

“AnnoTools”: Extending AnnoTree and AnnoView for Database-Wide Genome Annotation, Visualization, and Comparison

by

Huagang Tan

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Biology

Waterloo, Ontario, Canada, 2025

© Huagang Tan 2025

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Genomic analysis has revolutionized our understanding of the biology and evolutionary history of bacterial and archaeal microorganisms, leading to numerous applications in biotechnology, medicine, and environmental sciences. One of the fundamental aspects of genomic analysis is protein functional annotation, which involves assigning biological functions to protein-coding sequences identified within genomes. These annotations are widely used to support analyses, such as examining gene or function distributions across the tree of life and comparing gene neighborhoods across taxa. By combining these analyses, researchers can comprehensively explore gene functions and the mechanisms of given genes or gene clusters. In this thesis, I will introduce a pipeline that supports genomic analysis. The project consists of three parts: data annotation, visualization, and the language model.

The first part of the pipeline is the generation of protein function annotations. Raw protein sequence data is downloaded from the Genome Taxonomy Database (GTDB) and submitted to two tools: Kofamscan and DIAMOND. Kofamscan assigns KEGG ORTHOLOGY IDs to each input sequence, while DIAMOND assigns Uniref IDs, which are then mapped to InterPro IDs. Combining these IDs provides comprehensive and reliable annotations. The data is filtered for quality and stored on a remote server as an annotation database for further analysis.

The second part of the pipeline involves updating two user-friendly, web-based visualization tools, AnnoTree and AnnoView, which utilize the annotation database. AnnoTree displays the distribution and taxonomy of different protein annotations across GTDB using a tree of life representation, offering insights into biological and evolutionary patterns through species phylogenies and supporting genome-wide co-occurrence analysis. AnnoView focuses on comparing and exploring gene neighborhoods, identifying functionally related genes clustered together in genomes as "gene clusters," thus emphasizing window-based co-occurrence analysis. The new annotation database not only provides more comprehensive and accurate annotations, enhancing the databases that both visualization tools rely on, but also extends their functionalities for fast data retrieval and new features.

The last part of the pipeline involves the application of the Word2Vec language model, which treats genome contigs as sentences in natural language and trains the model using the annotation database. After training, the updated model can encode each annotation from a specific protein family into high-dimensional vectors with continuous number, allowing researchers to explore annotations that share similar genomic contexts. This allows protein functions prediction based on this comparative gene neighborhood analysis.

Finally, I will use one protein domain in the Type VI Secretion System (T6SS) as a case study. T6SS is a cell envelope-spanning machine that translocates toxic effector proteins into

eukaryotic and prokaryotic cells. Besides the conserved essential core components, there are various effector and accessory proteins in the system. Some proteins are annotated as Domains of Unknown Function (DUF) and are poorly explored. In this case, I will focus on PF20598 (DUF6795), which shares a similar genomic context with one of the T6SS proteins. Using the visualization tools AnnoTree and AnnoView, I will demonstrate that this DUF is part of the T6SS cluster, supporting the hypothesis that it may function as an adaptor protein in T6SS.

In summary, the AnnoTools pipeline integrates all components to enhance comparative genomic analysis with a large-scale annotation database. The user-friendly web-based tools enable researchers to visualize data both genome-wide and at a window-based scale. The ultimate goal of this thesis is to provide researchers with a comprehensive and easy-to-use method for predicting functions of genes or gene clusters of interest.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor, Dr. Andrew C. Doxey, for his passion, dedication, and invaluable guidance throughout my research journey.

I am deeply thankful to my committee members, Dr. Laura A. Hug and Dr. Gabriel Moreno-Hagelsieb for their insightful feedback and constructive suggestions.

Finally, I extend my sincere appreciation to my collaborators—Zijing Wu, Xin Wei, Briallen Lobb, and Jake Colautti—for their contributions and support, which have greatly enriched and improved my thesis.

Dedication

This is dedicated to my loving family and friends.

