A Neural Network Approach to Classification of Protein Residue Interactions

Andrea Auletta
andrea.@studenti.unipd.it

Marco Bernardi
marco.bernardi.11@studenti.unipd.it

Niccolò Zennaro
niccolo.zennaro@studenti.unipd.it

Abstract

Bhooooooooooooo

1. Introduction

Residue Interaction Networks (RINs) are a representation of the non-covalent interactions between amino acid residues within a protein structure, derived based on their geometrical and physico-chemical properties. These networks provide a detailed mapping of intra-protein contacts, which are crucial for understanding the structural and functional dynamics of proteins. RING (`https://ring.biocomputingup.it/`) is a computational tool that facilitates the analysis of these networks by processing Protein Data Bank (PDB) files (`https://www.rcsb.org/`) to identify and classify residue-residue interactions within a given protein structure. RING categorizes these interactions into distinct contact types, including Hydrogen Bonds (HBOND), Van der Waals interactions (VDW), Disulfide Bridges (SBOND), Salt Bridges (IONIC), π-π Stacking (PIPISTACK), π-Cation Interactions (PICATION), Hydrogen-Halogen Interactions (HALOGEN), Metal Ion Coordination (METAL_ION), π-Hydrogen Bonds (PIHBOND), and a category for Unclassified Contacts.

This project is centered around the development of a predictive model that can infer the RING classification of residue contacts using statistical or supervised learning approaches, as opposed to purely geometrical methods. The objective is to design a program that calculates the likelihood or propensity of a residue-residue interaction belonging to each contact type defined by RING, starting from the structural data of the protein.

2. Data Source

The dataset utilized in this study comprises a collection of training examples derived from 3,299 Protein Data Bank (PDB) structures. Each PDB structure is represented by a separate file, which provides detailed information on the residue-residue contacts identified within the corresponding protein. The files are available for download and are organized such that each file contains a tab-separated table of interactions for a single PDB structure.

Across all 3,299 PDB structures, the dataset contains a total of 2,476,056 residue-residue contacts, distributed among various interaction types as follows:

Contact Type	Count
HBOND	901,814
VDW	640,469
PIPISTACK	32,965
IONIC	30,355
SSBOND	1,792
PICATION	7,623
PIHBOND	1,836
Unclassified	860,202

Table 1: Distribution of contact types across the dataset.

Each file is formatted as a tab-separated table, consisting of columns that provide essential details for each contact. The columns include residue identifiers, which follow the same naming conventions as those used in BioPython, along with several pre-calculated features. The final column in each table specifies the type of interaction, categorizing the contact according to the RING-defined classifications.

2.1. Data Preprocessing

The files containing the information for each PDB structure were merged into a single dataframe, creating the dataset for the model. This dataset underwent a preprocessing phase, which involved several crucial steps for data cleaning and preparation.

First, the null values present in the dataset were replaced with the mean of the corresponding column values. This

operation ensured data continuity, minimizing the impact of missing values on the model's performance.

Furthermore, all rows lacking a classification, i.e., without a value in the "interaction" column, were reclassified under the "Unclassified" category. This update involved modifying the values in the "interaction" column, ensuring that every interaction in the dataset was consistently labeled according to the contact categories defined by RING.

3. Approaches

3.1. Balancing the Dataset

How is noticed in the section 2, the dataset has a larger amount of hbond, vdw and unclassified data w.r.t. the others. Have been tried different teqniques to balance the dataset and are all described in the following paragraphs.

Undersampling The number of the samples of the classes with a higher number of samples has been reduced randomly to the number of the class with the lower number of samples or simply to a lower number. The problem of this approach is that we are losing a lot of information and the model could not be able to learn the features of the classes with a higher number of samples.

Oversampling - SMOTE SMOTE is a technique that generates synthetic samples of the minority classes. Basically here we are leading minority classes to have a higher number of examples. It takes the difference between a sample and its nearest neighbors and multiply it by a random number between 0 and 1. The new sample is generated by adding this difference to the original sample. For resource reasons and to avoid creating too many false data so as to confuse the model we decided to increase the value of the smallest classes by about 10 times.

Merge of the classes Also here we've tried different approaches to try to rebalance the number of examples in the dataset:

- Merge all the majority classes in a single class;
- Merge all the minority classes in a single class;
- A combination of the two previous approaches.

The real problem As you will see in the next sections the model is not able to recognize well and to distinguish the samples of the majority classes despite having much more data than the minority classes. This is due to the fact that if we plot the data in a 2D space we can see that the classes are very mixed together. Initially we tried to focus on this problem.