Frequency of Different Amino Acid Sequences in Protein-Protein Interactions

COMS7059 Project

Ruan Pretorius (790674)

October 2020

1 Introduction

Proteins are macromolecules that influence cell behaviour in many living organisms. These proteins are made up of smaller molecules called amino acids. These amino acids are in turn made up of atoms.

The amino acids that make up a protein molecule are chained together in a linear sequence. This linear sequence of amino acids is referred to as the primary structure of a protein.

However, proteins fold into complex three-dimensional shapes due to the chemical and physical properties of their constituent amino acids and do not resemble their primary structures. This three-dimensional structure of a folded protein is called its tertiary structure.

Protein-protein interactions largely determine the functions of cells and are even used in drug-development. For the purpose of this project, locations of protein-protein interactions are defined as locations (in the tertiary structure) where two sections of one or more proteins are in close proximity.

More specifically, these interactions are defined as locations where the centroid of one amino acid comes within a distance of θ ångström (Å) of the centroid of another amino acid. These two amino acids can be on different protein chains or, if they are on the same chain, it will only be considered an interaction if they are separated by at least 20 amino acids in the primary structure.

Centroids of amino acids are defined as the average location of their constituent atoms which are assumed to have point locations. The centroid $(\bar{x}, \bar{y}, \bar{z})$ of an amino acid can be calculated using equation 1.

$$(\bar{x}, \bar{y}, \bar{z}) = \left(\frac{1}{n} \sum_{i=0}^{n-1} x_i, \frac{1}{n} \sum_{i=0}^{n-1} y_i, \frac{1}{n} \sum_{i=0}^{n-1} z_i \right) \tag{1}$$

Where x_i, y_i, and z_i are the x, y, and z coordinates of the i^{th} atom respectively.

The aim of this project was to test the hypothesis that certain sequences of k consecutive amino acids are more likely to be involved in protein-protein interactions than others.

The rest of this document is structured as follows. Section 2 describes the data set used for experimentation. Section 3 then describes the methodology followed to test the hypothesis. This is followed by an exposition and discussion of the results in section 4 and a conclusion in section 5.

2 Data Set

The data set used in this project consisted of a number of PDB files. These were files from the Protein Data Bank [1] that contain the primary and tertiary structure of proteins determined by methods such as X-ray crystallography.

Inside each PDB file was a list of all atoms making up the protein. The three-dimensional location of each atom was also contained in the file along with its amino acid name, sequence number, and protein chain name.

3 Methodology

3.1 Instruments

Because a large amount of PDB files had to be processed for this project, distributed memory processing was used on the university's cluster (Wits Core Cluster). This was chosen so that the same algorithm could be implemented on different files (SIMD) by different worker nodes on the cluster.

The set of files chosen for analysis were scattered between different worker nodes by the head node of the cluster. After analysis, the results from each worker node were gathered by the head node for amalgamation and final analysis.

The Wits Core Cluster consists of about 45 worker nodes with more than 1000 hyper-threaded cores. Each worker node has between 24GB and 1TB of RAM and there is a total of 1PB of globally addressable disk space. The scheduling system used on this cluster is SLURM.

For this project, the mpi4py library was used with Python 3.7 for managing the scattering and gathering of PDB files bwetween worker nodes. Some intermediate processing was done using the pandas library; the PDB files were parsed using the prody library; and finally, some operations of the numpy library supporting OpenMP was also used for multi-threaded parallelism.

3.2 Algorithm

The algorithm used to list the sequence of k interacting amino acids within a distance θ Åin a PDB file is described in Algorithm 1. After parsing a PDB file using the prody library to get the coordinates of atoms names of protein chains, and names of amino acids, the centroid of each amino acid was calculated using equation 1. An empty list was then initialised to store the interacting sequences that were found.

For every amino acid in the file, the Euclidean distance was calculated between its centroid and every other amino acid's centroid. Whenever an amino acids was found within the the cut-off distance θÅof another one (and further than 20 amino acids away in the primary structure if on the same chain), the neighbouring k amino acids were appended along with that amino acid to the list of interacting sequences.

