTION

3.1 PROBLEM DEFINITION

For an amino-acid chain of $L$ residues. Predicted protein contacts $\mathbf{C} = \{c_1, c_2, \ldots\}$ can be represented a list of 3-tuple $c_i = (r_i, r'_i, p_i)$ where $r_i$ and $r'_i$ are the indices of residues that are predicted to be in contact with and $p_i$ is the confidence score of the contact. Without loss of generalization, let us set that $r_i < r'_i$. For most contact prediction programs, $r_i$ and $r'_i$ is simply a enumeration over all combinations of the residues, so that $|\mathbf{C}| = \binom{L}{2}$

A contact selection algorithm selects a subset of contacts $\mathbf{S} \subset \mathbf{C}$. The size of $\mathbf{S}$ needs to be small enough (usually in the scale of $\Theta(L)$) and another goal of the selection is to choose $\mathbf{S}$ to be informative enough so that the decoy models can have high quality.

Here, the key strategy we use is to pose the contact selection problem as an integer programming problem. We will transform various ideas in our motivation into objective function and constraints. Since integer programming problems are in general NP-hard, we will only choose linear objective functions and constraints to keep the program convex so that the optimizer can result in a quick approximated solution.

3.2 PRE-PROCESSING

$\alpha$-helices always produce a signature contact every 4-10 residues as a helix has 3.6 residues per turn. However, as contact-guided protein folding algorithm takes protein secondary structures predicted by state-of-the-art algorithms[12][13], the short-range contacts only provide redundant information. On the other hand, long-range contacts are important for forming protein structures, especially for $\beta$ class and $\alpha + \beta$ class of proteins[18][19]. For this reason, we filtered out all short and medium range contacts. The filtering also limits the number of decision variables in our integer programming formulation. The filtered contacts are used the input $\mathbf{C}$ to our integer programming algorithm.
3.3 THE INTEGER PROGRAMMING PROBLEM

First we introduce a binary variable $X = (x_1, x_2, ... x_{|C|})^T$ as the variable to be optimized in the integer programming problem. Here, $x_i$ represents a binary decision whether $c_i$ is selected for the downstream structure prediction algorithm, where $x_i = 1$ represents $c_i$ is selected and $x_i = 0$ represents the variable is not selected. That is,

$$S = \{c_j | x_j = 1\}.$$

3.3.1 Objective Function

We will now introduce two quality scores as components of the objective function to be maximized.
Contact quality score

The confidence score $p_i$ given by the contact selection algorithms is an important feature for the contact. Wrongly predicted contacts will lead to false spatial constraints and produce a poor result. To encourage high-quality contacts to be selected, we introduce the contact quality score

$$S_{\text{quality}} = \sum_{i \in 1..|C|} p_i x_i.$$  \hspace{1cm} (3.1)

Maximizing the contact quality score promotes high-quality contacts to be selected as shown in Figure 3.1 Note that without further objective function terms, the integer program will select contacts with highest possible confidence, which is the strategy used by conventional contact based structural prediction pipeline.

Contact range score

To encourage long-range contacts, as we discussed in the motivation part, we add a linear term in the objective function

$$S_{\text{range}} = \sum_{i \in 1..|C|} |r_i' - r_i| x_i.$$  \hspace{1cm} (3.2)

Here, $|r_i' - r_i|$ is the contact range of the contact $c_i$. Maximizing this term “pushes” contact selections further from the diagonal of the contact map as shown in figure 3.2

Objective function

Finally, we took a linear combination

$$S_{\text{final}} = \lambda S_{\text{range}} + (1 - \lambda) S_{\text{quality}},$$  \hspace{1cm} (3.3)

as our objective function. Where $\lambda$ is a hyper-parameter to be tuned to balance the trade-off of high-quality contacts and the range of the contacts. The final objective function is apparently linear with all coefficients of $x_i$ positive.
3.3.2 Constraints

Count constraint

The count of the selected contacts needs to be limited for two reasons: The running time of the structural prediction algorithm grows exponentially with respect to the number of constraints so that \( |S| \) needs to be small enough to generate decoy models in reasonable amount time; if too many contacts are selected, many will have a low confidence score generating false constraints.

