The instructions for protein secondary structure element assignment program P2PSSE

Lincong Wang*

September 27, 2020

1 Introduction

Given a protein structure P2PSSE is a program for the assignment of its protein secondary structure elements (SSEs) using the geometrical data extracted from its *pc*-polyline *and* a convolutional neuron network (CNN) model trained using the same type of data. The close relationship between *pc*-polyline and protein secondary structure has been described previously [1] and the P2PSSE program is described in a paper submitted for publication. The program itself consists of two sub-programs, the first one P2P computes the *pc*-polyline for a structure and extracts its geometrical features (data) to be used for both the training of CNN models and SSE assignment. The second sub-program, P2PASSIGN, assigns the SSEs for a protein structure using a trained CNN-model and its p2p geometrical data. In the following we will first describe the P2P program and then the P2PASSIGN program.

2 The P2P program

2.1 The preparations required by P2P

The input to P2P is a protein structure in PDF format. Protons are added using the program REDUCE [2] to any PDB that lacks their coordinates. The program still works without protonation but the center of such a peptide plane will differ slightly from that with an amide proton. Though it is not required in theory the current version assigns Charmm force field parameters [3] to the structure. Furthermore it uses the standard amino acids and an upper limit for a peptide bond length to check the structure for possible missing atoms or gaps[1]. A gap may result from the chemical modifications of any backbone atoms of an amino acid. At present any chain with any gaps is excluded. If there exist more than one form in a chain only the most frequent one is selected for the computation of *pc*-polyline and its geometrical features.

2.2 The usage of P2P

P2P program is written in C++, and compiled and tested on Ubuntu 18.04 with gcc 7.5.0 and ubuntu 20.04 with gcc 9.3.0. It uses GSL (https://www.gnu.org/software/gsl/) for the computation of a peptide plane center. The current version is compiled using GSL-2.5 with the GSL libraries installed in /usr/local/lib.

To use P2P in command line two arguments are required. The first one is the path to the directory where the two Charmm files, "par_all27_prot_na.prm" and "top_all27_prot_na_correct.top", are located. The second one is a protonated protein structure. For example one could run P2P in the current directory as follows.

```
./p2p    /home/lincong/bioParam/charmm    2FR3FH.pdb
```

where "/home/lincong/bioParam/charmm" is the path to the Charmm file directory and "2FR3FH.pdb" is the PDB file in the current directory.

*Lincong Wang, Email: wanglincong@jlu.edu.cn, wlincong@gmail.com.

[1] A gap in a protein chain means that either one or several consecutive interior residues have no ATOM statement in the PDB file.

A typical output for a chain without any gaps is as follows.

```
************ PROcess 2FR3FH.pdb *************
– READ 2FR3FH.pdb ——
        chainid=A Has multiple forms
– ANAlyze Structure: ASSign FF parameters / CONstruct molGraph —-
          Has 99 missing atoms:
        Chain A OK !

– COMpute p2p polyline ——
         Save to: 2FR3FH_A_p2p.dat
```

where "2FR3FH_A_p2p.dat" is output file that saves the computed geometrical features.

A typical output for a structure of multiple chains with gaps is as follows.

```
************ PROcess b5bkFH.pdb *************
– READ b5bkFH.pdb ——
        chainid=B Has multiple forms
        chainid=A Has multiple forms
– ANAlyze Structure: ASSign FF parameters / CONstruct molGraph —-
            has NO N-terminus NHs !
        Chain A OK !

        has NO N-terminus NHs !
            ^^ Peptide bond too long: Bond Length=11.0015
            Possible gap between LEU174 – ASP193
          Has 18 missing atoms:
        Chain B OK !

– COMpute p2p polyline ——
          Save to: b5bkFH_A_p2p.dat
        !! chainID=B has GAP
```

If each chain in a structure has a gap then no output file is written.

2.3 The geometrical data from a *pc*-polyline

The output from P2P program is the geometrical data for each residue in a structure[2]. The data include three parts: (a) the p2p distances (d_{pp}s) between pairs of peptide plane centers that are smaller than a pre-specified threshold T_{pp} (Table 1), (b) the sequence distances (s_{pp}s) between the corresponding pairs of residues (Table 2), and (c) five angles computed using six consecutive peptide centers (Table 3). For a residue the number of pair-wise p2p distances is determined by its location in the protein and by T_{pp}. Typical value for T_{pp} is 9.0Å. To make data uniform for all the residues in a structure if $d_{pp} > T_{pp}$ we set $d_{pp} = 9.001$ (Table 1) and set the corresponding $s_{pp} = 0$ (Table 2).