One small detail left out of Algorithm 1 for simplicity reasons is the way in which the neighbouring k amino acids were selected. When an interacting amino acid was found in the middle of a protein chain, an equal amount of neighbouring amino acids were selected from each side to make up a sequence of k amino acids. If an amino acid was found at the end of a chain that satisfied the interaction criteria, the last k amino acids of the chain were appended to the interacting sequence list.

Algorithm 1: Interacting Amino Acid Sequences $(file, k, \theta)$

```
parse PDB file to extract atoms, coordinates, protein chain names;
calculate centroid (x̄, ȳ, z̄) of each amino acid in file (eq. 1);
initialise empty sequence list seq = [ ];
for amino acid i do // for all amino acids in file
    for amino acid j ≠ i do // for all amino acids in file
        Euclidean distance d_E = sqrt((x̄_i − x̄_j)^2 + (ȳ_i − ȳ_j)^2 + (z̄_i − z̄_j)^2) ;
        linear structure distance d_L // if i, j on same chain;
        if (d_E ≤ θ) and (d_L ≤ 20) then
            append k neighbouring amino acids of i to seq;
        end
    end
end
return seq
```

In order to assess the accuracy of this algorithm, a small PDB file was created with a specific structure to verify that the expected result was returned (see section 4).

3.3 Performance Evaluation

In order to select the optimal amount of nodes and threads to use for the final analysis, a range of different values were tested on small sets of 24 and 54 PDB files.

The execution of these analyses on these small sub-sets of data were timed and recorded, then plotted to visualise the effects.

From these timed executions, the empirical speed-up $S(n)$ and efficiency $E(n)$ were calculated for different numbers of nodes n and threads using equations 2 and 3 respectively.

$$S(n) = \frac{T(1)}{T(n)} \tag{2}$$

Where $T(n)$ time taken to complete a process using n nodes. This means $T(1)$ is sequential processing.

$$E(n) = \frac{S(n)}{n} \tag{3}$$

The efficiency $E(n)$ was used to assess the marginal gains in speed-up attained by using additional nodes.

3.4 Final Analysis

After the optimal configuration was determined in terms of the number of nodes and threads to use, a final analysis was conducted on a much larger data set consisting of 870 PDB files in order to test the hypothesis.

This final analysis was done using the optimal parallel configuration for different values of k and θ.

After visually inspecting some tertiary structures of proteins (see examples in Figure 1), it seemed that a reasonable interaction cut-off distance would be between $\theta = 5$Åand $\theta = 10$Å. Therefore, the final analysis was conducted for both $\theta = 6$Åand $\theta = 10$Å. Tests were done for all combinations of $\theta = \{6, 10\}$Åand $k = \{4, 5, 6\}$.

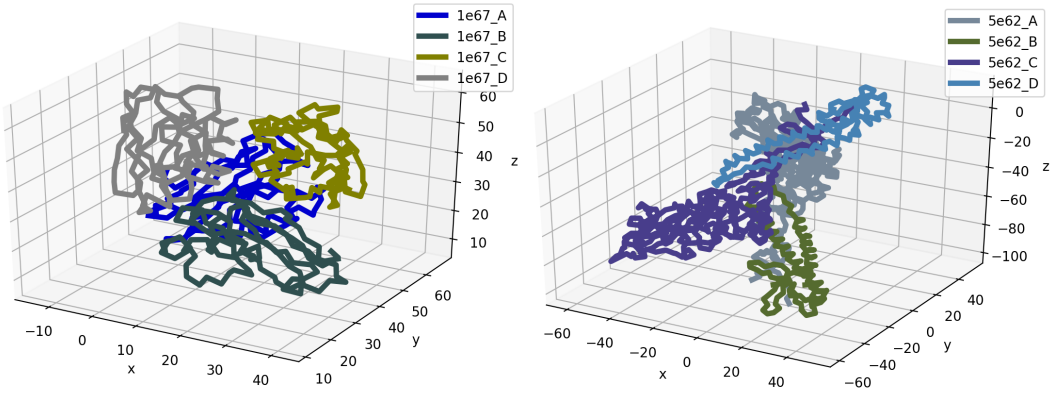

Figure 1: Example visualisations of tertiary structures of interacting protein chains

After the lists of k interacting sequences of amino acids were determined for each PDB file assigned to a worker node according to Algorithm 1, they were sent back to the head node.

