Template-based protein structure modelling

Olha Kholod, Carlos Martinez Villar, Farhan Quadir, Sriganesh Pera, Md. Bodruzzaman Sarker

Abstract

We present a template-based computational framework to predict 3D structure of the protein from its sequence. We choose bond distances as constraints for our model. Then, we generate probability density functions (pdfs) and maximize them with logarithmic function. We optimized predicted Ca-Ca distances with gradient descent method. We performed 703 iterations, resulting in logpdf = 26309 and error = 1.04e-05. We have plotted predicted target coordinates and superimposed them with a template protein. We evaluated our model using TM-score and RMSD metric. The source code along with project documentation is publically available on GitHub template-based-modeling.

Introduction

Protein structure-prediction algorithms consist of template-based models and *ab initio* models. Template-based models employ various restrains, such as bond length, Ψ and φ angles, ect. Different tools have been designed due to date for template-based protein structure modeling, including Modeller and MTMG. Although they have a number of limitations, these approaches still work quite efficiently, when template structures are available. In order to predict protein spatial coordinates more precisely, a gradient descent optimization might be beneficial. The advantage of gradient descent method is that it tweaks model parameters iteratively to minimize a given function to its local minimum. Here, we designed a template-based approach relying on gradient descent that employ bond length (Ca-Ca distances) as constraints to predict a structure of T0951 protein target.

Methods

A) Data selection

The protein sequence T0951 had been selected as a target for template-based protein structure prediction from CASP13 target list. This sequence corresponds to 3D protein structure 5z82.pdb in the PDB Data Bank.

We performed blast search [1] in order to identify a suitable template for a target sequence. We have discarded sequences that share more than 90% similarity with a

target sequence. Thus, we select the protein 4i1a, which share 60.59% of sequence similarity with a target sequence.

B) Libraries and software tools

We employed the following Python libraries: numpy, math, Bio.PDB and matplotlib. We perform alignment procedure using Modeller tool [2].

C) Initialization

First, we defined Ca-Ca distances as a square root of the sum of the squares of the differences between corresponding coordinates. Then, we defined probability density functions (pdfs) using the following formula:

$$f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

, where σ corresponds to dispersion, μ corresponds to distance between amino acid residues in the template and x corresponds to distance between amino acid residues in the target. Then, we initialized arrays for the following data structures: number of target points, number of template points, target Ca-Ca distances, template Ca-Ca distances and pdfs.

D) Restrain extraction and representation

- extracting Ca-Ca distances;

We computed Ca-Ca distances for both template and target sequences. We have extracted distances between each pair of amino acid residues (e.g. [1-2, 1-3, 1-4, ect]). The bond distance between neighboring amino acid residues was ~ 3.8 Å. We stored these distances in target and template distance arrays, respectively.

- probability density function (pdf) calculation;

We computed pdfs using the following function and store obtained values in an array:

```
pdf_array[i,j] = pdf[target_dist_array[i,j], template_dist_array[i,j], σ)
```

We did not set the weights for a pdf function, because our prediction based on a single template. We set arbitrary value for σ = 0.5. After computing these pdfs for each pairwise distance, we multiply these values (excluding values on the diagonal) and maximize its value with logarithmic function.

```
product_pdf = prod(pdf_array_excuding_diag[pdf_array_excuding_diag > 0])
```

```
log_product_pdf = -log(product_pdf)
```

E) Gradient descent optimization

First, we defined gradient for target points, target distance array and pdfs in order to construct a vector that helps to pinpoint direction to the local minimum. Then we calculate gradient for each target coordinate and target distances, respectively

```
grad_x[i]=grad_dist[i,i+1]*(target_points[i,0]-target_points[i+1,0])/target_dist_array[i,i+1]
grad_y[i]=grad_dist[i,i+1]*(target_points[i,0]-target_points[i+1,0])/target_dist_array[i,i+1]
grad_z[i]=grad_dist[i,i+1]*(target_points[i,0]-target_points[i+1,0])/target_dist_array[i,i+1]
```

Then, we build a gradient descent function to adjust target coordinates in a way that minimize the error of the predicted protein structure:

```
target_points = - alpha*gradient_points
error = abs(log_pdf - new_log_pdf)
```

The iterative process will stop, when the error function reach value lower then 10^{-5}. After this, we obtained three-dimensional array, where each point corresponds to target coordinates. We optimized only points that were aligned with the template structure, using Modeller tool.

