Protein

Jingchao Zhou

January 26, 2023

Protein 1

1. Ideas behind the Recurrent Geometric Network (RGN)

The RGN is an end-to-end model , and the input are amino acid sequences and PSSMs that correspond to the adjacent units. It has three different stages, the first stage is using RNN to generate each residue's 3 torsional angles ϕ, ψ, ω. In the first stage, this model implements a gating units chain on the amino acids and PSSMs. Each unit is based on the LSTM [1], and it gets information from the one residue of the amino acids sequence and PSSMs. Each unit also can get information from nearby units and can output the information to the other nearby units. In this way, the units can learn geometric information from the whole amino acid sequence. Each unit will output its residue's torsional angles as the input of the second stage.

Then the second stage get the input from the first stage (torsional angles), and inputs from upstream units in the second stage(partial backbone) and extends the backbone by one residue. Then out put the new backbone to the downstream units.

After all geometric units finish their tasks, The last unit of the second stage will output the predicted protein structure into the third stage. The third stage will use the dRMSD as the metric to calculate the distance between the predicted structure and the experimental structure.

The RGN2 only uses the amino acid sequence as the input of this algorithm. So it enables we can get a structure of an amino acid sequence without knowing the PSSMs. It implements the AminoBERT to get the latent information from the amino acid sequence. This AminoBERT is based on the BERT[2], and it treats one residue as a token in this algorithm. To training this AminoBERT, they masked some residues and let this model to reconstruct it. The training procedure is independent from the geometric part and in a self-supervised manner.

They use the discrete version of the Frenet–Serret formulas for one dimensional curve to represent the geometric property of each residue. This parameterization method is rotationally and translationally invariant.

Finally, they use AF2Rank-based protocol[3] to impute the backbone and sidechain atoms. Then this model uses dRMSD metric to calculate the distance between the predicted structure and the experimental structure.

2. Differences and similarities with the approach taken by AlphaFold2

The similarities between RGN and AlphaFold2 is that, they are both end-to-end model. However, the RGN has 2 inputs, but AlphaFold2 has 3 inputs, single input, pairwise input, and templates, using MSA technique. The RGN2 has only one input. The AlphaFold2 implements the attention technique to calculate pairwise representation, however, the RGN2 uses the attention technique to learn the latent information in the whole amino acid sequence. The RGN2 uses the Frenet-Serret Formulas to represent the geometric properties of residues, then uses AF2Rank-based protocol to generate the 3d structure. The methods to generate the 3d structure are same, however, RGN2 train the geometric properties first, then use the generated 3d structure from the geometric properties to calculate the loss.

3. The claimed advantages of the RGN2 method with respect to AlphaFold2

RGN2 represents one of the first attempts to use ML to predict protein structure from a single sequence. It is the first sequence-to-structure methods without any evolutionary information. RGN2 is in a manner that is more similar to the physical process of protein folding than MSA methods(AlphaFold2), so it would give some idea to the prediction of the protein folding process. The RGN2 is much more faster than the AlphaFold2, both in prediction running time and refining running time. RGN2 has better prediction on average accuracy however AlphaFold2 has better performance on the orphan proteins and de-novo proteins. RGN2 has better performance on Alpha-helical targets.

Protein 2

The idea

I implement a class that has 2 functions, one is download the PDB file and parse it to structure, the other one is find the regions with specified quality.

I got the b-factor list of the polypeptide first. I draw the plot of each position's atom with its b-factor. By observing the

Then I used the k-means Cluster to find the 3 different qualities with the input is the polypeptide. See Appendix 1.

I choose factor-b because factor-b indicates the how disordered the parts of the structure are. See Appendix 2.

Functions

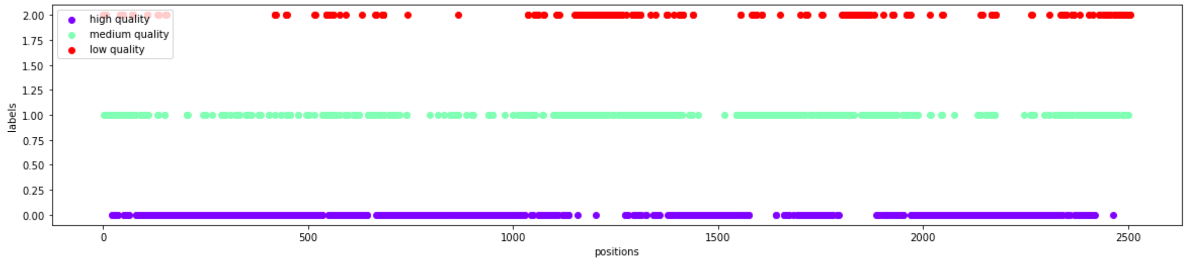
`download_pdb(*pdbid*)` : pdbid is the string that user decided to download. Return `Bio.PDB.Structure`

`consecutive(*data*, *stepsize*=1)`:`data`, the input list. `stepsize`, the consecutive step size we want. Return `List`, the list of consecutive lists with a given step size in the input list.