Upon receiving all the lists of interacting amino acids from the worker nodes, the head node counted the occurrences of each unique amino acid sequence.

As a final result, the top 25 amino acid sequences based on count was then returned along with a percentage representing the fraction of the total interactions each sequence were involved in.

4 Results and Discussion

4.1 Algorithm Accuracy Evaluation

A visualisation of the amino acid centroids of the small PDB file used to assess the accuracy of the algorithm can be seen in Figure 2. It shows two protein chains that interacted at two locations. Chain A interacting with itself by looping back and chain B in close proximity above it. Note that this was a simple two-dimensional case with all amino acids in the same z-plane.

This correctness test was done using $k = 2$ and $\theta = 6$Å. With these parameters, the expected result was four interactions each of ALA-ALA, GLY-GLY, LYS-LYS, and SER-SER sequences. The actual results matched expectation and can be seen in Table 1. This verified the correctness of the algorithm.

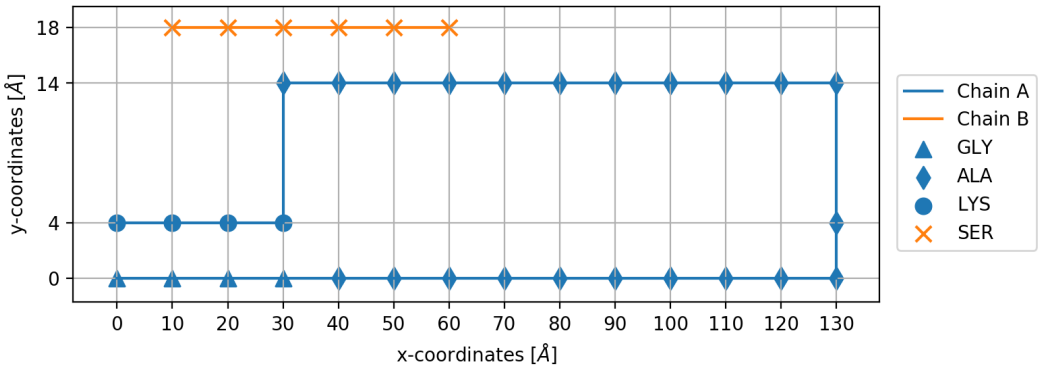

Figure 2: Visualisation of the amino acid centroids and protein chains of the small PDB file used to assess the accuracy of Algorithm 1

Table 1: Interacting Amino Acid Sequence Results of Algorithm 1 Correctness Test

Sequence	**Count**	**%**
GLY-GLY	4	25.0
ALA-ALA	4	25.0
SER-SER	4	25.0
LYS-LYS	4	25.0

4.2 Parallelism Performance Evaluation

The results of the time taken to get the sequences of interacting amino acids on small data sets of 24 and 54 files can be seen in Figure 3. For these tests, the numbers of $k = 2$ and $\theta = 6$Åwere arbitrarily selected and kept constant.

Note that the time taken for the calculations on the data set of 54 files was scaled by a factor of 0.15. This was done to get the time scales on the same level so that the trends were easier to identify.

These results show that the number of threads used did have an impact on the computation time. Using a single thread led to the worst performance. However, after increasing the amount of threads beyond two, little to no performance increase was gained. In some cases the performance even decreased for both data sets.

For this reason, the optimal number of threads selected for the final analysis was two.

