DeepTracer: Automated Protein Complex Structure Prediction from CoV-related Cryo-EM Density Maps

Jonas Pfab, Nhut Minh Phan, Dong Si*

July 20, 2020

Abstract

Information about the macromolecular structure of viral protein complexes such as SARS-CoV-2, and the related cellular and molecular mechanisms can assist the search for vaccines and drug development processes. To obtain such structural information, we present DeepTracer, a fully automatic deep learning-based method for de novo multi-chain protein complex structure prediction from high-resolution cryo-electron microscopy (cryo-EM) density maps. We applied DeepTracer on a set of 62 coronavirus-related raw experimental density maps, among them 10 with no existing deposited model structure. We observed an average residue match of 84% with the deposited structures and an average RMSD of 0.93Å. Larger comparative tests further exemplify DeepTracer's competitive accuracy and efficiency of multi-chain all-atom complex structure prediction, with the ability of tracing around 60,000 residues within two hours. The web service and prediction results are globally accessible at `https://deeptracer.uw.edu`.

1 Introduction

The determining factor for a protein's functionality is its structure, which is given by a unique sequence of amino acids that make up the protein and their three-dimensional arrangement [1]. Consequently, researchers can draw conclusions about the behavior of a protein based solely on its molecular structure. These outcomes can be useful in the development of new vaccines and drugs as viral fusion proteins play a central role in how the viruses invade the host's cells [2]. In order to prevent infections, researchers attempt to develop vaccines and medicines that target these fusion proteins. This strategy is currently applied to find an effective vaccine for the SARS-CoV-2 virus [3, 4, 5]. The structural information about the fusion proteins is crucial for researchers to predict their behaviors and ultimately find the right vaccine [6].

To determine the structure of a protein, this work builds upon cryo-electron microscopy (cryo-EM) data [7]. Cryo-EM allows researchers to capture three-dimensional maps of macromolecules, which describe the density of electrons at a near-atomic resolution. The technology has gained popularity in recent years as an alternative to established structure determination methods, such as X-ray crystallography, due to its improved quality and efficiency [8, 9]. In the midst of the current global crisis, it is telling that cryo-EM is being deployed right alongside X-ray crystallography in support of the search for medicines and vaccines to fight the current COVID-19 pandemic [10]. To derive the structure of a protein based on its 3D cryo-EM electron density map, researchers currently either have to manually fit the atoms or resort to existing template-based or homology modeling methods [11, 12, 13]. The manual fitting of atoms represents an enormous effort as proteins complexes usually consist of several thousand atoms, making it virtually impossible for larger structures. Therefore, there is a tremendous demand for a prediction method that automatically predicts the molecular structure from a cryo-EM density map. Unfortunately, existing prediction tools [14, 15, 16, 17, 18] such as Rosetta, MAINMAST, and Phenix predict fragments of a protein complex, or require extensive manual processing steps. Due to the ability of cryo-EM to capture multiple large proteins in the course of a single study [19, 20], a fully automated, efficient tool to predict complex structures would be crucial to increase the throughput of the technology and speed up the development of medicines.

In this paper, we present DeepTracer, a fully automated software tool that predicts the all-atom structure of a protein complex based solely on its cryo-EM density map and amino acid sequence (Figure 1). No manual processing of the density map is necessary, and the tool requires no further parameters to run predictions. The core of the prediction method is a tailored deep convolutional neural network that allows for fast and accurate structure predictions when combined with complex pre- and post-processing steps. We also provide a web service and a CoV-related dataset along with the prediction results at DeepTracer's website. To our knowledge, this is the first web service for fully-automated protein complex prediction and coronavirus modeling using 3D cryo-EM.

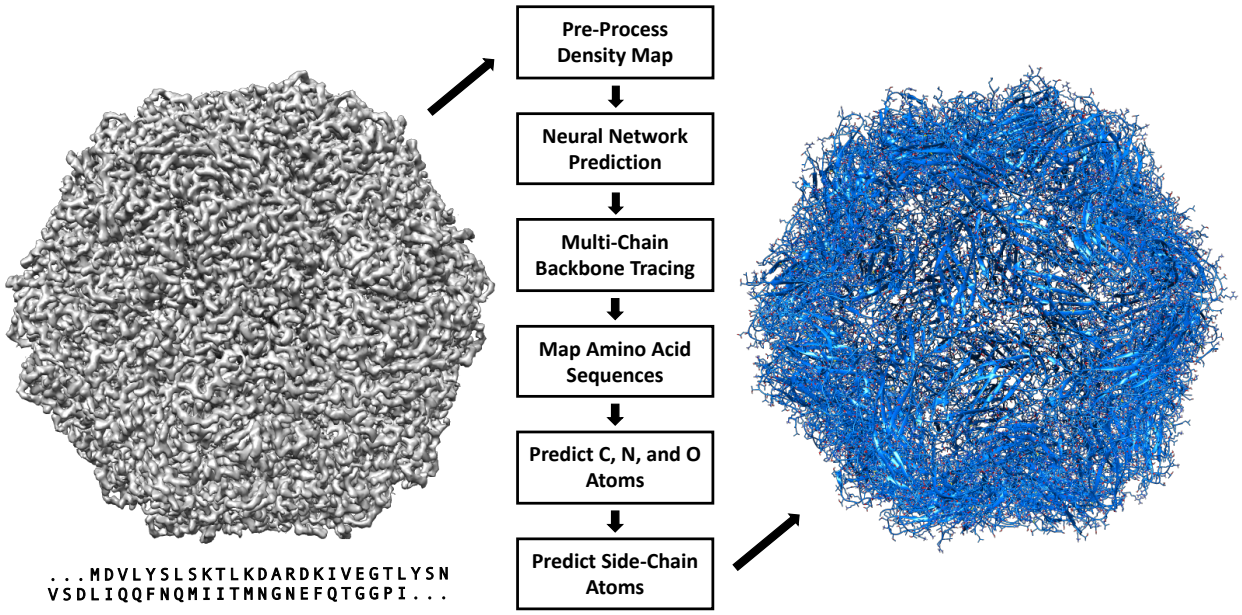

Figure 1: DeepTracer prediction pipeline. All-atom structure of multi-chain protein complexes is predicted fully automatic solely from a density map and amino acid sequence using the steps shown in the center of the figure. The structure shown on the right side is an actual DeepTracer prediction.

2 Results

In this section, we applied DeepTracer to experimental coronavirus-related density maps for evaluation purposes. Specifically, we compared its effectiveness with the existing Phenix map-to-model function. Larger comparative tests can be found in the supplementary materials. The structure of this section is as follows: Section 2.1 describes the metrics we utilize to facilitate this comparison; Section 2.2 presents the prediction results of both methods for a dataset of coronavirus-related density maps; Section 2.3 conducts a brief analysis of DeepTracer's computation time.

2.1 Metrics

To ensure the objectivity of the comparison with the existing Phenix method, we used the `phenix.chain_comparison` tool [21], which is available at no cost as part of the Phenix software suite, to evaluate the accuracy of predicted structures. This tool compares predicted and deposited model protein structures by finding a one-to-one matching between model and predicted residues based on Cα positions. For a model residue to match a predicted residue, it cannot be further apart from the other than 3Å. Based on this matching, several metrics determining the prediction accuracy are calculated. The first metric is the root-mean-square deviation (RMSD), which expresses the average distance between Cα atoms of the matched model and predicted residues. Second, the coverage of the prediction is expressed using the matching percentage. This value represents the proportion of deposited model residues, which have a matching predicted residue and is calculated by dividing the number of matches by the total number of model residues. Third, to evaluate how well the amino acid types were predicted, the `chain_comparison` tool calculates the sequence matching percentage, which denotes the percentage of matched model and predicted residues that have the same amino acid type. Lastly, to get a sense of how well the predicted residues are connected, the mean length of matched segments is calculated where consecutive matches are connected both in the deposited model and predicted structure. Besides the metrics calculated by the `phenix.chain_comparison` tool, we also apply the LGA (Local-Global Alignment) algorithm, which aligns the predicted and native structures and computes the GDC (Global Distance Calculation) score. This score measures the similarity of two structures based on all atoms (including side-chains) on a range of 0 to 100 with 100 being a perfect match [22, 23]. We applied it on the most important dataset of SARS-CoV-2 density maps due to the high manual and computational effort involved in the calculation of this metric.

2.2 Coronavirus-Related Predictions

In the search for an effective COVID-19 vaccine and medicine, structural information about the viral protein are crucial. Therefore, we applied DeepTracer on a set of coronavirus-related density maps to demonstrate how it can aid researchers in obtaining such structural information. To create a point of comparison, we applied Phenix on the same set of density maps. The dataset was aggregated by the EMDataResource and contained 62 high-resolution density maps, 52 of which have a deposited model

PDB structure [24]. The dataset and prediction results will be actively updated at DeepTracer's website as more and more data is deposited to EMDR. To our knowledge, this is the first CoV-related 3D cryo-EM modeling test dataset.