To enforce that we always select at most \( |S| \) contacts, we add the constraint

\[
\sum_{i \in 1..|C|} x_i \leq |S| \tag{3.4}
\]

Note that because all coefficients in the objective function are positive, the maximum will always be obtained when the most possible constraints are selected. That is, the constraint results in exactly \( |S| \) contacts to be selected.
Diversity constraints

Contacts that are spatially similar provide duplicated information: the spatial constraints on the protein structure would be satisfied or violated together when there are changes to the protein conformation. Therefore, choosing similar contacts will be wasteful.

As a measurement of similarity of two contacts \( c_i \) and \( c_j \), we can define the Manhattan Distance of the contacts as,

\[
ManhattanDistance(c_i, c_j) = |r_i - r_j| + |r'_i - r'_j|.
\]

Unfortunately, we cannot take the sum of the Manhattan Distances and use it as a penalty function for two reasons. The crossing term involves all pairs of constraints, making the objective function expensive to evaluate. The penalty term is now non-convex, which makes it difficult to result in a quick and accurate approximate solution. Therefore, we will add a relaxed penalty in the form of a constraint. That is, for every pair of contacts that are closed by in terms of Manhattan Distance, only one contact is selected. The constraint can be mathematically expressed as:

\[
\forall i, j \text{ s.t. } ManhattanDistance(c_i, c_j) \leq D, x_i + x_j \leq 1, \quad (3.5)
\]

where \( D \) is a hyper-parameter to be toned representing the maximum Manhattan Distances that pairs of contacts can have to be considered “similar”. Note that the constraint is a linear one since we can easily rewritten the set of constraints in the form of \( AX \leq 1 \) where \( A \) is a binary matrix: each row has only two elements \( i, j \) equates to 1 corresponding one pair of \( ManhattanDistance(c_i, c_j) \leq D \) The diversity constraint cause similar contacts, an example shown in Figure 3.3 to be rejected.

### 3.4 ALGORITHM

The ContactSel algorithm transforms the input contacts into an integer program and the pseudo-code is shown in Figure 3.4.

Now, a standard optimization tool can be used to solve the integer program in the following
Output: $a, b, M$ as components in the integer program

$a \leftarrow$ vector of length $|C|$;
$b \leftarrow 1$ vector of length $|C|$;
$M \leftarrow$ Identity Matrix of size $|C|$;

for $i \leftarrow 1$ to $|C|$ do
- $a_i \leftarrow \lambda p + (1 - \lambda)|r'_i - r_i|$
end

for $i \leftarrow 1$ to $|C| - 1$ do
  for $j \leftarrow i + 1$ to $|C|$ do
    if $|r_i - r_j| + |r'_i - r'_j| < D$ then
      Add a new row $m$ to $M$ where $m_i = 1, m_j = 1$;
      Add a new element 1 to $b$
    end
  end
end

Add a 1 row to $M$;
Add an element $n$ to $b$;

Figure 3.3: Example of similar contacts

Figure 3.4: ContactSel algorithm
canonical form.

maximize \[ a^T X \]
subject to \[ M X \leq b \]
\[ X \geq 0 \]
and \[ X \in \mathbb{Z}^{|C|} \] (3.6)
CHAPTER 4: EXPERIMENTS

To assess the performance of the contact selection algorithm, we created a contact-based protein structure prediction pipeline. We generate decoy models for a small dataset of proteins of known structures using our pipeline. The decoy models can be compared against the known models and the accuracy of the decoy models can be used to evaluate the pipeline.