Table of Contents

Author's Declaration	ii
Abstract	iii
Acknowledgements	v
Dedication	vi
List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 Overview of Protein Sequence Databases	2
1.2 Overview of Protein Functional Annotation Methods	4
1.3 Overview of Tools for Both Genome-Wide and Window-Based Visualization	5
1.4 Language Model for Vectorizing Protein Functional Annotation Based on Ge- nomic Context	8
1.5 Methods for Exploring Uncharacterized Proteins Using Visualization Tools and the Language Model	10
1.6 Research Aims	11

2	The AnnoTools Pipeline	13
2.1	Background	13
2.2	The Annotation Pipeline	13
2.2.1	Coverage Evaluation	18
2.2.2	Future Work	22
2.3	Advancements in AnnoTree and AnnoView: Development of Version 2.0	23
2.3.1	AnnoTree Web Application	23
2.3.2	AnnoView Web Application	30
2.4	Language Model for Protein Function Prediction	34
2.4.1	Word2Vec Model	34
2.4.2	Application	39
2.4.3	Evaluation	43
2.4.4	Future Work	47
3	Case Analysis Using AnnoTools Pipeline	49
3.1	Background	49
3.2	Language Model Prediction and Analysis	50
3.2.1	AnnoTree-Based Analysis	51
3.2.2	AnnoView-Based Analysis	54
3.3	Follow-Up Analysis Using Foldseek	56
4	Conclusion and Future Directions	59
4.1	Contribution	59
4.2	AnnoTools Applications and Language Model	60
4.3	Limitations	60
4.4	Future Directions	61
	References	62

List of Figures

1.1	Sequence counts in GenBank, WGS, and RefSeq databases.	3
1.2	Comparison of species counts for Archaea and Bacteria in RefSeq and GTDB. . .	4
1.3	Description of CBOW and Skip-gram Architectures	8
1.4	Overview of AnnoView visualizations for different protein annotations	10
2.1	AnnoTool pipeline	14
2.2	Workflow for the annotation pipeline	17
2.3	The MySQL database schema	19
2.4	Distribution of annotation coverage for bacteria across various databases	20
2.5	Distribution of annotation coverage for archaea across various databases	21
2.6	AnnoTree visualization for version r95	24
2.7	AnnoTree visualization for version r214	25
2.8	Top 10 occurrences of annotation IDs from dataset	27
2.9	AnnoTree 1.0 Application	28
2.10	AnnoTree 2.0 Application	29
2.11	combination search diagram, case 1	32
2.12	combination search diagram, case 2	33
2.13	Semantic Analysis in AnnoView	38
2.14	PCA visualization for word embedding	39
2.15	Diagram for the T6SS structure	40
2.16	Metrics for Word2Vec model	46

3.1	AnnoTree visualization of domain ID PF20598	52
3.2	AnnoTree visualization with order level	54
3.3	Visualization of case studies using AnnoView	55
3.4	Diagram for The TransThyretin-Related (TTR) domain	57
3.5	AnnoView visualization for the gene neighborhood for PF19933	58

List of Tables

1.1	Comparison between AnnoTree 1.0 and AnnoTree 2.0	6
1.2	Comparison between AnnoView 1.0 and AnnoView 2.0	7
1.3	Overview of Member Databases Integrated by InterPro	7
2.2	Accuracy in predicting different KEGG pathways in prokaryotes	47
3.1	Top-scoring DUFs with semantic similarity to the T6SS protein VasX (PF20249)	51
3.2	Hypergeometric Test for Co-Occurrence Significance across the Bacterial Database	53
3.3	Foldseek Results for Three DUFs	56

Chapter 1

Introduction

Genome sequencing enables researchers to identify genes and predict their functions, providing crucial insights into the biological processes and metabolic pathways in prokaryotes. This knowledge is fundamental for understanding the biological significance of proteins and how these organisms live, reproduce, and interact with their environments.[1].

Recently, with the development of computational methods and the availability of public genomic databases, an increasing number of researchers now rely on protein function annotation as a foundation for major downstream bioinformatics analyses following genome sequencing. Databases of corresponding protein sequences, functional domains, and motifs are a critical basis for this type of annotation [2]. These analyses include, but are not limited to, examining gene or function distributions across the tree of life, comparing gene neighborhoods across different taxa, and predicting the functions of uncharacterized proteins. Such analyses play a significant role in advancing our understanding of molecular biology and the evolutionary relationships among organisms.

Therefore, a comprehensive pipeline that provides both accurate protein function annotations and integrated data analysis tools is essential. The annotated database can then serve as the basis for data analysis tools for visualization and function prediction, enabling researchers to more easily interpret complex genomic data, generate new hypotheses, and make further discoveries in the field of biology.