The scatter plots in Figure 2 show the evaluation results for the metrics calculated by Phenix's `chain_comparison` tool, for the 52 coronavirus-related density maps that have a deposited model structure. We can see that DeepTracer outperformed Phenix in all four metrics. The average percentage of matched model residues is 84% for DeepTracer and 49.8% for Phenix. This means that, on average, around 34% more residues were correctly predicted by DeepTracer than by Phenix. The RMSD metric calculated an average value of 1.37Å for Phenix's structure predictions compared to 0.93Å with DeepTracer. Thus, DeepTracer not only predicts more residues correctly than Phenix, but the correctly predicted residues were also closer to the model residues by around 0.4Å. For the sequence matching results, Phenix scored 24.95%, while DeepTracer achieved a sequence matching percentage of 63.08%. For higher-resolution maps this value is significantly higher, as side-chain information, which DeepTracer uses to determine the amino acid type of a residue, is usually hardly visible in lower-resolution maps. Finally, the mean length of consecutively matched predicted and model residues increased from 8.9 with Phenix to 20 with DeepTracer. Although multiple factors can influence this value, the results show that DeepTracer can provide predictions that match the deposited model structures better than those from Phenix.

The SARS-CoV-2 results from Table 1 show a similar pattern as the results of all coronavirus-related maps. DeepTracer outperformed Phenix in every metric with the most significant differences in the matching percentage and sequence matching. Additionally, the DeepTracer achieved a GDC score almost three times that of the Phenix method.

Figure 3 illustrates a comparison between the deposited model and predicted structures from DeepTracer and Phenix for the EMD-30178 density map showing a SARS-CoV-2 polymerase. The deposited model structure consists of 1213 residues connected in 4 chains. DeepTracer's prediction comprises 1194 residues in 7 chains whereas Phenix predicted only 1057 residues split into 33 different chains. We can see that Phenix's prediction is more fragmented with many missing parts where DeepTracer correctly placed residues. This observation aligns with the metric numbers described above and shown in Figure 2.

In Figure 4, we can see DeepTracer's predictions of the EMD-30044 density map, which captures the human receptor angiotensin-converting enzyme 2 (ACE2) to which the spike protein of the SARS-CoV-2 virus binds to [11] and the EMD-21374 density map of a SARS-CoV-2 spike glycoprotein. No model structure has been deposited to the EMDR for either density map as of the date this paper is announced. This represents an ideal opportunity to showcase the potential of DeepTracer. Without any other parameters or manual processing steps, DeepTracer can predict detailed structural information based on the density maps. Researchers can use the predictions to develop therapeutics targeting the binding process between the spike protein and the human enzyme.

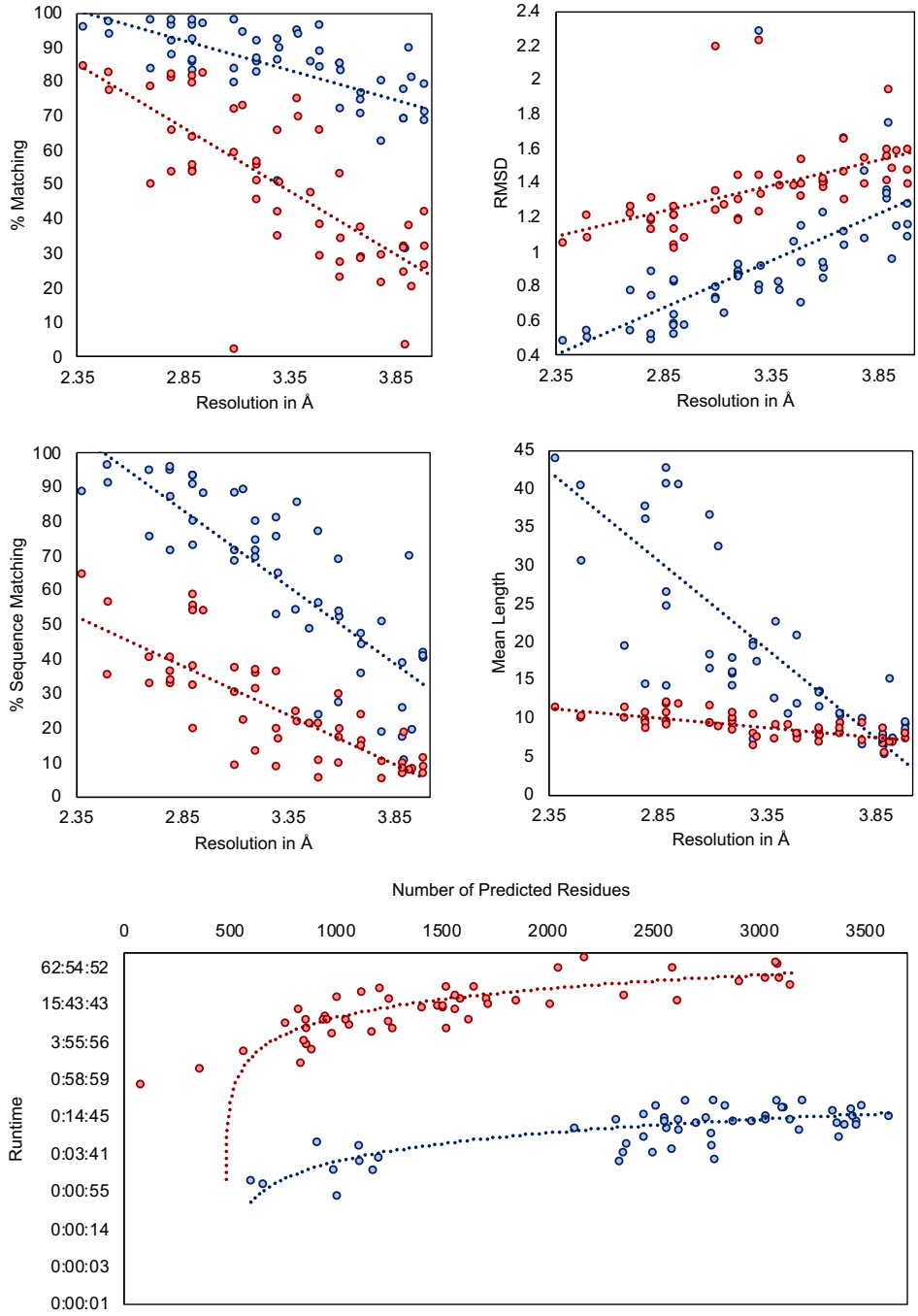

Figure 2: Prediction results for coronavirus-related density maps. Evaluation of prediction results from DeepTracer (blue) and Phenix (red) for 52 coronavirus-related high-resolution density maps. The dotted lines represent the trend for each prediction method. Computation times are shown on a logarithmic scale. Precise data can be found in Table S2.