4.1 OVERVIEW OF THE PIPELINE

Here, we present an overview of the pipeline and the overall process is shown in Figure 4.1

![Figure 4.1: Our Contact-based Protein Structure Prediction Pipeline](image)

The input to the pipeline is an amino acid sequence. From it, a multiple sequence alignment (MSA) is performed by Jackhmmer to provide evolution coupling context for DeepContact. The MSA is then used to predict secondary structures as building blocks for the
The Table 1 presents the parameters for Jackhmmer, which is used to provide additional sequence level feature for DeepContact. The contact map generated by DeepContact is used by our ContactSel algorithm to provide spatial constraints. The spatial constraints and the secondary structures are finally used by CONFOUND to build the final decoy models.

4.2 DATASET

The Critical Assessment of protein Structure Prediction (CASP) experiments aim accessing the performance of various methods of protein structure prediction. [14] The CASP assessment provides us with a rich dataset of protein domains to evaluate our pipeline. We selected 36 domains of known structures in the CASP 12 dataset[14] to generate the decoy models. The domains are manually analyzed and cut from 30 proteins, of which the structures were unknown during the CASP 12 competition. The length of the proteins ranges from 133 residues to 670 residues and contains confirmation from all four protein classes of folds (all α, all β, α + β, α/β).

4.3 PROTEIN CONTACT PREDICTION

4.3.1 Multi-sequence Alignment

Multi-sequence alignment provides context information for the contact prediction program and the protein folding program. We perform multiple sequence alignments on the 36 domains on the UniProt_2016_4 database[20] using Jackhmmer[15]. The older UniProt is selected to ensure that functional information provided by the aligned sequences will not be
used by the protein folding program.

The parameter we used for Jackhmmer is shown in Table 5.2

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Filtering Iterations</td>
<td>1</td>
</tr>
<tr>
<td>Alpha Helix Bias</td>
<td>1.0</td>
</tr>
<tr>
<td>Beta Strand Bias</td>
<td>1.0</td>
</tr>
<tr>
<td>Weight Matrices</td>
<td>Default</td>
</tr>
</tbody>
</table>

Table 4.2: Parameters for Jackhmmer

4.3.2 Secondary structure Prediction

We generate secondary structures for the proteins using PSIPRED[3] on the full amino acid sequence. Because the CASP12 dataset is very difficult, the built-in NCBI non-redundant database [21] usually used with PSIPRED cannot provide enough context for secondary structure prediction. Therefore, we use a larger database UniRef[20] instead. The local alignment search needed is performed with BLAST[16] on this database. The second pass of the PSIPRED algorithm is run with parameters shown in Table 4.2.

Lastly, we align the predicted secondary structure to each domain sequence and the gaps in the predicted secondary structure are filled with loops.

4.3.3 Deep Contact Settings

DeepContact [11] is used to produce the contact map needed for the downstream tasks. To ensure general contact quality, the contacts with inferred probability lower than 0.9 are filtered out.

For training of the DeepContact Model, we used the ASTRAL SCOPe 2.06[22] genetic domain filtered at 40% sequence identity. The database does not contain sequences in the CASP 12 dataset. From the MSA and the secondary structure, we can generate the features that DeepContact needed to produce the contact map for the CASP 12 targets. For the 2D features, we used CCMPred[8], EVFold[10], mutual information[23], normalized mutual information[23]. For the global feature, we used the number of effective sequence from JackHmmer. For the 1D feature, we used predicted solvent accessibility[23], predicted
### Table 4.3: Parameters for ContactSel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-processing Filtering Range</td>
<td>12</td>
</tr>
<tr>
<td>Selection Count</td>
<td>(L/2)</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>0.01</td>
</tr>
<tr>
<td>(D)</td>
<td>3</td>
</tr>
</tbody>
</table>

secondary structure[3], and column-wise amino acid frequencies.

#### 4.4 CONTACTSEL IMPLEMENTATION AND SETTINGS

We implement the ContactSel algorithm in Python and Pyomo[24]. We use the current state-of-the-art solver, the CPLEX Mixed Integer Programming Optimizer[17] for solving our integer linear program. The CPLEX Solver is set to balance optimality and feasibility. Dynamic search is used as the CPLEX MIP search method.

