Project Report: Annotation Similarity Based on Bio-LLM Encodings

Project Participants:

- **Beshara Theodora**
- **Rossböck Sebastian**
- **Supervisor:** Alexander Pfundner

0. Submission Notes

The git repository for this project can be found at https://github.com/WyruSeppo/BioBert_Annotations. The included zip file contains all the files from the repository, as well as the output file generated by the program. The output file is not present in the repository, as it is too large for github.

The user and developer documentation pdfs can be found in the repository as well. Also, the contents of the user documentation are the same as the README.md of the repository.

1. Introduction

The importance of protein annotation in biological research, medicine, and drug discovery is well established. The rapid advancements in sequencing technologies have led to an exponential increase in biological data. However, effective annotation of protein functions remains a challenge due to data inconsistencies, incomplete information, and semantic ambiguities across different sources. This project leverages BioBERT, a domain-specific large language model (LLM), to generate embeddings for protein annotations and assess their similarity.

The primary goal of this project is to annotate a dataset using multiple sources and calculate annotation similarity using Bio-LLM encodings. The key steps include:

- Fetching data and annotations from functional sources such as UniProt and Pfam.
- Encoding annotation strings using BioBERT.
- Performing clustering and visualization to analyze annotation similarity.

This report details the theoretical background, methodologies used, technical implementation, results, and challenges encountered during the project.

2. Biological and Bioinformatics Background

2.1 Protein Annotation Protein annotations provide essential information about a protein's function, structure, and interactions. Accurate annotations are crucial for drug discovery, genomic research, and understanding molecular mechanisms. However, annotation processes often suffer from:

- **Large and complex datasets**: The volume of sequencing data makes manual annotation infeasible.
- **Incomplete or ambiguous information**: Many proteins lack comprehensive functional descriptions.
- **Inconsistent data formats**: Different databases use varied nomenclatures and structures.

2.2 BioBERT: A Domain-Specific LLM BioBERT is a pre-trained transformer model optimized for biomedical text. It enhances various tasks such as:

- Named entity recognition
- Relation extraction
- Protein annotation

In this project, BioBERT is utilized to generate embeddings for protein annotation descriptions, transforming textual information into numerical representations to facilitate similarity analysis.

3. Methodology

3.1 Data Sources and Preprocessing The project processes protein annotations from two primary sources:

- **UniProt**: A comprehensive protein sequence and function database.
- **Pfam**: A database of protein families and domains.

The input data consists of a FASTA file containing protein sequence identifiers. These identifiers are mapped to UniProt and Pfam entries to fetch relevant annotation descriptions.

3.2 System Configuration Key parameters include:

- **FASTA file path**: Location of protein sequence identifiers.
- **Annotation input and output files**: To store retrieved data.
- **BioBERT model**: dmis-lab/biobert-base-cased-v1.1 for embedding generation.

A configuration file (biobert.ini) manages system settings for automated execution.

3.3 Annotation Retrieval and Processing

- get_uniprot_annotation(protein_id): Fetches UniProt annotations.
- get_pfam_annotation(protein_id): Retrieves Pfam descriptions.
- annotate_data(annotationData): Integrates annotation data from both sources.
- getUniProtConversion(from, to, refseqIds): Converts protein identifiers between formats.

3.4 Embedding Generation with BioBERT

- **Tokenization**: Annotations are tokenized for BioBERT input.
- **Vectorization**: BioBERT converts textual annotations into numerical embeddings.
- **Storage**: Embeddings are saved for further analysis.

3.5 Data Analysis and Visualization To analyze annotation similarity, embeddings are visualized using:

- **t-SNE**: A dimensionality reduction technique for clustering and visualization.
- **Pairwise Distance Metrics**: To evaluate similarity distribution across annotations.
- **Interactive Scatter Plots**: Implemented using Plotly for exploratory data analysis.