The computation time $T(n)$ for calculating the interacting amino acid sequences with different amounts of nodes n was recorded again for the two data sets of 24 and 54 PDB files. The speed-up $S(n)$ and efficiency $E(n)$ was then calculated using equations 2 and 3 respectively. These results are plotted in Figure 4.

These results show that the amount of nodes used for computation had an effect on the computation time (and therefore also on the speed-up and efficiency). After increasing the amount of nodes beyond six, no significant speed-up was achieved. This was likely because of an I/O bottleneck or time overhead incurred due to coordinating the distributed computing beyond this point.

The I/O bottleneck could have occurred due to transferring the PDB files to

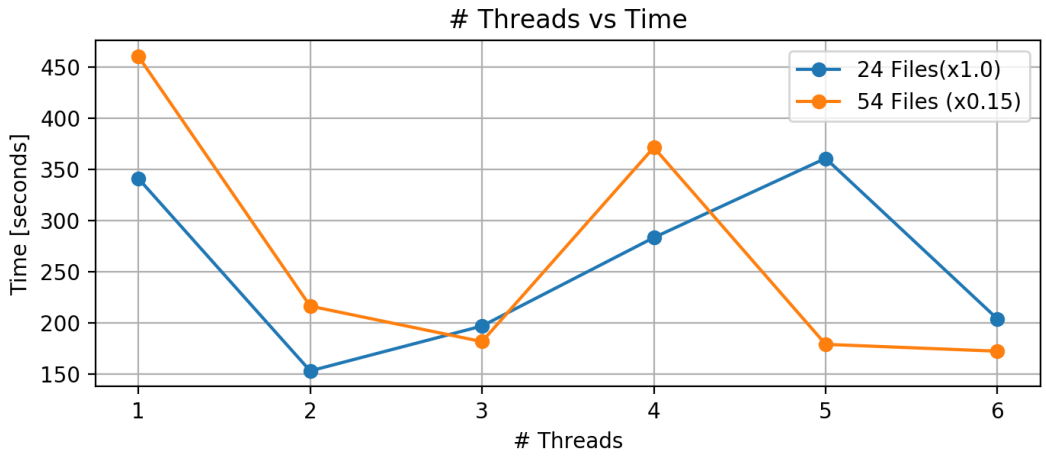

Figure 3: Time to calculate interacting amino acid sequences for different amounts of threads on 24 and 54 BDP files (note $k = 2$; $\theta = 6$Å; times were scaled for different data set sizes for ease of comparison)

the worker nodes for analysis. However, In all experiments, when running the `top` command, CPU usage of between 97% and 100% was reported. This suggests that no CPUs were idle while waiting for data, making the coordination of parallel computing the more likely reason for the performance reduction beyond six nodes.

A maximum speed-up of around three times was achieved when using six and 12 nodes. Because the efficiency started to plateau after using more than four nodes, the number of nodes selected for the final analysis was $n = 6$. This resulted in an efficiency of around 40%.

4.3 Final Analysis Results

After experimentally determining the optimal parallel computing configuration of six nodes and two threads, the final analysis was done. This consisted of using the accuracy-verified Algorithm 1 on a larger data set of 870 PDB files to find longer amino acid sequences of $k = \{4, 5, 6\}$ with interaction cut-off distances of $\theta = \{6, 10\}$Å.

The mean time taken to compute the interacting sequences for all combinations of $k = \{4, 5, 6\}$ and $\theta = \{6, 10\}$Åon the data set of 870 PDB files was 13.61 hours with a standard deviation of 3.22 hours.

For reasons of brevity only the results of $k = 6$ with interaction cut-off distances of $\theta = \{6, 10\}$Åare shown here.

Table 2 shows the top 25 interacting sequences of amino acids (by count) for $\theta = \{6, 10\}$Å. This data is also shown visually in Figures 5 and 6.

Looking at the results for both $\theta = 6$Åand $\theta = 10$Å, it seems like there were indeed some sequences of amino acids that were involved in more protein-protein interactions.