The detailed parameter settings for ContactSel is shown in Table 4.3. In the table, \(L\) is the length of the amino acid sequence. \(\lambda\) and \(D\) are determined by performing grid search on three CASP 12 targets: T0866-D1, T0904-D1 and T0915-D1. We only pick three domains for hyper parameter tuning as the generation of decoy model is computationally expansive. With these settings, we selected contacts from the contact maps generated by DeepContact.

#### 4.5 PROTEIN FOLDING

We use CONFOLD[12] as the contact-guided folding program. The selected contacts are used as the input to the algorithm and the same secondary structure prediction we used for DeepContact is also used as an input. For stage-2 of the CONFOLD algorithm, the sheet detector is enabled. 1000 decoy models (of different random seeds) are generated by CONFOLD using our selected contacts. For each target, the top-10 decoy models with the lowest energies are picked as the predicted structures for the protein domain.
4.6 BASELINE AND EVALUATION

To obtain a benchmark baseline, we created another pipeline with the same settings. However, instead of using ContactSel as the contact selection algorithm, we naively select the top $L/2$ DeepContact contacts with highest confidence and range greater or equal to 12 amino acids.

We use TM-score generated by TM-align[25] to evaluate the decoy models generated by the two pipelines. The TM-score is a measurement of structural similarity with value in $(0, 1]$ where 1 indicates a perfect match. In general, a value lower than 0.2 is considered to be pairs of unrelated protein and value over 0.5 are usually in the same fold. The TM-score is measured for each decoy model with its referencing ground truth. And the average TM-score of the top 10 decoy models is used as the evaluation score for each target.
5.1 VISUALIZATION OF THE CONTACT SELECTION

As discussed in [18], the long-range contacts actively stabilize the protein tertiary structure and the residue pairs for long-range interactions may help to improve the de novo design of the protein structure. The lacking of the long-range contacts might comprise the downstream structure prediction task. Thus, we want to ensure that ContactSel can select long-ranged interactions.

For that purpose, we created a visualization of the selected contacts. In Figure 5.1 we plot the contact selection for the CASP 12 domain T0918-D1.

In Figure 5.1a, we can observe that even though the contacts selected are in the high confidence level region, the contacts selected by the naive method clumps into three small regions. Many potential long-range contacts embedded in the contact map of residue 33 - 108 is lost due to the highly uneven selection. Since the selection only involves three regions, the contribution to the spatial constraints is very limiting. In terms of the amino acid chain, the selected contacts effectively closely tie three pairs into “knots” on the chain whereas the rest of the amino acid chain still suffer from great degrees of freedom. The folding algorithm needs to explore a much larger conformation space in order to fully explore the possible folds.

On the other hand, the ContactSel selection is diverse and preserve many long-ranged contacts in the region that were previously ignored by the naive method. The similarity constraints Equation 3.5 prevents neighboring contacts on the contact map to be selected. Therefore, instead of selecting contacts to fill a few high confidence regions, ContactSel effectively outlines and expand the high confidence region. The saved selections are even used for many other high confidence regions. In the end, the selections are more spread out but the spatial constraints still offer some restrictions in the “knots” selected by the naive pipeline. However, since many more regions are selected, instead of folding a chain with a few “knots”, the spatial constraints effectively create small fingerprinting modules, leaving a much smaller confirmation space for the folding algorithm to explore.

We can conclude that ContactSel indeed selects a more diverse set of contacts and we
should theoretically see an improvement in the average TM-score of the predicted models. For this particular domain, the pipeline with ContactSel enjoys a significant 2.09 TM-Score improvement. Since the only differences between the pipelines are the contact selection process, the result can only be explained by the quality of the selection.