A:P1	3.230	5.927	7.658	8.205	8.291	8.550	8.584	8.593	8.734
		8.867	9.001	9.001	9.001	9.001	9.001	9.001	

Table 1: **The p2p distance $d_{pp}(i, j)$s for residue A:P1 in structure 2FR3**. The second row continues those in the first row. If $d_{pp} > T_{pp}$ then $d_{pp} = 9.001$ where a threshold of $T_{pp} = 9.0$ is used. The first column lists residue ID and the second one lists $d_{pp}(i, i + 1)$ values. The unit is Å.

2	89	44	90	45	111	3	46	43	0	0	0	0	0	0

Table 2: **The corresponding sequence distance $s_{pp}(i, j)$s for residue A:P1 in structure 2FR3**. The sequence distance $s_{pp}(i, j)$ between two residues i and j is defined as $s_{pp}(i, j) = j - i$ where i, j are residue numbers. If $d_{pp} > T_{pp}$ then $s_{pp} = 0$.

[2]No p2p geometrical data is computed for the last two residues of the C-terminus.

$\theta(i, i+1)$	$\theta(i, i+2)$	$\theta(i, i+3)$	$\theta(i, i+4)$	$\theta(i, i+5)$
16.664	40.275	21.539	54.555	70.432

Table 3: **The five angles for residue A:P1 in structure 2FR3**. The angle $\theta(i, k)$ is between $\mathbf{v}_{pp}(i, i+1)$ and five other $\mathbf{v}_{pp}(k, k+1)$s where $k - i = 1, 2, 3, 4, 5$ where $\mathbf{v}_{pp}(i, i+1)$ denotes the vector from peptide plane center i to peptide plane center j. The angle unit is degree.

3 The training of CNN models and the P2PASSIGN program

Among a dozen of CNN models trained using different parameters the one that produces SSE assignments that agree best with those by DSSP [4] is "best_asg0921_8A5z.h5". These CNN models were originally trained using TensorFlow 1.4 with GPU support on a Dell T5810 workstation equipped with Nvidia RTX2080. The results described in the submitted paper were obtained using this particular model. The python script for SSE assignment has been updated for TensorFlow 2.3 on both Ubuntu 18.04 and Ubuntu 20.04.

3.1 The P2PASSIGN program

The P2PASSIGN program is a simple Python script, *p2pAssign.py*, that uses the TensorFlow-trained CNN model to assign the SSEs with the geometric data computed by P2P as its input. For example, in a directory with the best CNN model , "best_asg0921_8A5z.h5", the assignment is obtained by typing following command.

```
python3   p2pAssign.py   2FR3FH_A_p2p.dat
```

where "2FR3FH_A_p2p.dat" is the input file computed by P2P. The output will be "2FR3FH_A_p2p.asg" in this case. Except for the last part of the file that lists the SSE content for the structure, each line of the output file corresponds to the assignment of a residue to one of the six SSE types. For example the assignment for the first four residues of the structure 2FR3 are listed in Table 4.

A:P1	0.0	0.0	0.0	0.0	0.0	1.0000	U/e
A:N2	0.0	0.0	0.0	0.0	0.0	0.9999	U/e
A:F3	0.0	0.0191	0.0060	0.0	0.0	0.9749	U/e
A:S4	0.0	0.9995	0.0	0.0	0.0	0.0	E/u

Table 4: **The assignments by P2PASSIGN program for the first four residues of structure 2FR3**. Each line has 8 columns. The 1st column lists residue IDs, they are the same as those in the input file. The next six columns list respectively the probabilities for being a "H, G, E, B, T, U" SSE type. The last column lists the respective assignments to one of the six SSE types of the column (before /) with the highest probability and the column with the second highest probability (after /).

References

[1] Lincong Wang, Yao Zhang, and Shuxue Zou. The characterization of pc-polylines representing protein backbones. *Proteins: Structure, Function, and Bioinformatics*, 88(2):307–318, 2020.

[2] J. M. Word, S. C. Lovell, J. S. Richardson, and D. C. Richardson. Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation1. *Journal of Molecular Biology*, 285(4):1735–1747, 1999.

[3] A. D. MacKerell, D. Bashford, Bellott, R. L. Dunbrack, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The Journal of Physical Chemistry B*, 102(18):3586–3616, 1998.

[4] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.