User Manual of SUPRB (SUboptimal PRoBabilistic threading)

SUPRB is a threading prediction method designed from suboptimal alignment algorithm. This document is written to explain how to use this program. All necessary files mentioned here can be found at the webpage: http://dragon.bio.purdue.edu/suprb/.

(1) Generation of input files for SUPRB

Before running SUPRB, we need to prepare all input files for it. The following explain how to generate these input files.

Sequence file: Although sequence file can be obtained from many sources, I suggested that the template sequence be extracted from its PDB file. By doing so, it would be easier to maintain coincidence among all input files. **Note: Any mismatch among input files will crash SUBWAI and it will take long time to find where this mismatch is.** To extract the template sequence, first run this line:

```
/PDBSeq.pl template.ent (Chain) > output.seq                    (Step 1)
```

PDBSeq.pl is a perl script to extract the sequence of wanted chain from the ENT structure file (ENT file follows PDB format and you can get this kind of file from PDB website). The first input parameter of PDBSeq.pl is the filename of the ENT file (e.g. pdb1afi.ent) . The second is optional: Chain name (You can omit it if no chain you want to specify). Chain name need to be capital (e.g. A). If you just want to get the sequence of one chain in the structure file, then specify it and otherwise the script will extract all sequence of all chains. The default output of PDBSeq.pl is the standard output, so you have to redirect the output by “>” to the output file “output.seq” (The output can be any arbitrary name with the affix “.seq”.)

The target sequence can be get by other source rather than the PDB file. However, it need to be in FASTA format.

Profile file: Once we have the sequence file of the target and the template, run the following to generate PSI-BLAST file:

```
./blastpgp -d nr -i sequence_file -o output_file -m 6 -j 5 -e 0.002 -h 0.002 –FILter
```

(Step 2)

Here “–m 6” is required and otherwise in the next step, msa.pl cannot correctly recognize the PSI-BLAST file. All other option can be changed according to your need. “-d” provide the database name in which you make PSI-BLAST search. I use NCBI non-redundant database here. “-i” provide the input sequence file and “-o” points out which file to store the output. “-m” describe which format the output file uses. “-j” set the maximal iteration of PSI-BLAST. “-e” set the e-value cutoff for displaying in the output. “-h” set the e-value threshold for inclusion. “-FILter” is to filter out low-complexity region.

After the PSI-BLAST file is done, run the following command:

```
./msa.pl input_PSIBLAST_file
```

(Step 3)

Here *input_PSIBLAST_file* is the *output_file* in Step 2. Msa.pl script will read the PSIBLAST file and generate the output profile in FA format. The path of the output profile is defined inside msa.pl, so please read the head of msa.pl and modify the path to where you want.

Secondary structure and solvent accessibility (SSSA) file: The SSSA file of the target is generated in this way. First, copy the target sequence file to the directory where SABLE is and rename it as data.seq:

```
cp target_sequence /sable/data.seq                                  (Step 4)
```

then in the directory of SABLE, run:

```
./run.sable                                                         (Step 5)
```

After the script is done, copy the output file to your destination and rename:

```
cp /sable/ OUT_SABLE/OUT_SABLE_graph your_destination/output.sable    (Step 6)
```

The SSSA file of the template is generated by DSSP. To do this, the wanted structure need to be extracted from the ENT file. The ENT file contain the structure information of all chains. However, usually we only work on one chain so we should extract the structure of this chain from the whole ENT file. Otherwise, it may cause mismatch and crash the program. To extract the chain, run

```
./PDBChain.pl input_ENT_file (Chain) output_PDB_file                (Step 7)
```

Input_ENT_file is the ENT file from which the chain is extracted and ouput_PDB_file is the output file in which the information of the chain is stored. The option of “Chain” is the chain name. If no chain exists in the ENT file, then just omit this option. Then the output_PDB_file is fed to DSSP to generate the SSSA file of the template.

```
./dsspcmbi output_PDB_file output.dssp
```

(Step 8)

The first file is the PDB file of extracted chain and output.dssp is the output file of DSSP.

Torsion angle file: Torsion angle file is generated from the output_PDB_file of the template by the following command:

```
./angle output_PDB_file > out.angle
```

(Step 9)

Here angle is the executable file to calculate angles on the template. Output_PDB_file is gotten from PDBChain.pl. The default output of angle is the standard output so you should redirect the output to where you want by “>”.

Contact file: The contact file is generated from the output_PDB_file, too. First, run:

```
./listcSHA output_PDB_file 4.5 Chain > output.contact
```

(Step 10)

listcSHA is the executable file to calculate contacts around each residue of the template. Output_PDB_file is gotten from PDBChain.pl. 4.5 is the distance threshold for a contact and you can change it in your case. Chain is the chain name and if no chain is specified, put 0 there. The output is redirected to the file “output.contact”.

Then run:

```
./ctc.pl
```

(Step 11)

This will transform “output.contact” gotten from above to LIST format. Read the head of “ctc.pl” and you can modify where the input files are and where the output are.