4. Results and Discussion

4.1 Annotation Data Statistics

- **UniProt Annotations**:

- Total annotations: 113,003
- Average annotation length: 73.91 words
- Missing annotations: 73.76%

- **Pfam Annotations**:
  - Total annotations: 113,003
  - Average annotation length: 223.91 words
  - Missing annotations: 28.39%

4.2 Embedding and Similarity Analysis

- **Clustering**: t-SNE visualization revealed distinct annotation clusters.
- **Distance Metrics**:
  - Minimum distance: Closest annotation pairs.
  - Maximum distance: Outlier annotations.
  - Average distance: Overall similarity trend.

4.3 Visualization of Embeddings

To better understand the clustering of protein annotations, t-SNE was applied to the generated embeddings, followed by K-Means clustering. The resulting visualizations help illustrate how similar annotations are grouped together based on their vector representations.

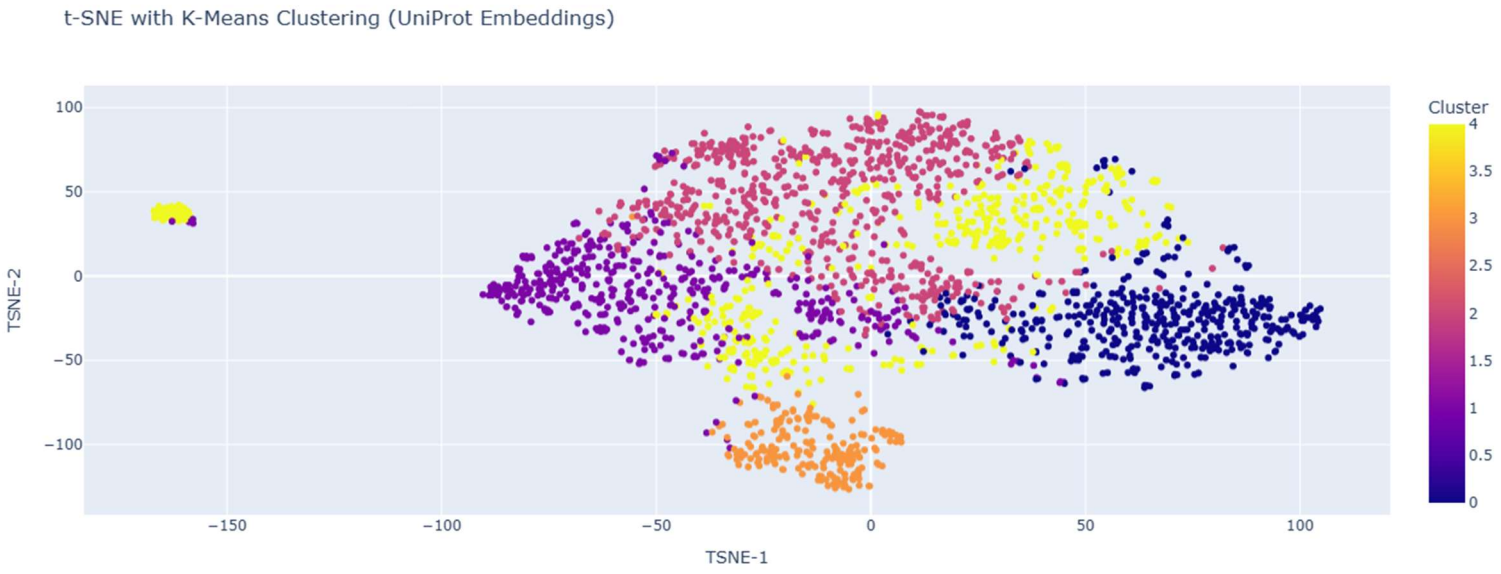

Figure 1: t-SNE with K-Means Clustering on UniProt Embeddings

Figure 1: t-SNE with K-Means Clustering on UniProt Embeddings

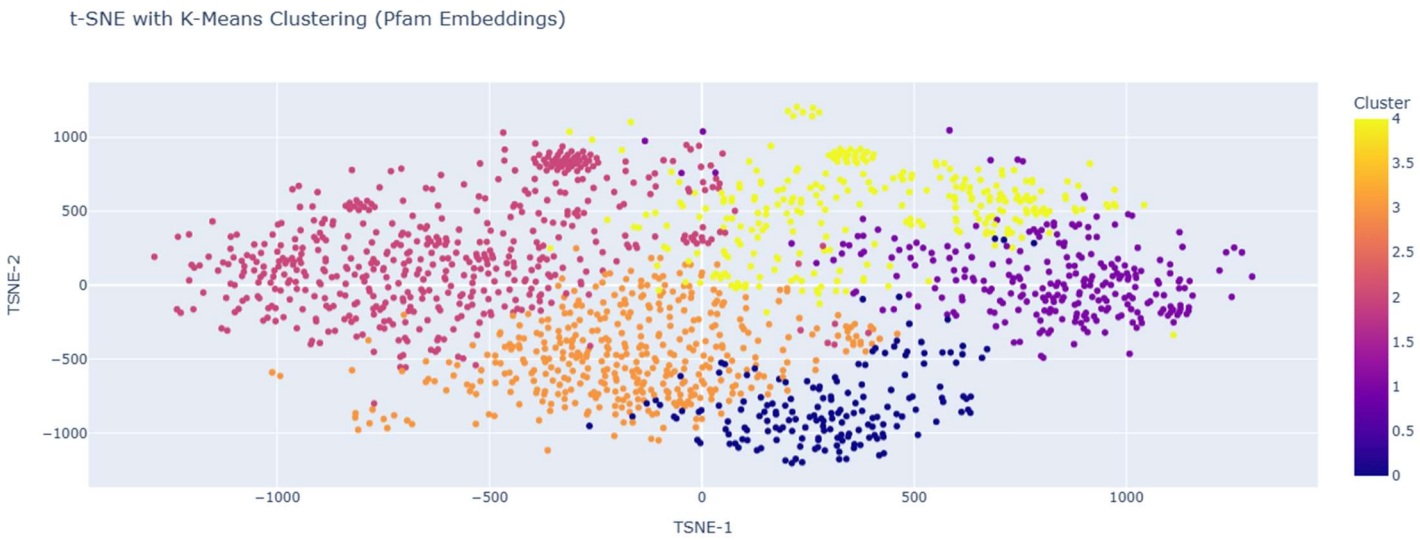

Figure 2: t-SNE with K-Means Clustering on Pfam Embeddings

5. Challenges and Solutions

5.1 Data and API Limitations

- **Issue**: API restrictions and lack of batch processing increased processing time.
- **Solution**: Implemented savepoints to avoid redundant requests and reuse stored data.

5.2 Handling Missing Data

- **Issue**: Many annotations were missing or incomplete.
- **Solution**: Replaced missing values with placeholders to prevent processing errors.