5.2 PERFORMANCE COMPARISON

We aggregate the average TM-score of the top-10 decoy models in Figure 5.2. In this head to head comparison, we can clearly see an improvement in the TM-score by ContactSel as most points lie above the diagonal.

To assess the significance of the improvement, we computed the p-value using a single tailed Student’s t-test using the TM-score of each set of predicted structures. The result is shown in Table 5.1 Out of the 36 CASP domains, 21 has seen statistically significant improvements in the TM-score. However, the failure of improvements in the 15 cases cannot be attributed to ContactSel: in 9 cases, the predicted contacts from DeepContact suffers from a top $L/2$ F-1 score lower than 20. The F-1 score indicates the poor quality of the predicted contacts. Since our method is still based on the success of contact prediction, such low F-1 score generates many false constraints and cause the folding algorithm failed to explore the conformation space.

Table 5.1: TM-score improvement for Top-10 decoy models: The $\Delta$ TM-score shows the difference between the Top-10 decoy models generated by ContactSel contacts and the naively selected contacts. The p-value is computed using the single tailed Student’s t-test of the TM-scores. For reference, we also listed the $F-1$ score of the predicted $L/2$ contacts[14]

<table>
<thead>
<tr>
<th>Domain</th>
<th>$\Delta$ TM-score</th>
<th>p-value</th>
<th>F-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>T0899-D1</td>
<td>+0.086</td>
<td>1.722e-14</td>
<td>23.72</td>
</tr>
<tr>
<td>T0898-D1</td>
<td>-0.000</td>
<td>4.834e-01</td>
<td>4.05</td>
</tr>
<tr>
<td>T0897-D2</td>
<td>-0.007</td>
<td>3.828e-02</td>
<td>10.09</td>
</tr>
<tr>
<td>T0886-D2</td>
<td>+0.063</td>
<td>1.282e-06</td>
<td>37.06</td>
</tr>
<tr>
<td>T0912-D3</td>
<td>+0.038</td>
<td>1.067e-08</td>
<td>25.70</td>
</tr>
<tr>
<td>T0897-D1</td>
<td>+0.010</td>
<td>1.584e-02</td>
<td>2.30</td>
</tr>
<tr>
<td>T0894-D1</td>
<td>+0.012</td>
<td>3.798e-03</td>
<td>48.08</td>
</tr>
<tr>
<td>Domain</td>
<td>Δ TM-score</td>
<td>p-value</td>
<td>F-1</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>-----------</td>
<td>-----</td>
</tr>
<tr>
<td>T0863-D2</td>
<td>-0.010</td>
<td>4.398e-03</td>
<td>5.46</td>
</tr>
<tr>
<td>T0946-D1</td>
<td>+0.039</td>
<td>3.219e-07</td>
<td>18.52</td>
</tr>
<tr>
<td>T0923-D1</td>
<td>+0.020</td>
<td>2.544e-04</td>
<td>5.86</td>
</tr>
<tr>
<td>T0863-D1</td>
<td>+0.007</td>
<td>4.604e-02</td>
<td>7.20</td>
</tr>
<tr>
<td>T0941-D1</td>
<td>+0.016</td>
<td>9.786e-08</td>
<td>48.08</td>
</tr>
<tr>
<td>T0904-D1</td>
<td>-0.030</td>
<td>6.853e-03</td>
<td>28.57</td>
</tr>
<tr>
<td>T0900-D1</td>
<td>+0.009</td>
<td>8.365e-03</td>
<td>8.16</td>
</tr>
<tr>
<td>T0878-D1</td>
<td>-0.004</td>
<td>2.230e-02</td>
<td>20.47</td>
</tr>
<tr>
<td>T0859-D1</td>
<td>+0.000</td>
<td>4.663e-01</td>
<td>0</td>
</tr>
<tr>
<td>T0862-D1</td>
<td>+0.010</td>
<td>1.235e-02</td>
<td>21.28</td>
</tr>
<tr>
<td>T0888-D1</td>
<td>+0.102</td>
<td>5.197e-18</td>
<td>8.20</td>
</tr>
<tr>
<td>T0901-D2</td>
<td>+0.003</td>