1.1 Overview of Protein Sequence Databases

Existing annotation pipelines commonly focus on user-submitted data [3, 4], where the analysis cannot directly link to genomes from existing public databases for additional information. If researchers are interested in genomic-context-based analysis across public databases, current tools are limited. These analyses include evaluating gene presence in conserved and putative operons, gene co-occurrence (presence and absence) across genomes, and detecting functionally equivalent genes by analyzing anticorrelating gene occurrences [5], all of which require database-wide annotations.

This highlights the need for a large and phylogenetically consistent protein sequence database. The large size of such a database allows researchers to conduct comparative analyses among extensive data sets, and phylogenetic consistency ensures that the data is objective and supports taxonomy-based analysis.

The Whole Genome Shotgun (WGS) method involves sequencing random fragments of a genome and then assembling these fragments into a complete sequence [6]. In this context, the WGS database hosted by NCBI serves as a comprehensive repository for whole genome sequences. It contains an extensive collection of over 3 billion sequences from a diverse array of organisms (Fig 1.1). This database plays a crucial role in genomic research by providing raw sequencing data, including unassembled and draft sequences, as well as assembled genomes. However, the sequences in the WGS database can be redundant, or of variable quality. Additionally, the annotations within the database largely depend on the submitters' methods and expertise, which can lead to potential inconsistencies in data quality and interpretation. Despite these challenges, the NCBI WGS database remains an invaluable resource for researchers across various fields.

Another database, RefSeq [7], is a curated collection of reference sequences for a variety of organisms, providing comprehensive, non-redundant sequences that serve as a standard reference for genomic research and applications (Fig 1.1). It relies on NCBI taxonomy as part of its annotation. While NCBI taxonomy is a long-standing system that curates classifications based on a combination of phenotypic and genotypic data [8], recent research has highlighted inconsistencies arising from historical practices and the subjective nature of phenotypic traits. Additionally, the NCBI taxonomy is not strictly rank-normalized, meaning equivalent evolutionary distances can be assigned to different taxonomic ranks [8].

The Genome Taxonomy Database (GTDB) [9] is a newer system that focuses on phylogenetically consistent and rank-normalized classification based on whole-genome sequences. The genomes used to construct the phylogeny are sourced from RefSeq and GenBank, with GTDB releases indexed to RefSeq releases. Both databases contain a large number of species from

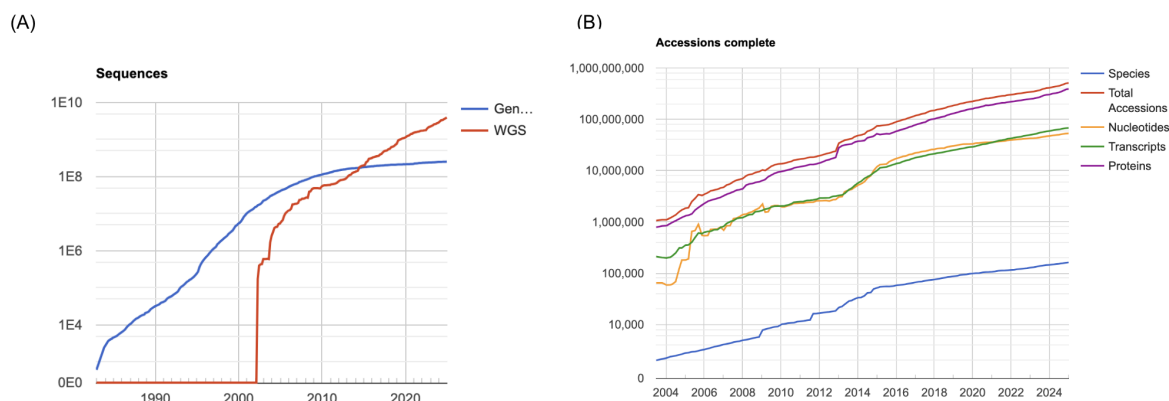


Figure 1.1: (A) The number of sequences in GenBank and the Whole Genome Shotgun (WGS) database (3,745,772,748 records from WGS as of October 15, 2024). (B) The number of sequences in the RefSeq database (377,783,847 proteins as of November 4, 2024).

both bacteria and archaea, and the number is continuously increasing to maintain current relevance (Fig 1.2)^{1 2}. GTDB employs Average Nucleotide Identity (ANI) to define species boundaries and Relative Evolutionary Divergence (RED) to establish higher taxonomic ranks. This methodology aims to create a more objective and evolutionarily informative classification system. Additionally, GTDB includes draft genomes from uncultured microorganisms obtained from metagenomes and single cells, thereby enhancing the representation of the microbial world compared to RefSeq, which primarily focuses on genomes from cultured microorganisms. GTDB requires that all genomes undergo independent quality control using CheckM [10] before inclusion, which guarantees that the database predominantly contains medium to high-quality genome data. Furthermore, GTDB provides amino acid FASTA files for all predicted protein-coding gene sequences using Prodigal [11], establishing a solid foundation for functional annotation by pre-identifying genes.