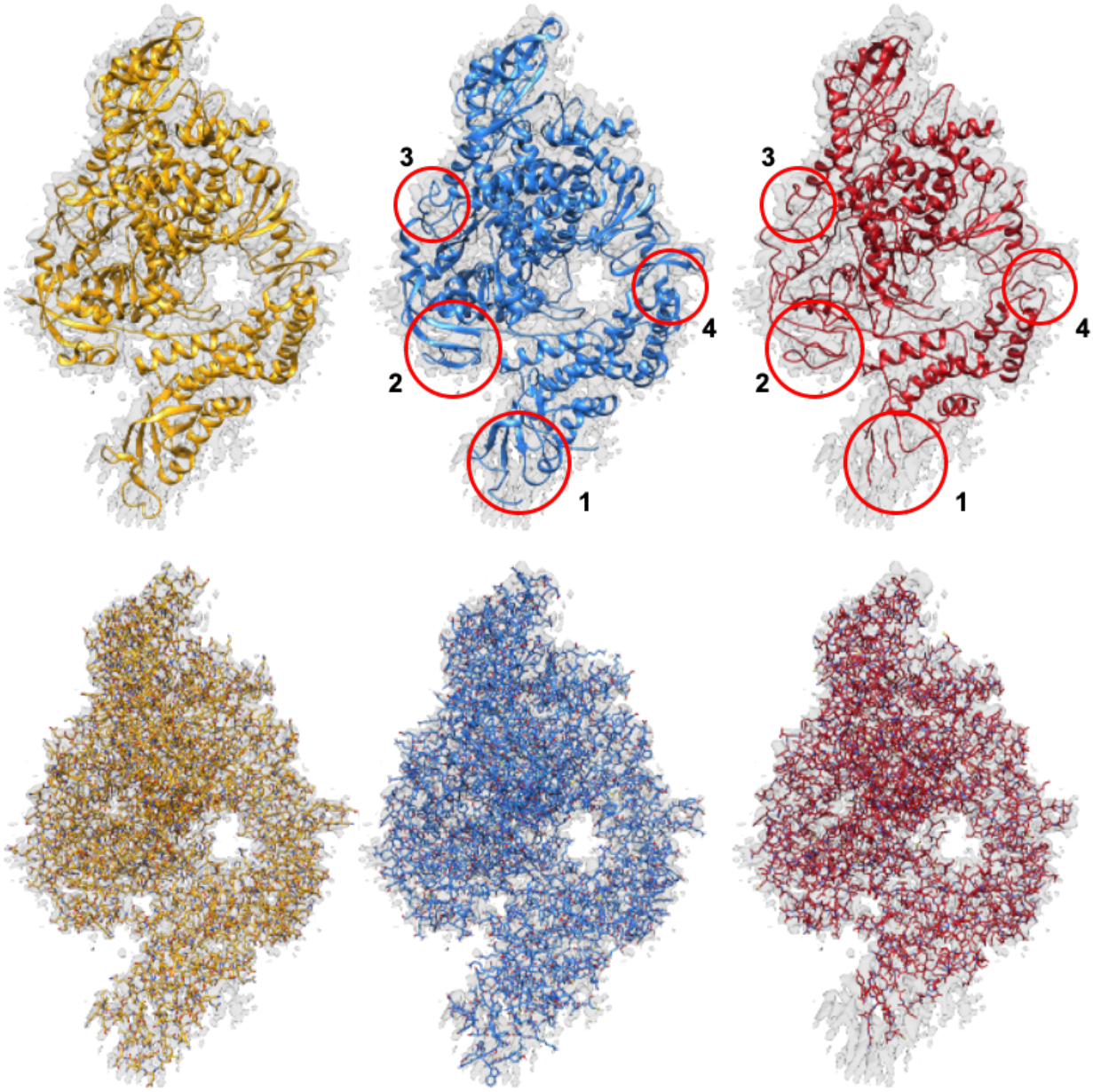

Figure 3: Prediction results for SARS-CoV-2 polymerase density map (EMD-30178). Comparison of the deposited model (gold) and predicted structures by DeepTracer (blue) and Phenix (red) for the SARS-CoV-2 polymerase density map (EMD-30178). Upper row shows structures in ribbon view and lower row in all-atom view. The four red circles mark areas where DeepTracer prediction is more accurate than Phenix's prediction.

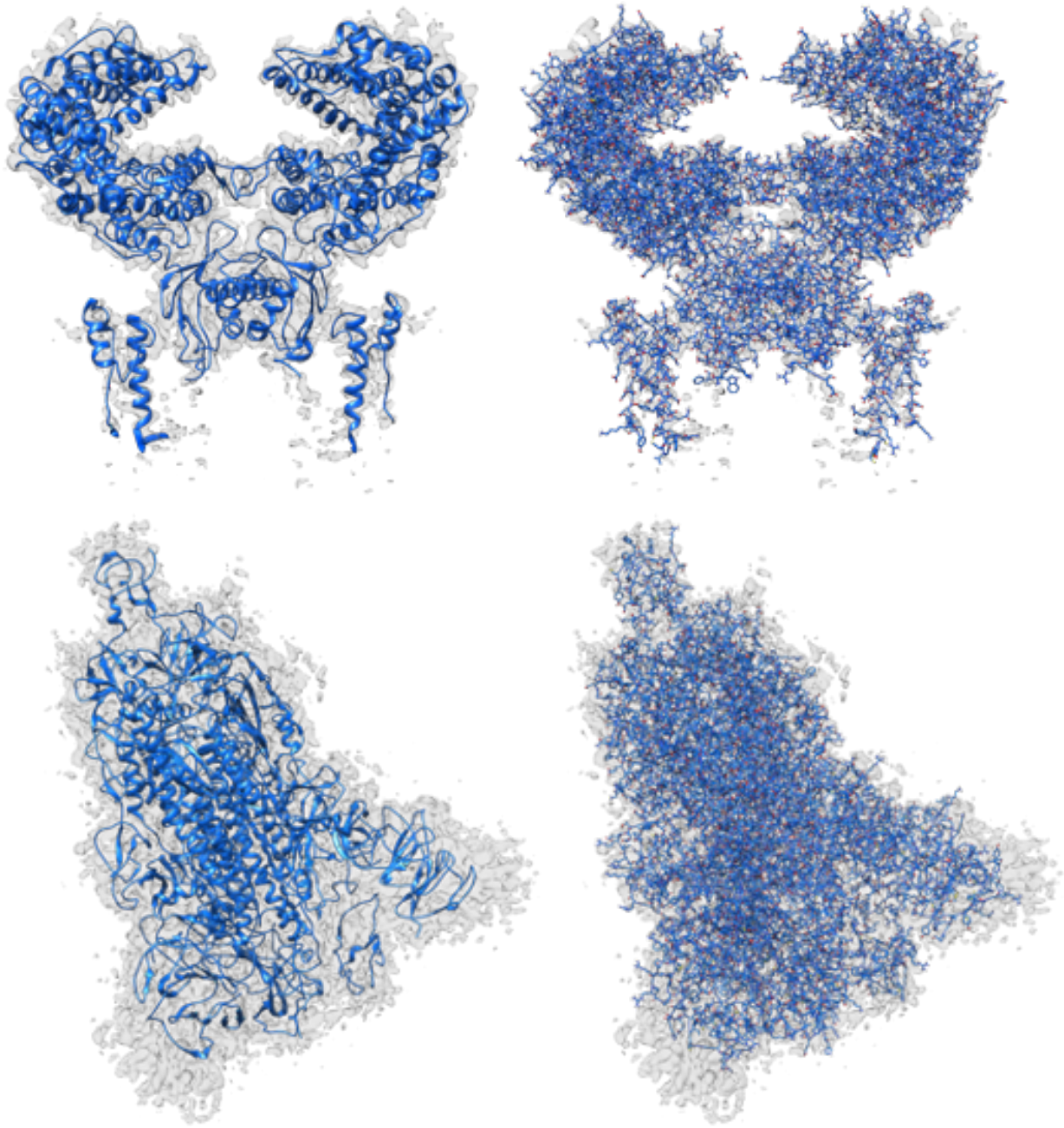

Figure 4: Predictions of SARS-CoV-2 density maps, which do not have deposited model structures in the EMDR. DeepTracer predictions for the EMD-30044 density map (top) showing a human receptor angiotensin-converting enzyme 2 (ACE2) to which spike proteins of the SARS-CoV-2 virus bind to and the EMD-21374 depicting a SARS-CoV-2 spike glycoprotein. No model structure has been deposited to the EMDataResource for the density maps as of the date this paper is announced.

Table 1: Comparison of DeepTracer (DT) and Phenix (P) for SARS-CoV-2 dataset.

EMDB	**PDB**	**Residues**	**% Matching**		**RMSD**		**% Seq ID**		**GDC**	
			DT	P	DT	P	DT	P	DT	P
21375	6vsb	2905	84.90	48.60	1.14	1.40	45.90	20.90	17.88	5.39
21452	6vxx	2916	91.40	53.80	0.96	1.18	61.30	40.00	-	-
30039	6m17	3072	80.30	53.10	1.72	1.72	69.80	54.60	11.84	8.31
30127	6m71	1077	91.70	54.20	1.02	1.20	58.60	16.60	20.74	8.89
30178	7btf	1227	94.90	81.00	0.83	1.09	85.80	51.80	65.57	23.06
30209	7bv1	1102	87.60	67.00	0.84	1.29	87.50	30.50	55.62	18.15
30210	7bv2	1006	92.40	78.20	0.78	1.08	88.90	53.70	40.90	13.32
Avg.			89.03	62.27	1.04	1.28	71.11	38.30	35.42	12.85

GDC score could not be calculated for the EMD-21452 density map as the LGA web service could not process the predicted structures due to their size.

2.3 Computation Time

A major bottleneck of the existing prediction methods is their computational complexity, which renders them unable to predict larger protein complexes. Thus, we conducted an analysis of DeepTracer's computational time versus Phenix's. The result is shown in Figure 2. The predictions were executed on a machine with an Nvidia GeForce GTX 1080 Ti GPU, 8 processors, and 62 GB of memory. Although a comparison with the Phenix method is not entirely fair as Phenix does not take advantage of the machine's GPU, this comparison provides a glimpse of the possibility that DeepTracer can achieve. We observed that Phenix took about 45 minutes to process a map containing 79 residues, while DeepTracer processed a map containing 2798 residues in only 26 minutes. Furthermore, the largest cryo-EM map (EMD-9891) that DeepTracer was tested on required around 14 minutes to complete the prediction, whereas Phenix's processing time for this map was over 60 hours. DeepTracer is able to exploit the processing power of the GPU, which is becoming a staple on modern computing systems, increasing the throughput of scientific discovery. This means that DeepTracer can predict even very large protein complexes in a matter of hours. As an example, it traced around 60,000 residues for the EMD-9829 density map within only two hours.

3 Discussion

In this paper, we present DeepTracer, a fully-automatic tool that predicts the all-atom structures of protein complexes based on their cryo-EM density maps, using a tailored deep convolutional neural network and a set of computational methods. We applied this novel software on a set of coronavirus-related density maps and compared the results to Phenix, the state of the art cryo-EM prediction method [14]. We found that DeepTracer correctly predicted, on average, around 34% more residues than Phenix with an average RMSD improvement of around 0.4Å, from 1.37Å to 0.93Å. We also applied DeepTracer on a dataset of 476 density maps aggregated by Phenix's team, and calculated a coverage of 76.93% compared to 45.65% with Phenix and an average RMSD value of 1.18Å for DeepTracer and 1.29Å for Phenix. Detailed description and discussion can be found in the supplementary material. Furthermore, we compared DeepTracer with Rosetta and MAINMAST on nine density maps and observed significant RMSD improvements in comparison with Rosetta from 1.37Å to 0.85Å and much more complete predictions compared to MAINMAST with a coverage increase of 57%, from 36.4% to 93.4%. Detailed description and discussion can be found in the supplementary material. These results represent a significant accuracy boost, resulting in more complete protein structures. Particularly, for large protein complexes, DeepTracer can complete a prediction much faster than other methods, predicting tens of thousands of residues with million of atoms within only a few hours. We achieved the results without any manual pre-processing steps, such as zoning or cutting of the density map using a deposited model structure. This means we can run predictions without any prior knowledge about the cryo-EM map, and the users do not need to tune any parameters in order to obtain an accurate prediction.