<td>2.766e-01</td>
<td>74.29</td>
</tr>
<tr>
<td>T0914-D2</td>
<td>-0.001</td>
<td>3.956e-01</td>
<td>9.69</td>
</tr>
<tr>
<td>T0918-D3</td>
<td>+0.100</td>
<td>1.304e-17</td>
<td>41.62</td>
</tr>
<tr>
<td>T0918-D2</td>
<td>+0.053</td>
<td>1.339e-09</td>
<td>29.7</td>
</tr>
<tr>
<td>T0896-D3</td>
<td>+0.002</td>
<td>1.874e-01</td>
<td>11.11</td>
</tr>
<tr>
<td>T0905-D1</td>
<td>+0.090</td>
<td>5.963e-17</td>
<td>36.81</td>
</tr>
<tr>
<td>T0915-D1</td>
<td>+0.063</td>
<td>3.153e-12</td>
<td>12.50</td>
</tr>
<tr>
<td>T0899-D2</td>
<td>+0.000</td>
<td>4.922e-01</td>
<td>23.90</td>
</tr>
<tr>
<td>T0890-D2</td>
<td>+0.012</td>
<td>1.805e-04</td>
<td>9.45</td>
</tr>
<tr>
<td>T0864-D1</td>
<td>+0.085</td>
<td>8.987e-19</td>
<td>38.31</td>
</tr>
<tr>
<td>T0869-D1</td>
<td>+0.001</td>
<td>4.234e-01</td>
<td>1.37</td>
</tr>
<tr>
<td>T0886-D1</td>
<td>+0.067</td>
<td>3.478e-10</td>
<td>40.00</td>
</tr>
<tr>
<td>T0892-D2</td>
<td>-0.024</td>
<td>1.742e-05</td>
<td>22.70</td>
</tr>
<tr>
<td>T0866-D1</td>
<td>-0.010</td>
<td>1.271e-02</td>
<td>46.08</td>
</tr>
<tr>
<td>T0905-D2</td>
<td>+0.025</td>
<td>5.110e-05</td>
<td>35.2</td>
</tr>
<tr>
<td>T0914-D1</td>
<td>-0.003</td>
<td>2.469e-01</td>
<td>10.00</td>
</tr>
</tbody>
</table>
We want to further understand when ContactSel may be successful by a case study. In Figure 5.4, we show 3 decoy structures significantly improved by ContactSel. Upon closer inspection of the domains, we discover that these domains are very $\beta$-strand rich and low in $\alpha$-helices. In fact, these domains are the members of the all beta class, $\alpha + \beta$ class and the $\alpha/\beta$. The performance may come from that long-range contacts promoted and preserved by ContactSel plays an active role in these three class of protein domains.

The structure of $\beta$-sheets usually cause many long-ranged contacts. Although the secondary structure information is already provided to the folding algorithm, only selecting few of closed by contacts to the folding algorithm only provides a weak constraint on pair of $\beta$-strands in the sheets. In this case, the initial decoy model in the first stage of Confold may not assemble the $\beta$-strands close enough so that the matching step in the second stage of Confold cannot be activated. Moreover, given their similarity, the $\beta$-strands on the loose, may be incorrectly identified as loops. In Figure 5.3, we can observe that the resulted decoy model only has a small local region folded whereas the rest of the model consists simply loops. The confirmation is much worse than the confirmation of T0981 in Figure 5.4. It is likely that the second stage of Confold completely filtered out all constraints to avoid the many initial contact constraints in the high confidence region in Figure 5.2.

For $\alpha$ class, the secondary structure information is already necessary for the folding algorithm to produce a right 3D structure since $\alpha$-helix is hydrogen bond rich and is very rigid. Therefore, structures rich in $\alpha$-helices may not enjoy an as much great improvement.