5.3 Computational Constraints

- **Issue**: Large-scale parsing and annotation were computationally expensive.
- **Solution**: Used efficient data structures and batch processing techniques.

6. Lessons Learned

- Implementing an annotation retrieval and embedding pipeline enhanced automation in protein annotation.
- BioBERT proved effective in generating meaningful embeddings for biological text.
- Visualization tools like t-SNE and Plotly facilitated annotation similarity analysis.

7. Applications of BioBERT Embeddings

The application of BioBERT embeddings extends beyond this project:

- **Genomic annotation**: Enhancing functional annotation in bioinformatics.
- **Drug discovery**: Identifying functional similarities among target proteins.

- **Multi-omics analysis**: Integrating transcriptomic, proteomic, and genomic data for holistic insights.

8. References

1. Lee, J., Yoon, W., Kim, S., et al. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4), 1234-1240.
2. UniProt Consortium. (2021). UniProt: The universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1), D480-D489.
3. Finn, R. D., Clements, J., & Eddy, S. R. (2011). Pfam: The protein families database. *Nucleic Acids Research*, 39(Database issue), D211-D222.
4. Mehdinia, M. (2023). Fine-tuned BERT embeddings and t-SNE visualization. Retrieved from https://medium.com/@minamehdinia213/fine-tuned-bert-embeddings-and-t-sne-visualization-bdfd09563744
5. Hugging Face. (2022). Transformers library documentation. Retrieved from https://huggingface.co/docs/transformers