baseline

Ella van Loenen, Liesbet Ooghe, Eric van Huizen

January 2024

1 Introduction

In the project of Protein Pow(d)er, we need to fold the amino acids of various proteins to make the protein as stable as possible. The stability of a protein is measured by the number of amino acids of certain types (first type "H" and later also type "C") located next to each other but not connected in the protein.

Figures 1 and **??** show examples of folded proteins. 1a is a visualisation in the terminal, of a protein with only H and P-type amino acids. 1b is a plotted version of another protein that also contains C-type amino acids.

2 Baseline Algorithm

Initially, the protein as input is in a string of "H"s and "P"s. The baseline algorithm follows a few steps to fold this randomly:

1. Each of the characters in the string is added to the dictionary 'aminos' with as key its index and as value an Amino class object.
    - The indices start at 0, and are also kept in a separate list.
    - the Amino class object is initialised with its type, direction 0, coordinates (0, 0), and the previous Amino class object.
    - the previous amino of the first amino is a None object.
2. The protein class object is put into the random algorithm.
    (a) The first amino acid gets a direction and keeps coordinates (0, 0).

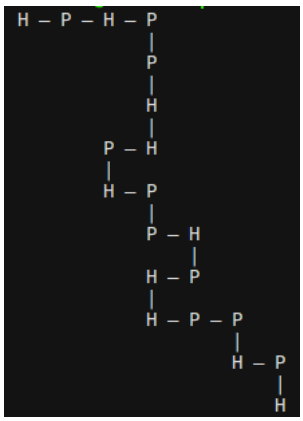

Figure 1: Visualisation in terminal of randomly folded "HPHPPHHPHPPHPHHPPHPH"

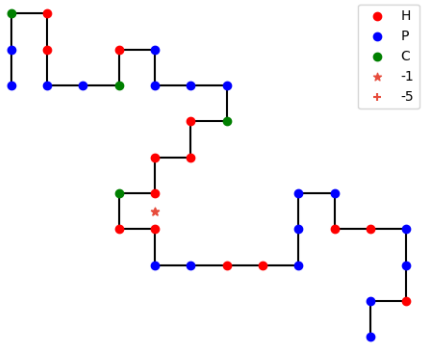

Figure 2: Plot of randomly folded "PPCHHPPCHPPPPCHHHHCHHPPHHPPPPHHPPHPP"

```
if abs(self.previous_amino.direction) == 1:
    self.coordinates = [self.previous_amino.coordinates[0] + self.previous_amino.direction, self.previous_amino.coordinates[1]]
elif abs(self.previous_amino.direction) == 2:
    self.coordinates = [self.previous_amino.coordinates[0], self.previous_amino.coordinates[1] + (int(self.previous_amino.direction / 2))]
```

Figure 3: Code for changing the coordinates of an amino acid

(b) Any other amino acid gets a random direction and its coordinates are changed according to the coordinates and direction of the previous amino acid, according to the code as shown in figure 3

(c) The last amino gets direction 0 and its coordinates are still updated.

3. The coordinates of all amino acids are checked to see whether none of them overlap. If they do, the random algorithm as described in step 2 is repeated, so a wholly new protein folding is done.

4. The stability of the protein is calculated by counting per H-type amino acid how many other H-type amino acids are placed next to it based on the coordinates but do not have adjacent indices.

   - Per H-H bond, the stability score decreases by 1. The stability should be as high as possible negative number.
   - When C-type amino acids are also included, H-H bonds and C-H bonds have a stability score of -1 and C-C bonds have a stability score of -5. Any bond between any of the types of bonds get a score of -1 and then the number of C-C bonds get an additional -4 score.

3 Results

The results of the distribution of this algorithm being applied 1000 times to the protein "HPHP-PHHPHPPHPHHPPHPH" are shown in figure 4.

The results of the distribution of this algorithm being applied 10 times to the protein "PPCHH-PPCHPPPPCHHHHCHHPPHHPPPPHHPPHPP" are shown in figure 5.

The algorithm took too long if a larger protein was used or the number of trials was increased, so these figures (mostly figure 5) are not a very reliable representation of the distribution of stabilities of randomly folded proteins.

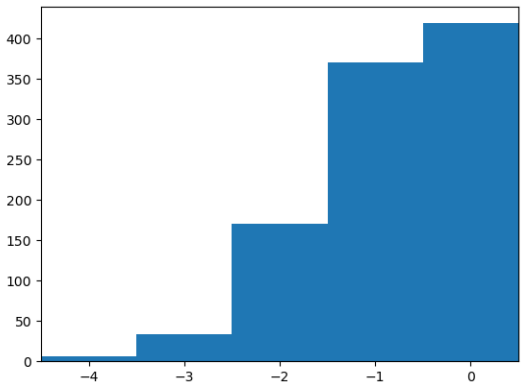

Figure 4: 1000 trials of the random algorithm on protein "HPHPPHHPHPPHPHHPPHPH"

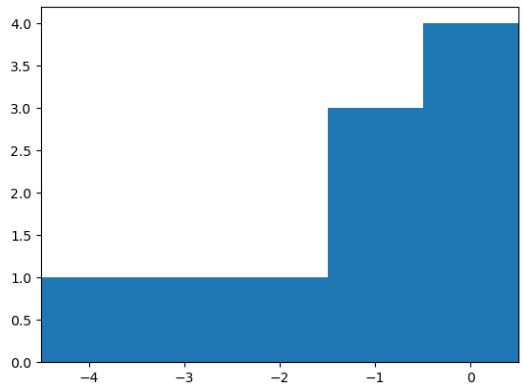

Figure 5: 10 trials of the random algorithm on protein "PPCHHPPCHPPPPCHHHHCHHPPHH-PPPPHHPPHPP"