This is especially clear in Figure 5 where it seems as though the top four sequences are involved in significantly more interactions. In Figure 6 there is a more gradual decline in interactions among the top four to six sequences.

Table 2: Interacting Amino Acid Sequence Results of Final Analysis on 870 PDB Files with $k = 6$, $\theta = \{6, 10\}$Å

$\theta = 6$Å			$\theta = 10$Å		
Sequence	**Count**	**%**	**Sequence**	**Count**	**%**
ASN-TYR-ALA-ASP-PHE-ASP	135	0.5	TYR-ALA-ASP-PHE-ASP-TYR	622	1.9
ASN-ASN-TYR-ALA-ASP-PHE	125	0.5	ASN-ASN-TYR-ALA-ASP-PHE	541	1.6
LEU-GLY-ARG-GLU-THR-ALA	110	0.4	ASN-TYR-ALA-ASP-PHE-ASP	525	1.6
TYR-LEU-ASN-SER-SER-GLY	95	0.4	ILE-LYS-ASN-PRO-ILE-LEU	470	1.4
ASP-CYS-HIS-GLY-HIS-VAL	82	0.3	LYS-ILE-LYS-ASN-PRO-ILE	437	1.3
ALA-PHE-VAL-ALA-ARG-ARG	80	0.3	ASP-CYS-HIS-GLY-HIS-VAL	422	1.3
GLY-MET-HIS-CYS-ARG-ASP	80	0.3	SER-THR-PHE-GLU-TRP-PHE	411	1.2
VAL-SER-THR-PHE-GLU-TRP	80	0.3	ARG-GLU-THR-ALA-ILE-GLN	378	1.1
LYS-ASN-PRO-ILE-LEU-THR	79	0.3	VAL-HIS-LEU-THR-GLY-ARG	377	1.1
ASN-PRO-ASP-SER-GLY-GLY	79	0.3	CYS-HIS-GLY-HIS-VAL-SER	365	1.1
PRO-LEU-GLN-LYS-ALA-GLY	79	0.3	PRO-LEU-GLY-ARG-GLU-THR	365	1.1
VAL-ASP-LEU-SER-VAL-SER	78	0.3	GLY-ARG-GLU-THR-ALA-ILE	364	1.1
ARG-GLU-THR-ALA-ILE-GLN	77	0.3	PHE-GLN-THR-LEU-ARG-ILE	364	1.1
ILE-LYS-ASN-PRO-ILE-LEU	72	0.3	VAL-ASN-TYR-TYR-ASN-THR	361	1.1
PHE-PRO-SER-ILE-VAL-GLY	70	0.3	GLY-ILE-ILE-LEU-ALA-GLY	352	1.1
GLU-PRO-HIS-ASN-LEU-HIS	68	0.3	ASN-TYR-TYR-ASN-THR-GLN	352	1.1
SER-GLY-ARG-THR-THR-GLY	66	0.2	VAL-SER-THR-PHE-GLU-TRP	351	1.1
ASN-PRO-LEU-LEU-THR-SER	64	0.2	ILE-ASP-CYS-HIS-GLY-HIS	350	1.1
ALA-GLU-THR-LYS-VAL-SER	64	0.2	THR-PHE-GLU-TRP-PHE-PRO	348	1.1
ILE-THR-SER-ILE-GLU-VAL	64	0.2	LYS-ALA-GLY-HIS-ALA-SER	346	1.0
TYR-ASN-HIS-ALA-ALA-THR	64	0.2	GLU-THR-ALA-ILE-GLN-ARG	342	1.0
ILE-LEU-THR-GLY-PHE-HIS	64	0.2	GLY-ASN-PRO-ASP-SER-GLY	339	1.0
ALA-ASP-PHE-ASP-TYR-PHE	64	0.2	GLN-THR-LEU-ARG-ILE-PRO	337	1.0
VAL-ASN-TYR-TYR-ASN-THR	63	0.2	ALA-ALA-PHE-THR-GLY-ALA	336	1.0
LYS-ILE-LYS-ASN-PRO-ILE	63	0.2	CYS-PRO-LEU-GLY-ARG-GLU	336	1.0

This suggests that the significance of the results depend on the selected interaction cut-off distance θ. This might be because some sequence interactions only occur within or beyond some critical distance.