(2) Compilation and usage of SUPRB

The main program of SUPRB is written in C. It has two versions, corresponding to two different strategies of using contact potentials: suprb_I.cpp for the reranking strategy and suprb_II.cpp for the probablistic contact strategy. The executable SUBWAI can be compiled by the following instruction:

```
g++ suprb_(I or II).cpp –o suprb_(I or II)
```

(Step 12)

The output executable file is *suprb_(I or II)***,** which is provided for downloading. To generate suboptimal alignments between target and template, run this command:

```
./suprb_(I or II) -am OO_AA_rel_10.pot -q target.seq -t template.seq -pq target.fa -pt
template.fa -sq target.sable -st template.dssp -at template.angle -ct template.list
```

(Step 13)

In the following, we are coming to what the options in above command means.

Options:

The options and their input files are listed as follows(The order doesn’t matter):

-am: The torsion angle potential matrix. Now, we use OO_AA_rel_10.pot but other matrices can be used here with slight modification in source codes.

-q: The target(query) sequence file. FASTA format is required.

-t: The template sequence file. FASTA format is required.

-pq: The target(query) profile. FA format is required.

-pt: The template profile. FA format is required.

-sq: The secondary structure and solvent accessibility of the target (query) predicted from the sequence. This file is generated by SABLE.

-st: The secondary structure and solvent accessibility of the template retrieved from the experimental structure. This file is generated by DSSP.

-at: The torsion angle on each residue of the template. ANGLE format is required.

-ct: The contacts on each residue of the template. LIST format is required.

There are some other option which may be used sometimes:

-mt: Maximal iteration in Strategy II. You should put a number after it. Currently, we use 5.

-r: Structural alignment between the target and the template. If you use this option, the program will calculate ALDs between the optimal alignment and the structural alignment and output them. CE format is required.

(3) Outputs of SUPRB

Output files:

The execution of SUPRB generates three files: output.dat, confidence.dat and distance.dat. In the following, we will show how these output files look like and what information they record.

Output.dat: This file contains all generated suboptimal alignments in plain text. The format looks like the following (Only the head of the example file, are displayed. Here Sequence A is 1afi in PDB and Sequence B is 1aps):

```
We get 436 suboptimal pathways.
The No.1 point new score is 19.116.
The position of this point in the matrix should be x=28, y=33.
The best alignment going through this point is:
A:----ATQTVTLAVPGMTCAACPI-TVKKALSKVEGVSK-VDVGFEKREAVVTFDDTKASVQKLTKATAD
AGYP-SSVKQ---------------------.
B:STARPLKSVDYEVFGRVQGVCFRMYAEDEARKIG-VVGWVK-NTSKGTVTGQVQGPEEKVNSMKSWLSK
VGSPSSRIDRTNFSNEKTISKLEYSNFSVRY.

The No.2 point new score is 19.0654.
The position of this point in the matrix should be x=2, y=6.
The best alignment going through this point is:
A:----ATQTVTLAVPGMTCAACPI-TVKKALSKVEGVSK-VDVGFEKREAVVTFDDTKASVQKLTKATAD
AGYP-SSVKQ---------------------.
B:STARPLKSVDYEVFGRVQGVCFRMYAEDEARKI-GVVGWVK-NTSKGTVTGQVQGPEEKVNSMKSWLSK
VGSPSSRIDRTNFSNEKTISKLEYSNFSVRY.

The No.3 point new score is 18.9955.
The position of this point in the matrix should be x=65, y=70.
The best alignment going through this point is:
A:----ATQTVTLAVPGMTCAACPI-TVKKALSKVEGVSK-VDVGFEKREAVVTFDDTKASVQKLTKATAD
AG-YPSSVKQ---------------------.
B:STARPLKSVDYEVFGRVQGVCFRMYAEDEARKI-GVVGWVK-NTSKGTVTGQVQGPEEKVNSMKSWLSK
VGSPSSRIDRTNFSNEKTISKLEYSNFSVRY.
```

……

Appedix Figure 1. The format of the output.dat file.

The first line depicts how many suboptimal alignments SUPRB generated. This number is set as 0.1*M*N, here M is the length of the target sequence and N is the length of the template. The second line records the alignment rank and the alignment score. The third line provides the starting point of this alignment in the matching matrix of suboptimal dynamic programming. The fifth and sixth line show the alignment and A is the target and B is the template. Line 2-6 record the information of the optimal alignment which has the highest alignment score. Line 8-12 record the suboptimal alignment with the second highest alignment score and so on.

Confidence.dat: This files contain local SPAD(SuboPtimal Alignment Diversity) scores for **the optimal alignment which has the highest alignment score**. The format is like (Partial display and it is for the pair of 1afi and 1aps as well as Appendix Figure 1):

```
0.473578
0.468085
0.481743
0.478142
0.571303
0.509333
0.475768
0.78602
0.503796
0.481078
0.86921
0.821494
0.765931
0.564151
0.536822
0.497604
0.607305
0.495844
…...
```

Appendix Figure 2. The format of the confidence.dat file.

One line in this file records local SPAD for one residue pair in the best alignment from N terminal to C terminal. The global SPAD can be gotten by averaging all local SPAD scores.

Distance.dat: If the structural alignment is provided by –r option, distance.dat will be generated. This file contains ALD(ALignment Distance) between the best alignment and the structural alignment. The format is like (Partial display. It is for the pair of 1afi and 1aps):

```
-1
-1
-1
-1
1
1
1
1
1
1
1
```

Appendix Figure 3. The format of the distance.dat file.

One line record local ALD for one residue pair in the best alignment. -1 here means there is no structural alignment in this position.