As the cryo-EM technology becomes more readily available, the number of captured density maps, especially larger protein complexes, is rising rapidly. DeepTracer allows for a greater throughput of cryo-EM as it can automatically and accurately infer structural information from density maps of macromolecule. This outcome ultimately accelerates the scientific discovery process, which is particularly urgent today, given the ongoing coronavirus pandemic. Coronavirus-related density maps are deposited to the EMDR on a daily basis. Our efficient and automated method to model these maps is an important tool for researchers to resolve the structural information of the virus-related macromolecules.

4 Methods

DeepTracer performs an array of tasks to predict the structure of a protein. It pre-processes each density map for the neural network, feeds the density map to the network, and then transforms the output into a protein structure. An overview of the steps involved in this process is provided in Figure 1. In this section, we focus on all prediction steps starting with the neural network. A detailed description of the pre-processing steps can be found in the supplementary material.

4.1 Neural Network Architecture

The convolutional neural network is the central piece of DeepTracer. Its job is to predict three vital pieces of information: the locations of amino acids, secondary structure positions, and amino acid types. Here, we take a closer look at the architecture of the neural network as used by DeepTracer. We start by looking at architectural details of our tailored U-Net, and then examine how multiple U-Nets are connected to form the complete network.

The U-Net is a convolutional network architecture developed by researchers at the University of Freiburg. Its name derives from the U-shape of its architecture. The U-Net excels in fast and precise image segmentation tasks, particularly for biomedical applications [25]. For DeepTracer, we modified its original 2D architecture for 3D density maps. The detailed architecture of the model used by DeepTracer can be seen at the bottom of Figure 5. The pre-processed cryo-EM density maps are fed to the 64^3 input layer. The output layer has the same 64^3 shape with `N` different channels. The number of channels depends hereby on the number of classes the U-Net predicts and is explained in more detail in the following.

As aforementioned, the deep learning model predicts multiple structural aspects of a protein, including the atom positions, secondary structure elements, and the amino acid types. The model also predicts the backbone location of the protein structure to allow for the post-processing step described in Section 4.3, which connects the predicted atoms. For each of those predictions we use separate U-Nets which are all combined to a single model as shown in Figure 5. The input of the model is a 64^3 volume data grid from the pre-processed density map. The atoms U-Net is responsible for predicting whether each voxel contains either a Cα atom, a nitrogen atom, a carbon atom, or no atom. Therefore, the output of this U-Net has four channels, one for each predicted class. The backbone U-Net predicts whether each voxel belongs to the backbone, meaning either carbon alpha, carbon, or nitrogen atom, part of a side chain, or not a part of the protein, which leads to three different output channels. The secondary structure U-Net is responsible for predicting the secondary structure of each voxel. Therefore, we have a four-channel output for loops, sheets, helices, and no structure. Additionally, U-Net predicts the amino acid type for every voxel. As 20 different types of amino acids have been found in nature, we have 21 output channels, representing the amino acids plus the case in which the voxel is not part of the protein. Next, we can take a closer look to the U-Net itself.

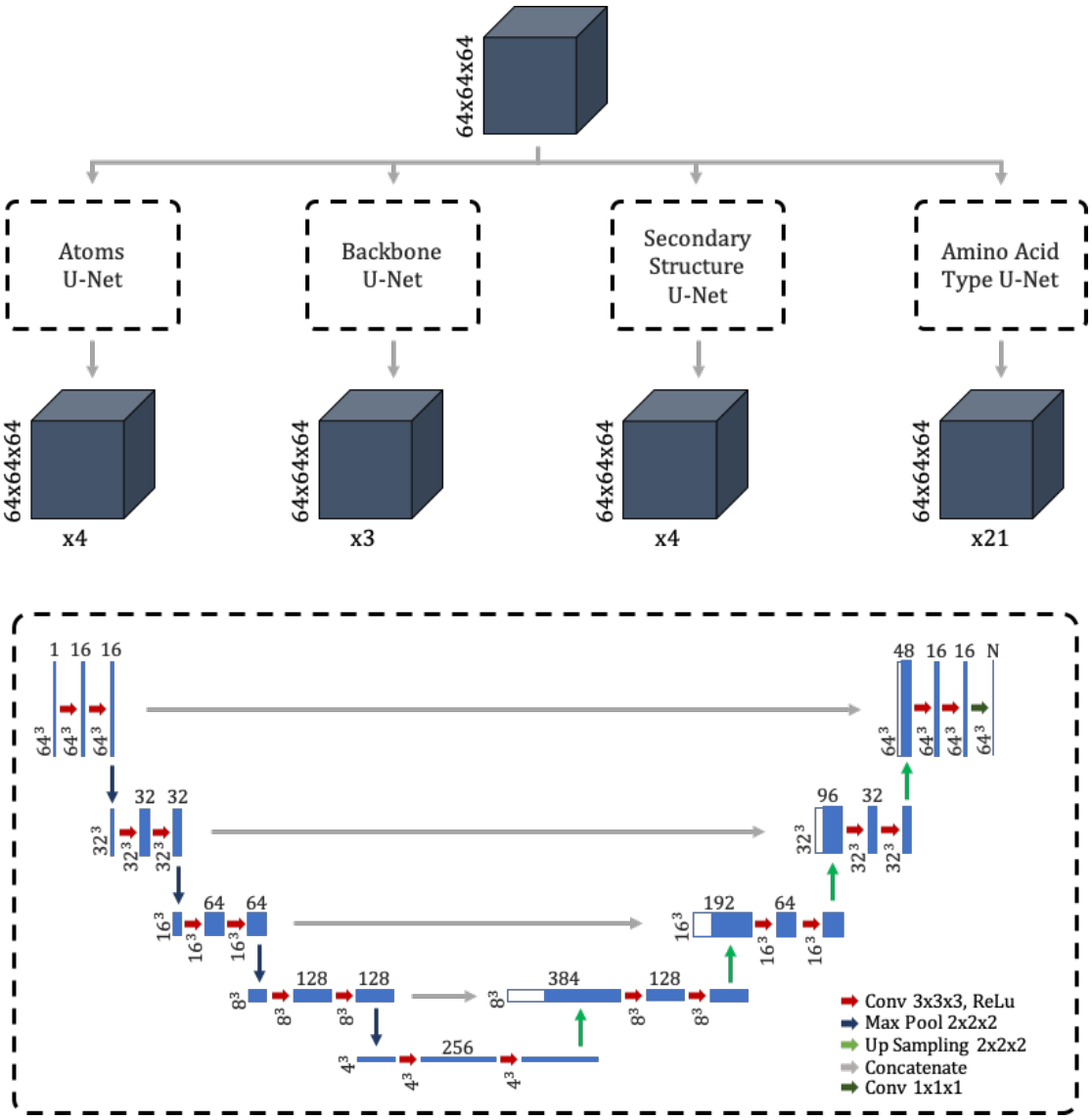

Figure 5: Architecture of tailored convolutional neural network. Top shows overview of DeepTracer's neural network architecture consisting of four parallel U-Nets. The gray boxes show the input and output maps, with their dimensions noted to the left and the number of channels marked below. Bottom dashed box shows the detailed architecture of each parallel U-Net. The blue boxes show the output maps of the different layers where the dimensions of the maps are depicted on the left and the number of channels is depicted on top.

4.2 Data Collection

Before training the U-Net model, we have to collect a training dataset. Previous projects, such as [17], used simulated density maps to train their neural networks. However, for the network to learn common noise patterns in cryo-EM density maps, we decided to use experimental maps. The maps were downloaded from the EMDataResource website [26] together with their deposited model structures that served as the ground truth in the training process and were fetched from RCSB Protein Data Bank [27]. As this work focuses on high resolution maps, we only used density maps with a resolution of 4Å or better. In total, we downloaded 1,800 experimental density maps and their corresponding deposited model structures. The maps were randomly split into training and validation sets with an 80:20 ratio.

To label each density map, we created masks with the same dimensions as the grid of the density map, providing a label for each voxel. The labels of the masks were hereby created based on the deposited model structures of each density map. As shown in Figure 5, the model has four different outputs, for each of which we created separate masks. The atoms mask should provide a label for each voxel whether or not it contains a Cα, C, or N atom. Therefore, we filtered out these atoms from the protein structure, calculated the corresponding grid indices for their location, and set that voxel and all directly neighboring voxels to the value representing the atom (1 for Cα, 2 for C and 3 for N atoms). A visualization of an atom mask can be found in the supplementary material in Figure S8.

