# User Guide

This document outlines the data preparation and implementation of the restricted Peaceman-Rachford splitting method (rPRSM) for the protein side-chain positioning (SCP) problem, as presented in the paper titled `A Peaceman-Rachford Splitting Method for the Protein Side-Chain Positioning Problem'.

## Table of Contents



## Data Preparation

Data preparation is done by executing the Python script: `rotamerCalculationsV8.py`. This script will require a class definition that is in the script: `shellNeighbors.py`. These two Python scripts are placed in `` `data preparation' `` folder.

Both scripts should be in some directory of your choosing.

**The rotamerCalculationsV8.py script must be executed in the Chimera environment.**

See: https://en.wikipedia.org/wiki/UCSF_Chimera for information related to Chimera.

After installing Chimera, use the Tools/General Controls menu to launch an IDLE Python Shell window. Then use the File/Open… menu of the shell window to open the `rotamerCalculationsV8.py` script.

The Run menu of the IDLE window can then be invoked to run the script.

### Running rotamerCalculationsV8.py

If you inspect the mainline section of the script, there is a sequence of lines that define the Protein Data Bank identification numbers for the PDB files that were used in the paper. The script will fetch a protein file from the Protein Data Bank and will then start generating the data output files. During this execution you should observe the application generating various rotamer conformations for adjacent residue pairs while it evaluates the interaction energy for each pairing.

### Output Files

The script will generate two output files: `` `xxxxdata.txt' `` and `` `xxxxsurvivorCount.txt' ``.

The prefix xxxx will be the PDB ID.

Each line in the `` `xxxxdata.txt' `` file contains 6 numbers:

- The line number.
- The position number of a residue (amino acid) in the PDB file.
- The index of a rotamer for this residue.
- The position number of a neighboring residue.
- The index of a rotamer for this neighboring residue.
- The interaction energy for these two residues when they adopt the specified rotamer settings.

The `xxxxsurvivorCount.txt' file contains extra information that indicates, for each residue, the rotamer count before and after Dead End Elimination (DEE) is performed. The typical line will contain:

- The position number of a residue (amino acid) in the PDB file.
- The residue type.
- The number of rotamers corresponding to that residue.

## Data Folder

The folder `data' contains the 131 PDB datasets tested with in the paper. The `xxxxdata.txt' files within this folder are generated using the methods described in the Data Preparation section above. The `131ProteinList.txt' lists the Protein Data Bank identification numbers ordered by protein amino acid count.

## Solver

The restricted Peaceman-Rachford splitting method (rPRSM) for the protein side-chain positioning (SCP) problem is implemented in MATLAB programing language.

### Running main.m

The script `main.m' is used for selecting problem instances and calls the rPRSM algorithm. A user inputs include:

- probnumbers : an integral array containing numbers between 1 and 131. Each element in probnumbers corresponds to the Protein Data Bank identification number listed in `131ProteinList.txt'.
- datafolder : a path to the location of the `xxxxdata.txt' files and `131ProteinList.txt'.
- tablefilename : file name of the report
- opts : a struct for rPRSM options. Only problem independent parameters listed in `main.m'. Problem dependent parameters such as $\beta$, maxiter and initial iterates are defined in `run_test.m' .

The script `main.m' then calls `run_test.m' to

1. prepare the rPRSM inputs;
2. call the rPRSM algorithm (PRSM_protein.m);
3. output the numerical report in Latex format that can be compiled in Latex to produce the reports as seen in the Appendix of the paper. The output filename is currently set 'tabletest.tex'.

### Solver Folder

This folder includes `PRSM_protein.m', which implements Algorithm 3.1 from the paper. It utilizes inputs prepared in `run_tests.m' described above. The folder `utils' contains functions used for preparing inputs and executing the algorithm.