Predicting Possible Oligomerization States of Protein Sequences

Agney K Rajeev & Joel Joseph K B

- **Idea:** We attempt to develop an ML algorithm that predicts possible oligomerization states given a FASTA sequence of a particular protein chain.
- **Targets achieved**:
  - Dataset Curation: A cleaned dataset of 148,820 unique amino acid sequences in FASTA format and their corresponding possible oligomerization states from the RCSB Protein Data Bank is prepared.
  - Data preprocessing and embeddings: Used ProtParam module from Biopython and Pseudo Amino Acid Composition to increase the feature space beyond just sequence similarity. Explored biovec and Glove as possible embedding techniques.
  - ML Implementation: Three kNN models with different distance parameters were implemented.

Previous Works

1. **QuatIdent** by Hong-Bin Shen and Kuo-Chen Chou involves the use of Functional Domain information (FunD) and Pseudo position-specific score matrix (PsePSSM) generated from the sequence to predict the oligomerization state using an OET-kNN algorithm. They constructed a two-level model, with the first level predicting the number of polypeptide chains, i.e., Monomer, Dimer, etc., and the second level predicting the type of polypeptide chains, i.e., Homo or Hetero. The first level was reported to have an accuracy of 71.1%, and the same for the second level was 84% - 96%.

2. **osFP** by Simeon *et al.* is designed to classify fluorescent proteins as Monomers and Oligomers(proteins with more than one sub-unit). They used Amino acid descriptors as sequence features to train a decision-tree algorithm to an excess of 80% accuracy.

Our Approach

We create three kNN models to classify each amino acid sequence to one of ten oligomerization states (Monomer, Homo 2-mer, Homo 3-mer, Homo 4-mer, Hetero 2-mer, Hetero 3-mer, Hetero 4-mer, and Oligomer). Each kNN uses a different distance parameter to find the nearest neighbor for classification.

1. Our first kNN uses the classical idea that proteins with similar amino acid sequences would have a similar structure and, thus similar quaternary structure.
2. Our second kNN uses the vector constructed from the descriptors. The distance parameter is then calculated as the Euclidean distance between these vectors.
3. Our third kNN uses the Pseudo Amino Acid Composition of a protein's sequence. We chose the weight parameter(w) of PseAAC to be 0.05, vector dimension(l) to be 2, and the amino acid functions to be the hydrophilicity value, hydrophobicity value, and the side chain mass. Here, the distance parameter is also calculated as the Euclidean distance between the PseAAC vectors.

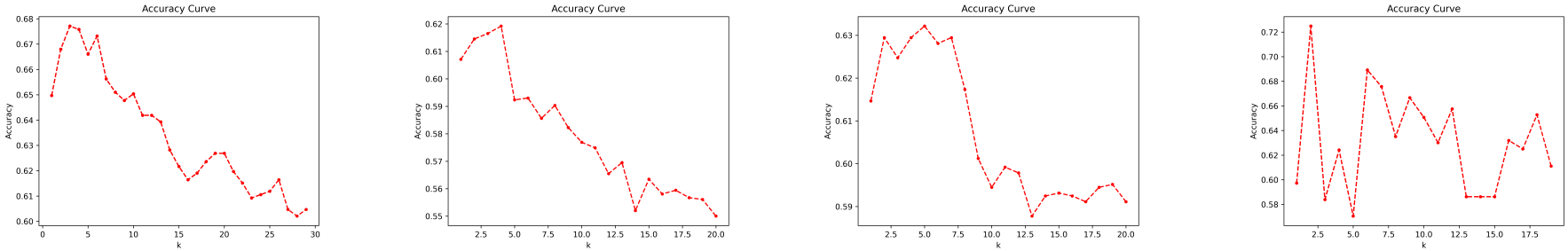

Figure: accuracy vs. k curve for sequence similarity, descriptors, and PseAAC respectively and accuracy vs. l curve for PseAAC

Model	**Accuracy (%)**	**Value of k**	**Distance parameter**
kNN1	70.68	3	Sequence Similarity
kNN2	58.88	4	Descriptors
kNN3	62.06	5	Pseudo AAC

Table: Observed accuracy on 10-fold cross-validation for each model

Future Plans:

- Curation of an Independent test database from Uniprot for more reliable and uniform testing.
- Deep learning model implementation
- Explore better descriptors and embeddings if required

References:

1. Shen, H., & Chou, K. (2009). QuatIdent: a web server for identifying protein quaternary structural attribute by fusing functional domain and sequential evolution information. Journal of Proteome Research, 8(3), 1577–1584. https://doi.org/10.1021/pr800957q
2. Simeon, S., Shoombuatong, W., Anuwongcharoen, N., Preeyanon, L., Prachayasittikul, V., Wikberg, J. E. S., & Nantasenamat, C. (2016). osFP: a web server for predicting the oligomeric states of fluorescent proteins. Journal of Cheminformatics, 8(1). https://doi.org/10.1186/s13321-016-0185-8