The masks for the backbone, secondary structure, and amino acid type U-Net, were created in a similar manner. The backbone mask filters all backbone atoms and side-chain atoms and sets the respective voxels and all surrounding voxels with a distance of 2 to 1 for backbone and 2 for side chain. To create the secondary structure mask, we filtered all atoms for helices, sheets, and loops and then set all voxels with a distance of 4 surrounding the atoms to 1 for loop, 2 for helix, and 3 for sheet. Finally, for the amino acid type mask, all Cα atoms for each of the 20 amino acid types were filtered out, and all surrounding voxels within a distance of 3 were set to a value between 1 and 20, where each value corresponds to a specific amino acid type. An example of all masks can be seen in Figure 6. An example of a raw prediction from the trained neural network for the EMD-6272 density map can be found in Figure S4 from the supplementary materials.

4.3 Tracing Backbone

This step uses the output of the U-Net to create an initial protein structure prediction which only contains Cα atoms connected into chains. This is a central post-processing step, and its accuracy determines to a great extent how well the remaining post-processing steps will perform. The step can be split into three different parts. First, we identify disconnected chains which can be processed independently. Second, we calculate the x,y, and z coordinates of the Cα atoms. Last, we connect them into chains by applying a modified travelling salesman algorithm.

Identifying chains prior to any atom prediction has two advantages. First, it improves the performance of the step as each chain will contain a lower number of atoms

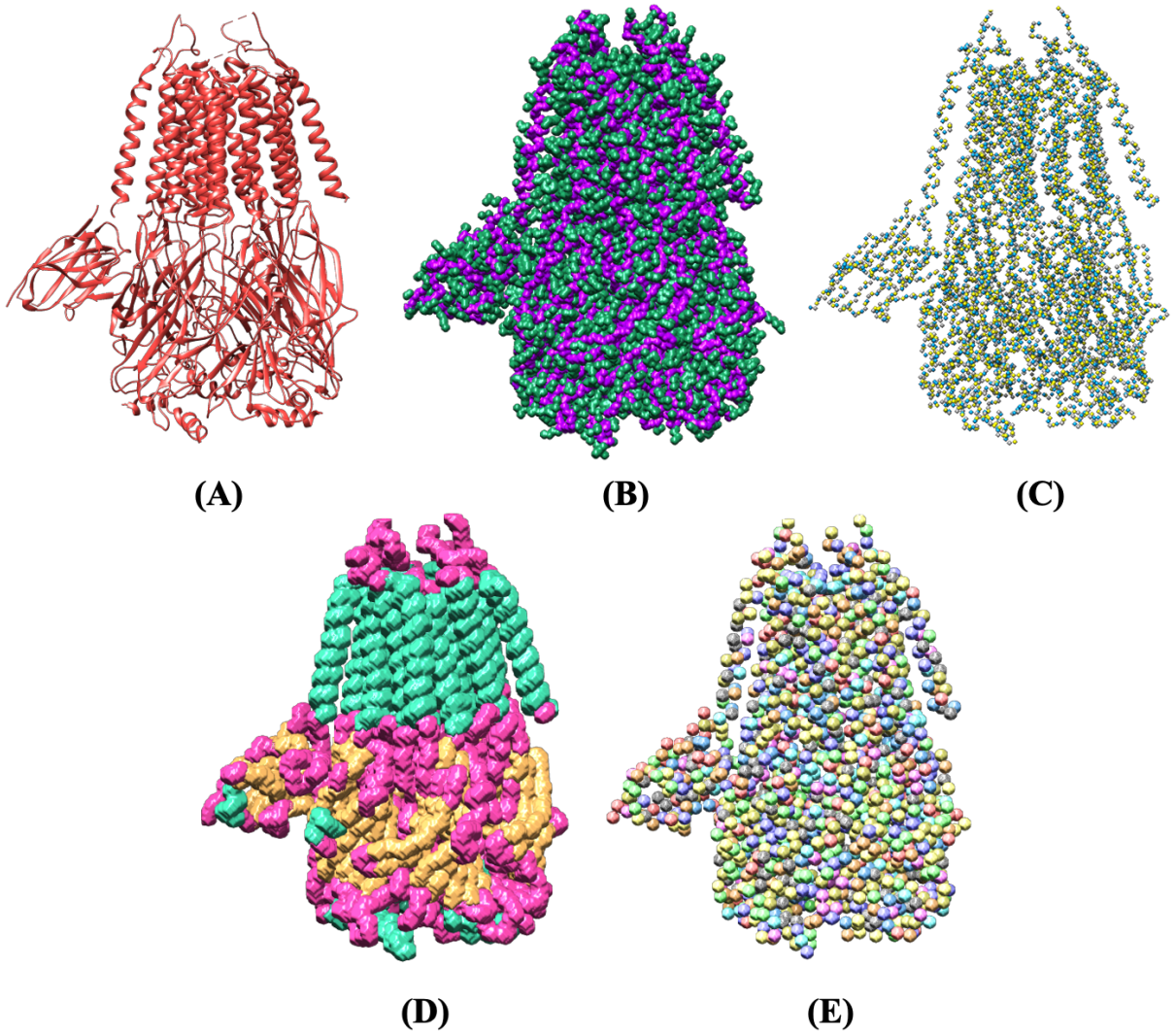

Figure 6: Example masks from the training dataset based on the PDB-6NQ1 deposited model structure. (A) Deposited model structure. (B) Backbone (Cα, C, and N atoms) in purple and side chains in green. (C) Atoms mask with labels for Cα, C, and N atoms. (D) Secondary structure mask with helices in turquoise, loops in pink and sheets in orange. (E) Amino acid type mask with 20 different colors.

that have to be connected by the travelling salesman algorithm. Second, it decreases the number of incorrect connections between atoms of separate chains as they are processed independently. To identify chains, we used the output of the backbone U-Net. We rounded each voxel of the confidence map to either zero or one and then found connected areas of voxels with a value of one. Disconnected areas were then identified as separate chains. An example of the chain identification process visualized for the EMD-0478 density map can be seen in Figure 7.

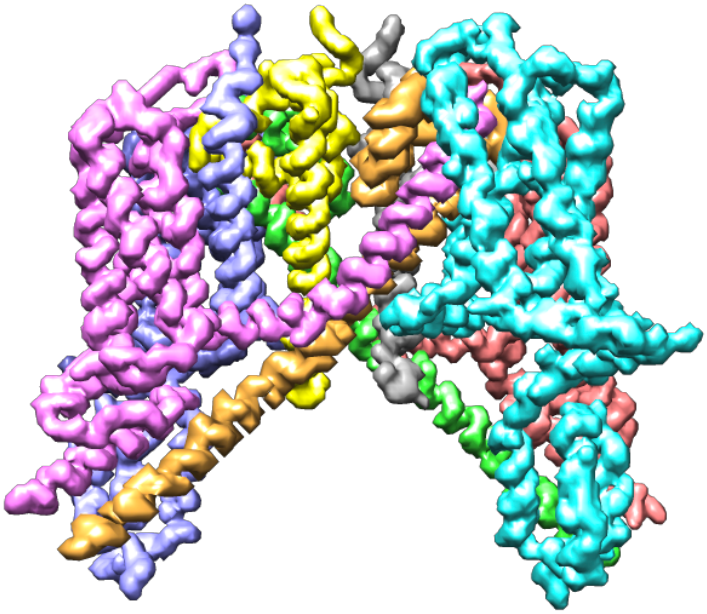

Figure 7: Backbone confidence map of the EMD-0478 density map with identified chains annotated in different colors.

To find the x, y, and z coordinates of the Cα atoms, we utilized the Cα channel from the output of the atoms U-Net. A voxel value in this map describes the confidence of whether this voxel contains a Cα atom. The coordinates were then calculated in two steps. First, we found the indices of all local maximums in the confidence map within a distance of 4 voxels that have a minimum value of 0.5. Next, we refined the indices by calculating the center of mass of all voxels within a distance of 4 surrounding the local maximums. This is possible as we moved away from integer indices towards floating point coordinates, giving us the opportunity to express locations more precisely.

The most challenging part of this prediction step is to connect the predicted Cα atoms into chains correctly. The factorial growth of the number of ways in which the atoms can be connected makes it infeasible to test all possible solutions even for a low number of atoms. Therefore, we decided to solve the problem using an optimization algorithm, particularly, for the travelling salesman problem (TSP). However, our problem does not match every criterion of the traveling salesman problem. The shortest possible path is not necessarily the correct one as the ideal distance between Cα atoms is 3.8Å [28]. Deviations from this value are, however, possible due to prediction inac-

curacies. Additionally, it is often difficult to decide only based on the distance which atoms to connect if there are multiple possibilities with a similar distance. To address these issues, we developed a custom confidence function instead of solely relying on the euclidean distance between atoms. The confidence function's idea is to return a score between 0 and 1, which expresses how confident we are that these two atoms are connected. The goal of the TSP algorithm is then to connect the atoms such that the sum of all confidence scores between connected atoms is maximized.