Nevertheless, for both distances of $\theta = \{6, 10\}$Åtested in this project, the top three interacting sequences had two sequences in common. These were ASN-ASN-TYR-ALA-ASP-PHE and ASN-TYR-ALA-ASP-PHE-ASP. Moreover, within the top four interacting sequences for both $\theta = \{6, 10\}$Å, there were shorter sub-sequences that kept appearing. For example, TYR-ALA-ASP-PHE appeared in the top two sequences for both θ-values.

This seems very promising at first. However, it is worth bearing in mind the fraction of interactions that even these apparently abundant sequences were involved in. The fraction of interactions that each sequence was involved in is expressed as a percentage in the "%" columns of Table 2.

Notice that even the sequences that were involved in most interactions were never involved in more than 0.5% and 1.9% of the total amount of interactions for $\theta = 6$Åand $\theta = 10$Å, respectively.

These are very small fractions which suggest that although some amino acid sequences might appear more frequently in protein-protein interactions, they cannot be used as reliable predictors of sequences that will be involved in most interactions.

5 Conclusion

This project involved creating an algorithm that counted the amount of protein-protein interactions where a sequence of k neighbouring amino acids were involved within a cut-off distance of θÅ.

This algorithm was deployed on a large data set of 870 PDB files to test the hypothesis that some sequences of amino acids are more frequently involved in protein-protein interactions.

This algorithm was capable of being scaled up for use in a distributed memory parallelised computing environment with hyper-threading. This was done in order to deliver results in a shorter amount of time while working on large data sets.

After experimentally determining the parallel computing configuration that led to the optimal speed-up and efficiency, a final analysis was conducted for all combinations of $k = 6$ and $\theta = \{4, 5, 6\}$Å.

Using the optimal configuration of six nodes, each with two threads, on the data set of 870 PDB files, led to a speed-up of three times compered to serial computing. A parallel efficiency of 40% was achieved and the results were computed in a mean time of 13.61 hours with a standard deviation of 3.22 hours.

After analysing the results, it appeared that some sequences of amino acids like ASN-ASN-TYR-ALA-ASP-PHE and ASN-TYR-ALA-ASP-PHE-ASP were indeed involved more frequently in protein-protein interactions. However, even the sequences that were involved in most interactions were never involved in more than 0.5% and 1.9% of the total amount of interactions for $\theta = 6$Åand $\theta = 10$Å, respectively.

Therefore, although some amino acid sequences might appear more frequently in protein-protein interactions, they cannot be used as reliable predictors of sequences that will be involved in most interactions.

References

[1] Worldwide Protein Data Bank. *Protein Data Bank*. (accessed October 16, 2020). URL: `http://www.wwpdb.org/`.