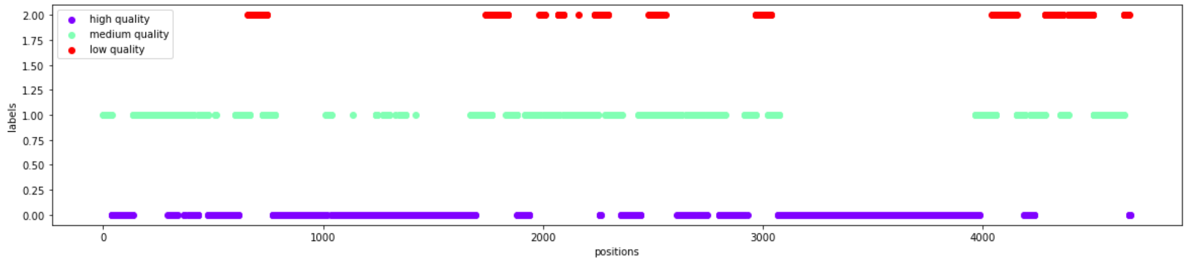
`get_region_with_quality(*self*,*structure*, *region_quality* = "low")`: `structure`, is an object of `Bio.PDB.Structure`, the structure we want to find the quality. `region_quality`, this parameter indicates the quality type we interested in, must be chosen between $\{''low'',''medium'',''high''\}$. Return `List` and `List`, the list of regions with name of AAs and a list of regions with positions.

Examples

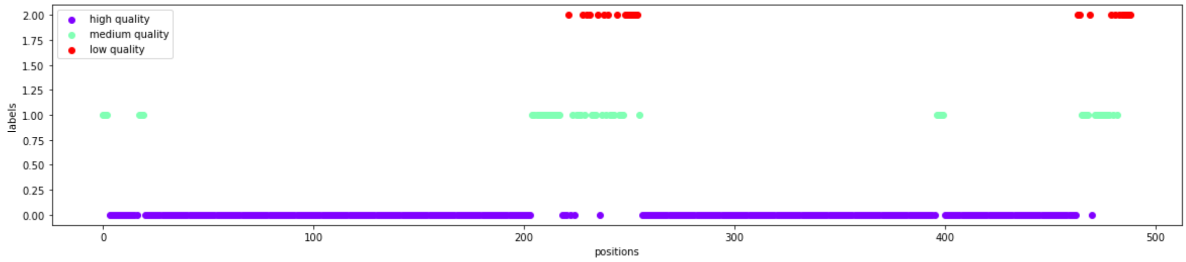
I tried the "2DN1" PDB file, and I plot a scatter for each quality.



Then I tried the "5VVM"



and "100D"



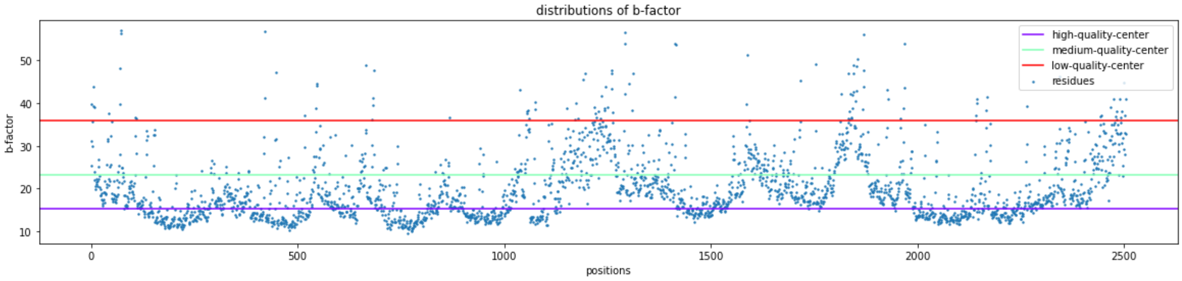
I also upload the .ipynb documents that include the plots.

Reference

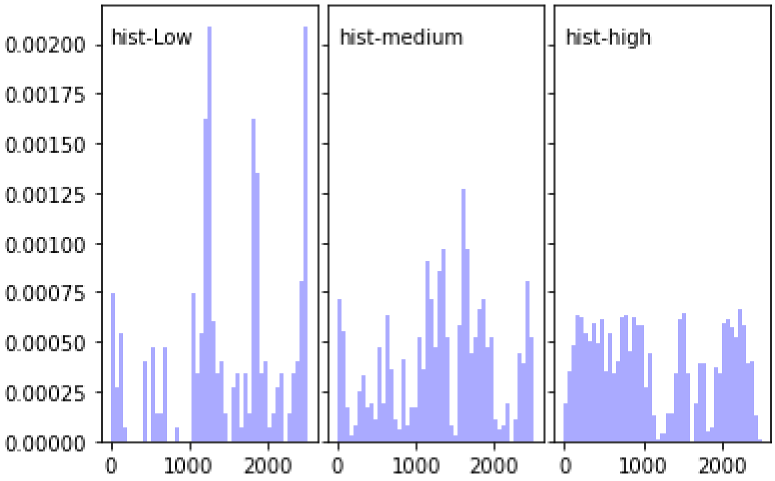
- [1] Chowdhury, R., Bouatta, N., Biswas, S. *et al.* Single-sequence protein structure prediction using a language model and deep learning. *Nat Biotechnol* **40**, 1617–1623 (2022). https://doi.org/10.1038/s41587-022-01432-w
- [2]BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al., NAACL 2019)
- [3]State-of-the-art estimation of protein model accuracy using AlphaFold. James P. Roney, Sergey Ovchinnikov. bioRxiv 2022.03.11.484043; doi: https://doi.org/10.1101/2022.03.11.484043

Appendix

1. The figure of the cluster center of each quality, with the scatter plots residues and its b-factor.



2. The figure of the hist graph of each quality. The x is position, and y is frequency.


hist-Low
hist-medium
hist-high