The calculation of the confidence score between Cα atoms considers two factors: the Euclidean distance between the atoms, and the average density values of voxels that lay in between the atoms on the backbone confidence map predicted by the backbone U-Net. The latter factor is to ensure that connections are made along the backbone of the structure. The voxels that lay between the atoms are found using Bresenham's algorithm [29]. To transform these metric values to a confidence score, we used a probability density function $p(x, \mu, \sigma)$ with a mean μ, which represents the ideal metric value, and a standard deviation σ. To make sure that the function returns exactly 1 at the mean, we normalized it by dividing it by the probability density value at the mean. For the euclidean distance, we used a mean of 3.8 and a standard deviation of 1. The average backbone confidence has a mean of 1 and a standard deviation of 0.3. The standard deviations were determined based on several rounds of testing. Both probability density functions can be seen in Figure S9. In order to combine both results into a single confidence score, we simply multiply both values. As the TSP algorithm was designed to minimize distances between paths, we then just subtract the confidence score from 1 and provide it to the algorithm.

To apply the TSP algorithm, we had to specify a start/end point. However, we could not know yet at which atom the chain will start and end. Therefore, we added a new atom that is connected to every other atom with a confidence of 1. This atom was then specified as the start/end and later on removed from the actual chain. An example of the application of the TSP on a list of Cα atoms can be seen in Figure 8.

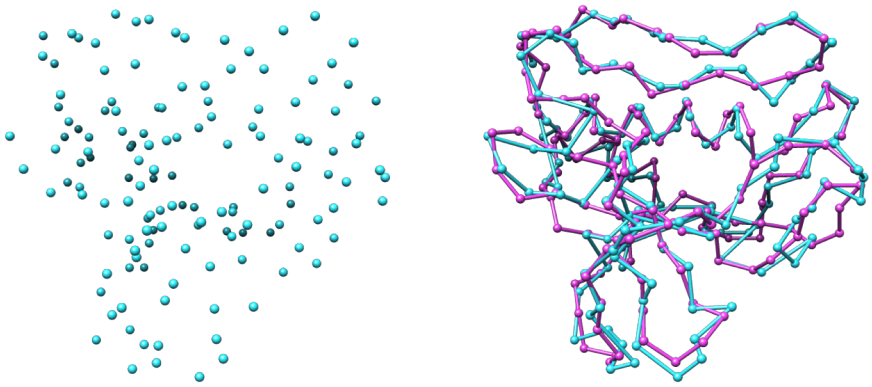

Figure 8: Traced backbone atoms. Predicted Cα atoms for the EMD-4054 density map in blue before (left) and after (right) the backbone tracing step compared to the deposited model structure in pink.

4.4 Amino Acid Sequence Mapping

To realize the side-chain prediction for the protein structure, we first need to know each amino acid's type. As discussed in Section 4.1, one output of the deep learning model is the amino acid type prediction. However, depending on the resolution of the density map, this prediction is of limited accuracy with around 10% to 50% since some amino acids have a similar appearance in electron density maps. The goal of this step is to improve the amino acid type accuracy by aligning intervals of the initially predicted sequence to the known true amino acid sequence (protein primary structure) and then updating the types of the predicted amino acids accordingly (see Figure 9).

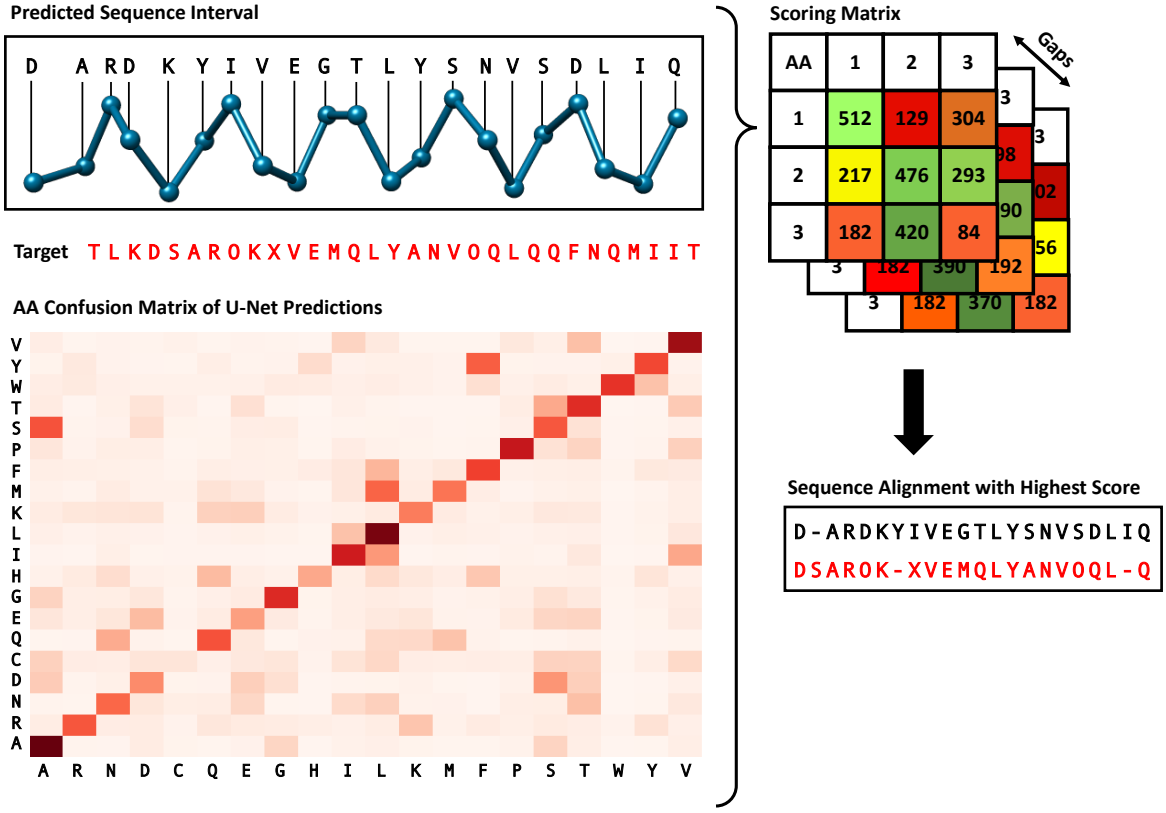

Figure 9: Protein sequence alignment algorithm. Interval of the predicted sequence is aligned with the target sequence using a custom dynamic algorithm. The amino acid confusion matrix depicts the relative frequency of pairs of predicted and true amino acid type and was calculated based on a set of test density maps. The numbers shown in the score matrix are solely for illustrative purposes and do not reflect real data.

Aligning amino acid sequences is a common problem in the field of bioinformatics, and previous research has led to the development of multiple algorithms [30, 31, 32]. However, these algorithms are usually applied between different proteins to measure their sequence similarities, which does not quite fit our use-case. The main problem is that we require an algorithm that does not treat all matches and mismatches in the

same way. This stems from the fact that some amino acid types have a more similar appearance in density maps than others, which leads to some mismatches of the U-Net being more likely than others. To analyze the relative frequency of a certain match of predicted and true amino acid type, we applied the U-Net to 200 different density maps and compared the predicted amino acid types with the actual types from the deposited model structures. The heatmap depicting this analysis is shown in Figure 9. As expected, the most frequent matches are those of the same predicted and true amino acid type. However, we can also see that the U-Net often mixes up some types (e.g., ALA and SER) and struggles more with other types (e.g., CYS).

To incorporate the U-Net prediction behavior described in the previous section into the alignment algorithm, we defined a reward function r which returns a score denoting how valuable a certain match of predicted type p and true type t is. With $f(p, t)$ defined as the relative frequency of a match, we constructed the reward function shown in Equation (1). The constant 100 as a multiplier is used to balance the match rewards with gap penalties described in the next section, and was chosen based on multiple rounds of testing. The 0.05 constant was chosen as this represents the likelihood of a correct match if we would chose the amino acid type randomly, since there are 20 different types of amino acids. The score is zero if the relative frequency equals this random likelihood.

$$r(t_p, t_t) = 100 \times (f(p, t) - 0.05) \tag{1}$$

In addition to the match reward, our algorithm also requires a gap penalty. A gap represents a skipped amino acid in either the predicted or true sequence. This penalty, however, cannot simply be a static value as not all gaps are the same. For example, gaps in the beginning of a sequence before any matches were made should not result in any penalties as we only match short intervals of the predicted sequence, meaning it is highly unlikely that they align at the first amino acid of the true sequence. Additionally, the number of consecutive gaps is important. Cases where DeepTracer misses an amino acid or predicts an extra amino acid appear relatively frequent meaning that a single gap is not unlikely. However, two missed amino acids in a row is very uncommon, and three gaps in a row virtually never happens. Therefore, we must define our penalty function p such that it takes the number of consecutive gaps g into account. Let i be the index of the amino acid that is not skipped. Then we can define p as shown in Equation (2). The constants 20 and 30 were chosen based on test runs to create a good balance with the rewards function.

$$p(g, i) = \begin{cases} 0, & \text{if } i = 0 \\ \infty, & \text{if } g \geq 3 \\ 20 + (g \times 30), & \text{otherwise} \end{cases} \tag{2}$$

Since we have defined a reward and penalty function, we can find the ideal alignment by maximizing the sum of all rewards and penalties using a dynamic algorithm. To do so, we defined a recursive equation which calculates the optimal solution based on an

index i, which points to the current amino acid in the true sequence, an index j, which points to the current amino acid in the predicted sequence, and g, which counts the number of previous consecutive gaps. With t and p as the true and predicted sequence, we defined this function as shown in Equation (3). To efficiently find the solution, we applied the dynamic programming "bottom up" approach [33].

$$\mathrm{OPT}(i,j,g) = \begin{cases} 0, & \text{if } i=0 \text{ or } j=0 \text{ or } g \geq 3 \\ \max\{\mathrm{OPT}(i-1,j-1,0) + r(t_i,p_j), & \\ \quad \mathrm{OPT}(i,j-1,g+1) + p(g,i), & \\ \quad \mathrm{OPT}(i-1,j,g+1) + p(g,j)\}, & \text{otherwise} \end{cases} \tag{3}$$

4.5 Carbon, Nitrogen, and Oxygen Prediction

So far, the predicted residues consist solely of Cα atoms. A complete protein backbone prediction also consists of carbon, nitrogen, and oxygen atoms. Previous research has introduced various methods for reconstruction of a protein backbone from a reduced representation, such as one contains only Cα atoms [34]. Instead of employing these theoretical methods, we chose to implement our own backbone reconstruction method to make use of the information captured from the 3D cryo-EM density maps. This section presents our all-atom backbone reconstruction method. This step is necessary for the next prediction step in the pipeline, resolving the side-chain atoms.