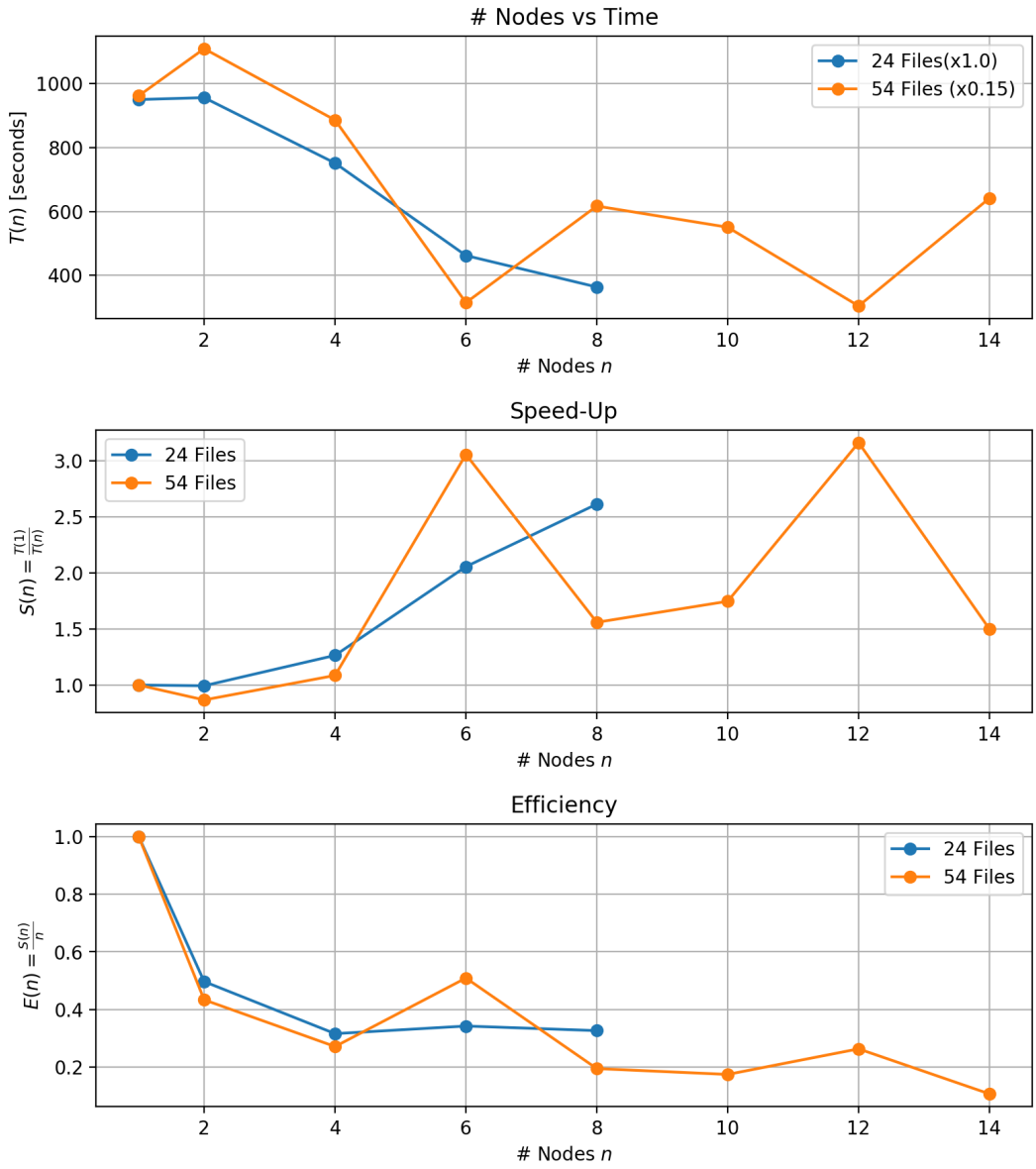

Figure 4: Computation time $T(n)$ for different amounts of nodes n on 24 and 54 BDP files as well as speed-up $S(n)$ and efficiency $E(n)$ (note $k = 2$; $\theta = 6$Å; $T(n)$ was scaled for different data set sizes for ease of comparison)

