1. **Prepare the prerequisite files**

All the .pdb files and the “modules.py” file should be in the same directory as the AmphiScan.py file. If additional parameters that require an input are to be analyzed in the post-processing step (i.e. resolution list, membrane thickness, whether the loop is a free loop table), these files should be put in a folder named “metric\_list” where the file name is separated from the given parameter by a comma.

1. **Run zα calculations**

The script takes as an input a single .pdb file. The following command line was used to run the zα calculations:

python3.6 AmphiScan.py -input\_pdb filename.pdb -thickness\_from\_file -keep\_starting\_tilt

These parameters set the final α angle to the default 180°. The default depth scan range of 10 Å from each face of the membrane surface is retained. The final β angle is set to 0 because only rotations of the α angle are considered. Because the β (tilt) angle from the input structures were used for the calculations with the natural helices, the “-keep\_starting\_tilt” option was used. If the user wants to run a calculation where the helix is parallel to the membrane surface, then this command should be omitted. Thicknesses are read from the provided “membrane\_thickness.txt” file under the “metric\_data” folder.

The calculation generates a “results” folder under which the results from multiple .pdb files will be stored in a folder under the name “filename”. Each “filename” folder has three folders in it: csv, output\_pdbs, and txt. The csv folder contains a file(s) in .csv format that has three columns. The first column is the depth, the second column is the α angle, and the third column is the score calculated at these parameters. Separate .csv files are generated for each tilt angle. The output\_pdbs folder contains the best pose(s) calculated for the run. Again, a separate .pdb file is generated for each tilt angle for which the calculation was done. Finally, the .txt folder contains .txt file(s) for each tilt angle that has a user-readable form of the scores calculated at different depth and α angles. The final two lines of the file summarize the best z and α values belonging to the lowest score and the time it took to run the calculation.

1. **Run zαβ calculations**

The following command line was used to run the zαβ calculations:

python3.6 AmphiScan.py -input\_pdb filename.pdb -y\_final 90 -thickness\_from\_file

These parameters set the final α angle to 180° and the final β angle to 90°. The default depth scan range of 10 Å from each face of the membrane surface is retained. The increments used for the scan are 1 for the α- and β- angles, and a depth increment of 0.1 Å. Thicknesses are read from the provided “membrane\_thickness.txt” file under the “metric\_data” folder. The output format is the same as described for the zα calculations. The ‘-keep\_starting\_tilt’ option was omitted since all the β angles were calculated starting with a β value of 0.

1. **Post-processing of the results**

The results of the AmphiScan calculations were analyzed with two scripts called for the zα and zαβ calculations separately using the following command prompts:

python 3.6 post\_processing.py

python3.6 post\_processing\_xyz.py

whereby the difference between the two scripts is that the latter script calculates the lowest-scoring β angle among all the generated results files and uses that file for the analyses.

The script in its current form analyzes multiple parameters and outputs two kinds of files. The first type is a results file in the form of a table that contains information that starts with the helix name and the RMSD values, and has further information on the calculated parameters as defined in the post-processing script. Some of the parameters calculated in our calculations were charge, database tilt angle and distance from the membrane center, ψ and φ angle deviations from ideal values, number of prolines, whether the helix is a terminal one, residue frequencies, etc. The user can edit the post processing script to calculate any other values pertaining to the calculations, or write their own script.

The second type is information on the membrane-embedded residue identities based on the transformed starting coordinates from a database. For each .pdb file, three files are written under the corresponding txt folder. The first one is “embedded\_all.txt” that has a list of the membrane-embedded residues calculated based on the AmphiScan calculations. The second one is “embedded\_opm.txt” that has a list of the membrane-embedded residues calculated from the database (in this case OPM) input file. The third file is “check\_accuracy.txt” that has information about the accuracy and MCC values calculated for the helix based on the calculated and database membrane-embedded residues.

**Notes**

The script has additional commands that can be used to run additional operations on the .pdb files. These are namely:

-show\_pymol: When this option is used, the pymol mover is activated by the AmphiScan script, which allows for real-time visualization of all the operations carried by it. Note that pymol should be open and the “PyMOL-RosettaServer.py” script must be run before this command is passed.

-keep\_history: This option allows stacking of the individual frames visualized with pymol so that a trajectory or a movie of the scan process can be generated.

-repack\_helix: This option repacks the helix at every rotation step.

-relax\_helix: This option relaxes helix at every rotation step. Note that relax may shift the position of the helix and therefore the zαβ values may not match that of the final relaxed helix geometry. Do not use this option with the ‘repack\_helix’ option.

-helix\_range: The AmphiScan protocol was developed to run calculations with single helices. However, sometimes amphipathic helices may be part of peptides that consists of multiple helical regions separated by kinks or turns. This command addresses the need to run AmphiScan using one helix as the reference point. It takes a range (i.e. 1-20) and defines the helix region based on the user’s input, therefore all the transformation operations are done only on this part of the peptide.

If you are running a system for the first time, visually check the transformation steps and the calculated angles to ensure that the transformation is done accurately. In some cases, structural imperfections may result in shifts from the desired starting α and β values that affect the rest of the calculations (i.e. transformation cannot force a parallel alignment with respect to the given axis). In these cases, the user can manually set the rotation angles to ensure correct starting behavior.