In addition to Cα prediction, the U-Net also provides information about carbon and nitrogen atoms in the confidence map predicted by the U-Net. We can use this information in combination with the previously predicted Cα atom positions to place the carbon and nitrogen atoms. Between the Cα atoms of two connected amino acids, there is always a nitrogen and carbon atom. Therefore, we can guess the initial position of these atoms by calculating the vector from one Cα atom to the other and then placing the nitrogen and carbon atoms at one third and two third of the distance of this vector. To refine these initial positions we calculated the center of mass around them in the carbon and nitrogen confidence maps. In Figure 10a we can see an example for the initial and refined prediction of the carbon and nitrogen atoms.

After the initial refinement, we can further refine the positions of the carbon and nitrogen atoms by applying well-known molecular mechanics of a peptide chain. We made several assumptions about the positions of carbon, nitrogen, oxygen atoms relative to the Cα atoms as seen in Figure 10b. First, we assumed the planar peptide geometry in which the Cα atom and carbon atom in the carbonyl group of an amino acid are in the same plane as the next amino acid's nitrogen and Cα atom [35]. Second, we constructed a virtual bond between the neighboring Cα atoms. The angles between this bond and $\mathrm{C}\alpha_{(i)}$–$\mathrm{C}_{(i)}$ bond (ϑ_2) and between this bond and $\mathrm{C}\alpha_{(i+1)}$–$\mathrm{N}_{(i+1)}$ bond (φ_2) are 20.9° and 14.9°, respectively [35]. Third, the peptide bonds in a protein are in the stable trans configuration [36].

To refine the position of the carbon atoms, we relied on the previous refinement. Let us call the unit vector pointing from $\mathrm{C}\alpha_{(i)}$ to $\mathrm{C}_{(i)refined}$ v_1, the unit vector pointing from $\mathrm{C}\alpha_{(i)}$ to $\mathrm{C}_{(i)}$ v_2, and the unit vector pointing from $\mathrm{C}\alpha_{(i)}$ to $\mathrm{C}\alpha_{(i+1)}$ v_3.

$$\begin{aligned} v_1 &=< a_1, a_2, a_3 > \\ v_2 &=< b_1, b_2, b_3 > \\ v_3 &=< c_1, c_2, c_3 > \end{aligned} \tag{4}$$

The goal is to solve for the components of v_1. Due to the planar peptide geometry, v_1, v_2, and v_3 exist in the same plane. Thus, their triple product equals to zero.

$$v_1 \times (v_2 \cdot v_3) = 0 \tag{5}$$

or

$$a_1(b_2c_3 - b_3c_2) - a_2(b_1c_3 - b_3c_1) + a_3(b_1c_2 - b_2c_1) = 0 \tag{6}$$

From this relation and the cross product of v_1 and v_2, and that of v_2, v_3, we can construct a system of equations:

$$\begin{cases} a_1b_1 + a_2b_2 + a_3b_3 = cos(\theta_2 - \theta_1) \\ a_1c_1 + a_2c_2 + a_3c_3 = cos(\theta_2) \\ a_1(b_2c_3 - b_3c_2) - a_2(b_1c_3 - b_3c_1) + a_3(b_1c_2 - b_2c_1) = 0 \end{cases} \tag{7}$$

Solving this system of equation yields a_1, a_2 and a_3. Next, the vector v_1 is scaled appropriately to resolve the new position of the carbon atom. The position of the nitrogen atom is refined in a similar manner.

To predict the location of the oxygen atom in the carbonyl group, we assumed the coplanar relationship between the oxygen, Cα, carbon, and nitrogen atom [35], and that the angle $A_{\alpha CO}$ and A_{OCN} (see Figure 10c) are approximately identical. We then derived a unit vector pointing in the direction of the C-O bond and scale it with the C-O bond length to get the position of the oxygen atom.

4.6 Side Chain Prediction

The final step of DeepTracer is the side chain prediction. Its goal is to position the side chain atoms of each amino acid based on its type and backbone structure. This is done by using SCWRL4 [37], a tool developed by the Dunbrack lab, which predicts side chain atoms for structures that have a complete backbone and amino acid types set. The tool is integrated in the prediction pipeline of DeepTracer and runs fully automatic as well. It also performs a collision detection to ensure that side-chains of different residues do not overlap. In Figure S10 we can see an example of an α-helix after the side chain prediction step.

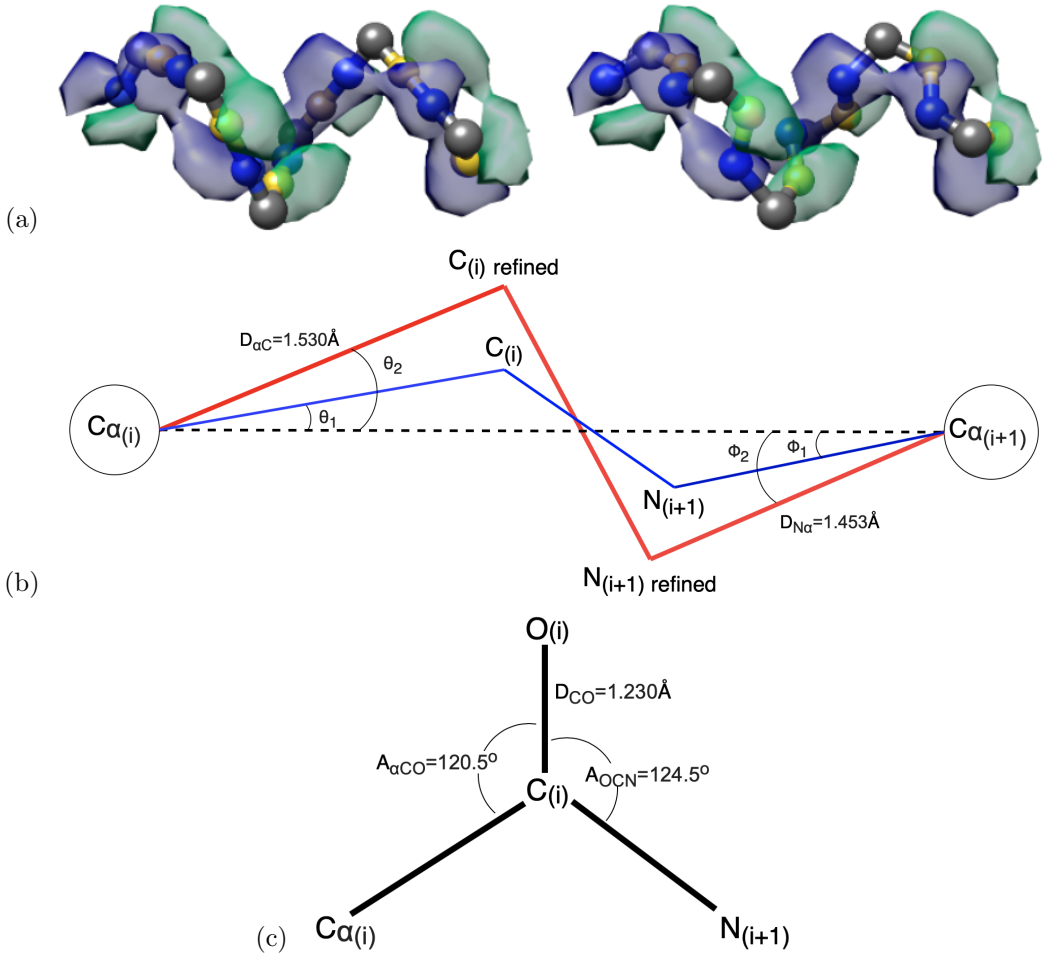

Figure 10: Carbon, nitrogen, and oxygen prediction. **(a),** Initial positioning of carbon (yellow) and nitrogen (blue) atoms in between the Cα atoms (gray) on the left and their initial refined positioning, which fits the U-Net prediction of carbon atoms (green volume) and nitrogen atoms (blue volume), on the right. **(b),** The positions of carbon and nitrogen atoms are refined further by forcing bond angles into their well-known values. The blue lines represent the bonds from the initial refinement. The red lines represent the bonds from the final refinement. **(c),** Position of oxygen atom in the carbonyl group by definition.