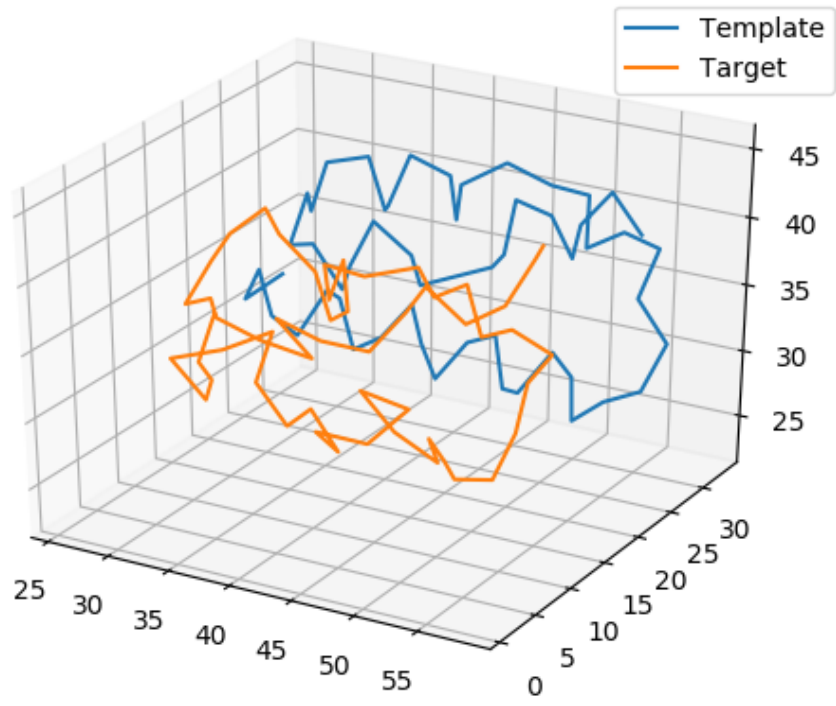
Results and Discussion

A) Distance matrix for protein sequences

0	3.80354861	5.25631982	5.21633712	6.61498367	8.50955557	10.04463	10.8140367	12.6723028	14.6265519	15.1495024	16.7478824
3.80354861	0	3.80195397	5.43371779	5.2973953	5.86864363	8.511825	9.77057383	10.5642437	12.4376328	13.8290149	15.2995608
5.25631982	3.80195397	0	3.79538918	5.59456048	4.88216817	5.950608	8.32283191	9.76050987	10.6080428	11.778059	14.0727708
5.21633712	5.43371779	3.79538918	0	3.80499934	5.3240602	5.288231	6.02269408	8.61113918	10.0894529	10.1065781	12.1163933
6.61498367	5.2973953	5.59456048	3.80499934	0	3.81313375	5.646052	4.99302954	6.22689762	8.8815904	9.45678725	10.3952904
8.50955557	5.86864363	4.88216817	5.3240602	3.81313375	0	3.800763	5.34636783	5.19498171	6.59188357	8.48540948	10.0672326
10.0446275	8.51182501	5.95060812	5.28823052	5.64605225	3.80076269	0	3.81987775	5.37425809	5.16325769	5.90832794	8.66614978
10.8140367	9.77057383	8.32283191	6.02269408	4.99302954	5.34636783	3.819878	0	3.81569941	5.49047685	4.63012926	6.09547882
12.6723028	10.5642437	9.76050987	8.61113918	6.22689762	5.19498171	5.374258	3.81569941	0	3.7908102	5.13792069	5.18848668
14.6265519	12.4376328	10.6080428	10.0894529	8.8815904	6.59188357	5.163258	5.49047685	3.7908102	0	3.78854867	5.58407826
15.1495024	13.8290149	11.778059	10.1065781	9.45678725	8.48540948	5.908328	4.63012926	5.13792069	3.78854867	0	3.80158428
16.7478824	15.2995608	14.0727708	12.1163933	10.3952904	10.0672326	8.66615	6.09547882	5.18848668	5.58407826	3.80158428	0

B) Modeling of predicted protein structure

We employed Python graphical library to visualize both target and corresponding template chains. The results presented on the following graph:



C) Evaluation

We evaluate the predicted structure using a TM-score, applying the following formula [3]:

$$\text{TM-score} = \max\left[\frac{1}{L_{\text{target}}}\sum_{i}^{L_{\text{aligned}}}\frac{1}{1+\left(\frac{d_i}{d_0(L_{\text{target}})}\right)^2}\right]$$

, where L_{target} and $L_{aligned}$ are the lengths of the target protein and the aligned region respectively. d_i is the distance between the ith pair of residues and

$$d_0(L_{\text{target}}) = 1.24\sqrt[3]{L_{\text{target}} - 15} - 1.8$$

TM-score takes as arguments the coordinates of two structures - predicted target coordinates and native protein structure coordinates - iteratively calculate score for each aligned region and returns the maximum score. We also calculated root-mean-square deviation between the predicted target structure and native target structure. The distance function is used for calculating the distance between ith-residue in predicted and native structures and returns a square root of the mean of these distances. The estimated TM-score was … .

Conclusions

The proposed approach is capable of predicting 3D structure of the protein, using Ca-Ca distances as constraints to the model. The implemented gradient descent optimization allows to model aligned regions with high precision. The future work includes more elaborative validation study, as well as incorporating various constraints, such as angles and Cb-Cb distances.

References

1. https://blast.ncbi.nlm.nih.gov
2. https://salilab.org/modeller/
3. https://en.wikipedia.org/wiki/Template_modeling_score