To further confirm this observation, we plotted the TM-score improvement to the abundance of $\alpha$-helices in the structure as in Figure 5.5. We can notice that the TM-score improvement is strongly negatively correlated to the $\alpha$-helix abundance.

To conclude, we discover that the diverse contact selection strategy can lead to high decoy model quality, especially for protein domains rich in $\alpha$-helices.
Figure 5.1: Contact selection on CASP 12 domain T0891-D1: The shaded blue shows the contact map for the CASP 12 domain T0891-D1. The axes corresponding to the index of the residues. The intensity represents the contact probability predicted by DeepContact. The red dots represent the selected contacts. The green lines denotes residue 33. Note that in subfigure 5.1a, no contacts are selected in the region from residue 33 to residue 108.
Figure 5.2: TM-score of decoy models generated with ContactSel contacts and top $L/2$ contacts: The decoy models are scored against the reference structure using TM-align. We took the top 10 decoy models with the highest TM-score and plotted the average of the TM-score.
Figure 5.3: Decoy model of the CASP target T0918 using the naively selected contacts as constraints.

5.3 COMPUTATIONAL EFFICIENCY

We also performed profiling of the entire pipeline for generating the decoy models for all of the 36 CASP domains.

<table>
<thead>
<tr>
<th>Task</th>
<th>CPU Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-sequence Alignment</td>
<td>291</td>
</tr>
<tr>
<td>Secondary Structure Prediction</td>
<td>460</td>
</tr>
<tr>
<td>Contact Prediction</td>
<td>4 (GPU aided)</td>
</tr>
<tr>
<td>ContactSel</td>
<td>0.2</td>
</tr>
<tr>
<td>Protein Folding</td>
<td>1,843,200</td>
</tr>
</tbody>
</table>

Table 5.2: Break Down of Computational Cost for the Pipeline

Clearly, the cost of ContactSel is negligible comparing to its upstream and downstream task. It is to be expected as the formulation into integer programming is only a $\Theta(L^2)$ task and LIP problems can be solved efficiently. ContactSel is thus a worthwhile trade-off for the potential quality gain of the decoy models.
Figure 5.4: 3D structures that are significantly improved by ContactSel. Figure a comes from domain T0918-D3 and is in the $\alpha + \beta$ family, Figure b comes from domain T0918-D1 and is from the $\beta$ family, and figure c comes from domain T0888-D1 and is from the $\alpha/\beta$ family. Note that all of these structures have some $\beta$—sheets formed by the $\beta$-strands.

Figure 5.5: TM-score improvement plotted against $\alpha$-helix percentage: The alpha-helix percentage is calculated using percentage of residues assigned as helices in the predicted secondary structure. The Spearman correlation between the TM-score improvement and the alpha-helix percentage is at -0.372 with a p-value of 0.0253.
CHAPTER 6: CONCLUSION AND FUTURE WORK

Through our experiments, we have shown that our selection algorithm is more advantageous than a method simply selecting the top predicted contacts. With the same number of contacts selected, ContactSel contact selections preserve more structural information. The improvement in structural prediction is more prominent in domains low in $\alpha$-helices. Here, we can conclude that selecting diverse and informative contacts is an important question.

Our hypothesis is that the protein folding algorithm can take advantages of the contacts to more easily pair $\beta$-strands into sheets and multiple local conformation modules are easier to fold into native conformation. However, large scale experimentation on different classes of protein domains is way beyond the scope of our thesis but could be a potential topic to be explored.

In addition, further exploring the quality gain from the improved selection can be a potential path for ab initio modeling. More work can still be done to the selection process. For instance, here we use the objective function Equation 3.2 to promote contacts of longer range. However, the contact range actually has different significance for protein domains of different fold class [18]. We could associate different weight for different ranges for domains in different fold class.

In addition, integrating the contact selection process, e.g. incorporating ContactSel into the second stage of Confold, could be another potential improvement to Confold.
REFERENCES