Acknowledgements

This research was funded by the National Science Foundation (award #2030381) and the graduate research award of Computing and Software Systems division at University of Washington Bothell. We thank Yinrui (Bobby) Deng and other DAIS group members for the frontend development and testing of DeepTracer.

References

[1] Carl Ivar Branden and John Tooze. *Introduction to protein structure*. Garland Science, 2012.

[2] Fredric S Cohen. How viruses invade cells. *Biophysical journal*, 110(5):1028–1032, 2016.

[3] Syed Faraz Ahmed, Ahmed A Quadeer, and Matthew R McKay. Preliminary identification of potential vaccine targets for the covid-19 coronavirus (sars-cov-2) based on sars-cov immunological studies. *Viruses*, 12(3):254, 2020.

[4] Meng Yuan, Nicholas C Wu, Xueyong Zhu, Chang-Chun D Lee, Ray TY So, Huibin Lv, Chris KP Mok, and Ian A Wilson. A highly conserved cryptic epitope in the receptor binding domains of sars-cov-2 and sars-cov. *Science*, 368(6491):630–633, 2020.

[5] Xiangyang Chi, Renhong Yan, Jun Zhang, Guanying Zhang, Yuanyuan Zhang, Meng Hao, Zhe Zhang, Pengfei Fan, Yunzhu Dong, Yilong Yang, et al. A neutralizing human antibody binds to the n-terminal domain of the spike protein of sars-cov-2. *Science*, 2020.

[6] Stefania Bambini and Rino Rappuoli. The use of genomics in microbial vaccine development. *Drug discovery today*, 14(5-6):252–260, 2009.

[7] Ewen Callaway. Revolutionary cryo-em is taking over structural biology. *Nature*, 578(7794):201–201, 2020.

[8] Xiao-Chen Bai, Greg McMullan, and Sjors HW Scheres. How cryo-em is revolutionizing structural biology. *Trends in biochemical sciences*, 40(1):49–57, 2015.

[9] E Callaway. 'it opens up a whole new universe': Revolutionary microscopy technique sees individual atoms for first time. *Nature*, 582(7811):156–157, 2020.

[10] Megan Scudellari. The sprint to solve coronavirus protein structures-and disarm them with drugs. *Nature*, 581(7808):252–255, 2020.

[11] Renhong Yan, Yuanyuan Zhang, Yaning Li, Lu Xia, Yingying Guo, and Qiang Zhou. Structural basis for the recognition of sars-cov-2 by full-length human ace2. *Science*, 367(6485):1444–1448, 2020.

[12] Wanchao Yin, Chunyou Mao, Xiaodong Luan, Dan-Dan Shen, Qingya Shen, Haixia Su, Xiaoxi Wang, Fulai Zhou, Wenfeng Zhao, Minqi Gao, et al. Structural basis for inhibition of the rna-dependent rna polymerase from sars-cov-2 by remdesivir. *Science*, 2020.

[13] Daniel Wrapp, Nianshuang Wang, Kizzmekia S Corbett, Jory A Goldsmith, Ching-Lin Hsieh, Olubukola Abiona, Barney S Graham, and Jason S McLellan. Cryo-em structure of the 2019-ncov spike in the prefusion conformation. *Science*, 367(6483):1260–1263, 2020.

[14] Thomas C Terwilliger, Paul D Adams, Pavel V Afonine, and Oleg V Sobolev. A fully automatic method yielding initial models from high-resolution cryo-electron microscopy maps. *Nature methods*, 15(11):905–908, 2018.

[15] Genki Terashi and Daisuke Kihara. De novo main-chain modeling for em maps using mainmast. *Nature communications*, 9(1):1–11, 2018.

[16] Brandon Frenz, Alexandra C Walls, Edward H Egelman, David Veesler, and Frank DiMaio. Rosettaes: a sampling strategy enabling automated interpretation of difficult cryo-em maps. *Nature methods*, 14(8):797–800, 2017.

[17] Dong Si, Spencer A Moritz, Jonas Pfab, Jie Hou, Renzhi Cao, Liguo Wang, Tianqi Wu, and Jianlin Cheng. Deep learning to predict protein backbone structure from high-resolution cryo-em density maps. *Scientific Reports*, 10(1):1–22, 2020.

[18] Dong Si, Shuiwang Ji, Kamal Al Nasr, and Jing He. A machine learning approach for the identification of protein secondary structure elements from electron cryo-microscopy density maps. *Biopolymers*, 97(9):698–708, 2012.

[19] Yuanchen Dong, Shuwen Zhang, Zhaolong Wu, Xuemei Li, Wei Li Wang, Yanan Zhu, Svetla Stoilova-McPhie, Ying Lu, Daniel Finley, and Youdong Mao. Cryo-em structures and dynamics of substrate-engaged human 26s proteasome. *Nature*, 565(7737):49–55, 2019.

[20] Kaiming Zhang, Huawei Zhang, Shanshan Li, Grigore D Pintilie, Tung-Chung Mou, Yuanzhu Gao, Qinfen Zhang, Henry van den Bedem, Michael F Schmid, Shannon Wing Ngor Au, et al. Cryo-em structures of helicobacter pylori vacuolating cytotoxin a oligomeric assemblies at near-atomic resolution. *Proceedings of the National Academy of Sciences*, 116(14):6800–6805, 2019.

[21] Comparison of ca positions in two models allowing any order of fragments. `https://www.phenix-online.org/documentation/reference/chain_comparison.html`. (Accessed on 07/19/2020).

[22] Adam Zemla. Lga: a method for finding 3d similarities in protein structures. *Nucleic acids research*, 31(13):3370–3374, 2003.

[23] Daniel A Keedy, Christopher J Williams, Jeffrey J Headd, W Bryan Arendall III, Vincent B Chen, Gary J Kapral, Robert A Gillespie, Jeremy N Block, Adam Zemla, David C Richardson, et al. The other 90% of the protein: Assessment beyond the cαs for casp8 template-based and high-accuracy models. *Proteins: Structure, Function, and Bioinformatics*, 77(S9):29–49, 2009.

[24] Coronavirus: Emdataresource. https://www.emdataresource.org/news/coronavirus_resources.html. (Accessed on 05/09/2020).

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[26] Emdataresource. https://www.emdataresource.org/. (Accessed on 03/04/2020).

[27] Rcsb pdb. https://www.rcsb.org/. (Accessed on 03/04/2020).

[28] Sandeep Chakraborty, Ravindra Venkatramani, Basuthkar J Rao, Bjarni Asgeirsson, and Abhaya M Dandekar. Protein structure quality assessment based on the distance profiles of consecutive backbone cα atoms. *F1000Research*, 2, 2013.

[29] Jack E Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1):25–30, 1965.

[30] Siavash Mirarab, Nam Nguyen, Sheng Guo, Li-San Wang, Junhyong Kim, and Tandy Warnow. Pasta: ultra-large multiple sequence alignment for nucleotide and amino-acid sequences. *Journal of Computational Biology*, 22(5):377–386, 2015.

[31] Desmond G Higgins and Paul M Sharp. Clustal: a package for performing multiple sequence alignment on a microcomputer. *Gene*, 73(1):237–244, 1988.

[32] Chuong B Do, Mahathi SP Mahabhashyam, Michael Brudno, and Serafim Batzoglou. Probcons: Probabilistic consistency-based multiple sequence alignment. *Genome research*, 15(2):330–340, 2005.

[33] David B Wagner. Dynamic programming. *The Mathematica Journal*, 5(4):42–51, 1995.

[34] Aleksandra E. Badaczewska-Dawid, Andrzej Kolinski, and Sebastian Kmiecik. Computational reconstruction of atomistic protein structures from coarse-grained models. *Computational and Structural Biotechnology Journal*, 18:162–176, 2020.

[35] Yoriko Iwata, Atsushi Kasuya, and Shuichi Miyamoto. An efficient method for reconstructing protein backbones from α-carbon coordinates. *Journal of Molecular Graphics and Modelling*, 21(2):119–128, 2002.

[36] Scott W. Robinson, Avid M. Afzal, and David P. Leader. Chapter 13 - bioinformatics: Concepts, methods, and data. In Sandosh Padmanabhan, editor, *Handbook of Pharmacogenomics and Stratified Medicine*, pages 259 – 287. Academic Press, San Diego, 2014.

[37] Georgii G Krivov, Maxim V Shapovalov, and Roland L Dunbrack Jr. Improved prediction of protein side-chain conformations with scwrl4. *Proteins: Structure, Function, and Bioinformatics*, 77(4):778–795, 2009.

[38] Eric F Pettersen, Thomas D Goddard, Conrad C Huang, Gregory S Couch, Daniel M Greenblatt, Elaine C Meng, and Thomas E Ferrin. Ucsf chimera—a visualization system for exploratory research and analysis. *Journal of computational chemistry*, 25(13):1605–1612, 2004.

[39] vop. `https://www.cgl.ucsf.edu/chimera/docs/UsersGuide/midas/vop.html#resample`. (Accessed on 03/01/2020).