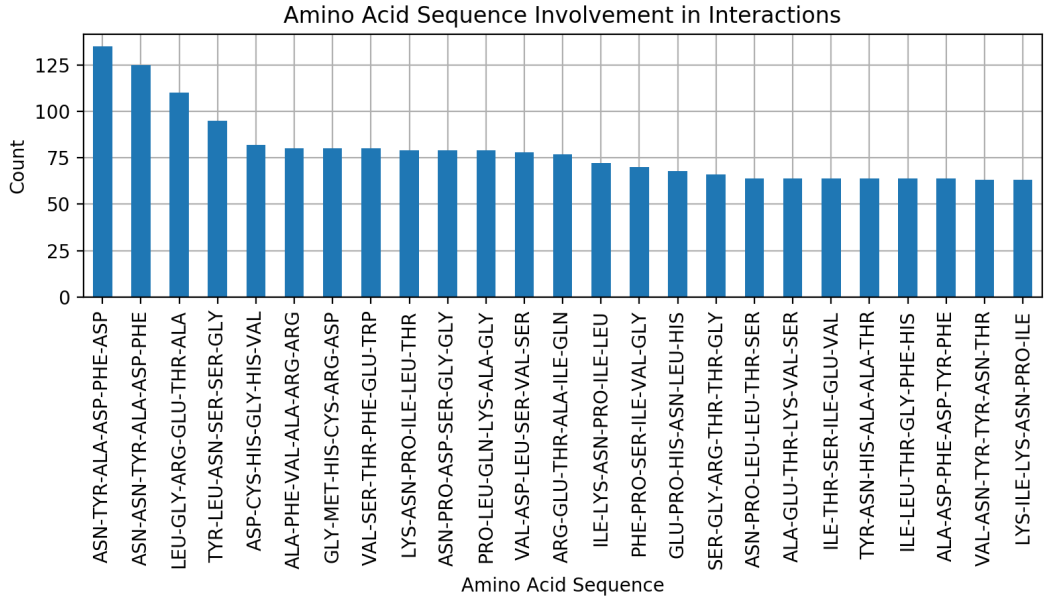

Figure 5: Counts of Interacting Amino Acid Sequences of Final Analysis on 870 PDB Files with $k = 6$, $\theta = 6$Å

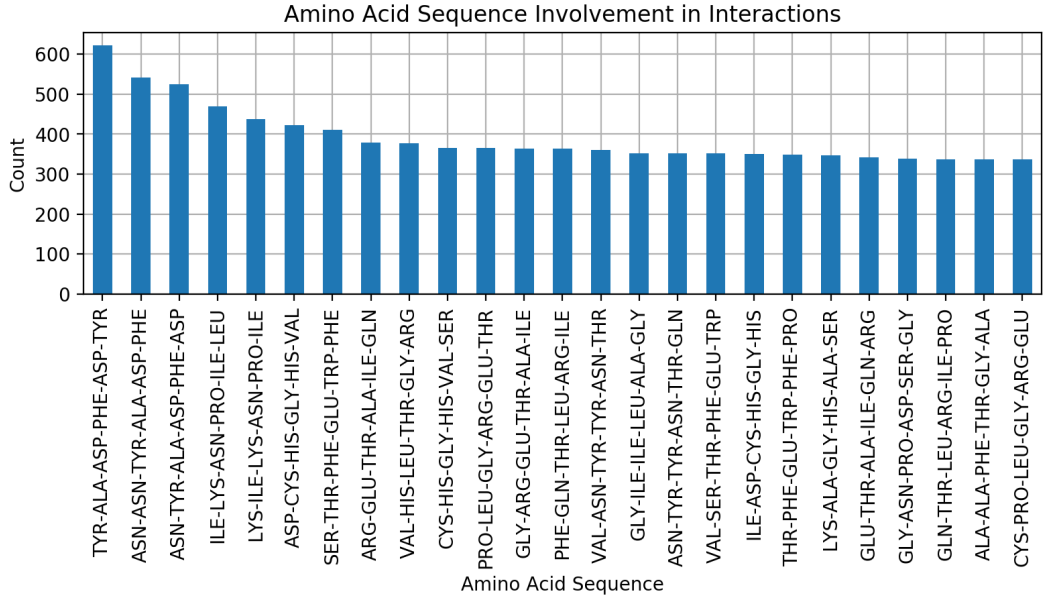

Figure 6: Counts of Interacting Amino Acid Sequences of Final Analysis on 870 PDB Files with $k = 6$, $\theta = 10$Å