Therefore, utilizing data from GTDB can effectively address the previously mentioned challenges by providing large-scale metadata that already includes phylogenetically consistent taxonomy information.

¹NCBI GenBank Statistics: <https://www.ncbi.nlm.nih.gov/genbank/statistics/>

²NCBI RefSeq Statistics: <https://www.ncbi.nlm.nih.gov/refseq/statistics/>

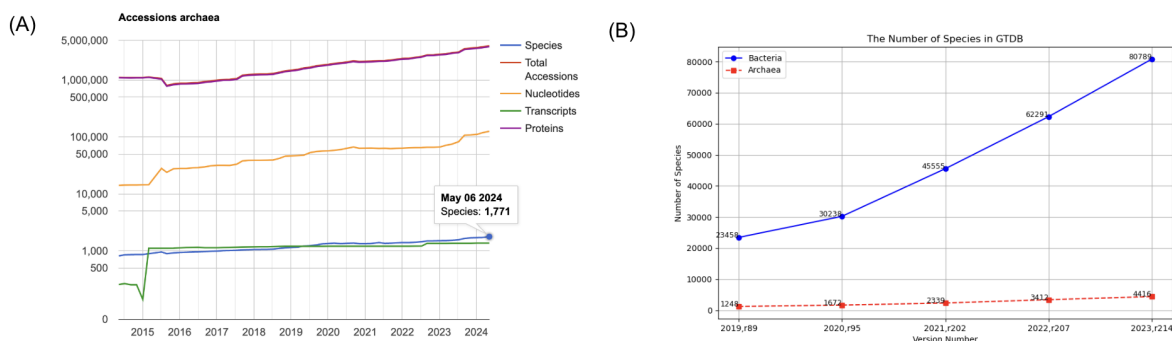


Figure 1.2: (A) The number of species for Archaea in RefSeq. The figure 1.1 shows the number of species for Bacteria in RefSeq. (B) The number of species for Archaea and Bacteria in GTDB.

1.2 Overview of Protein Functional Annotation Methods

The methods used to assign functional annotations to protein-coding genes are varied and can influence the usage of annotated data. One of the most widely used methods is homology-based functional annotation. Sequence similarity searches can identify homologous proteins or genes by detecting excess similarity—statistically significant similarity that reflects common ancestry [12]. As homologs, they are generally considered functionally similar [13], so clusters of these homologous sequences can be assigned the same functional annotation. Common approaches for this type of functional annotation include sequence alignments for homology and domain family profiles.

The first method generally utilizes sequence search tools like BLAST [14] and DIAMOND [15] to compare the sequence similarity between the sequence of interest and the reference protein database. The annotation in the reference protein database can then be transferred to the target sequence. For example, if a sequence is mapped to the UniRef100 [16] database which clusters similar protein sequences sharing at least 100 percent sequence identity from UniProtKB into groups to reduce redundancy, then, given a sufficiently low e-value, the sequence can use the functional annotation pre-computed for the UniRef100 representative sequence since they are considered homologous sequences and functionally similar. The e-value in alignment tools represents the number of hits one can expect to see by chance when searching a database; lower e-values indicate higher significance, suggesting that the match is less likely to have occurred by random chance.

The second method relies more on statistical models, such as Hidden Markov Models (HMMs)[17], which highlight regions of conserved residues as signatures based on the results of multiple se-

quence alignments (MSA) from protein families. This method focuses on the domain of the sequence instead of full-length comparison, making it more sensitive to finding remote homologous sequences where the functional domain is conserved while other parts may be divergent. Tools like KofamScan [18] for Kyoto Encyclopedia of Genes and Genomes Orthology (KO [19]) and InterProScan [20] for InterPro ID [21] are widely used for protein domain annotation, relying on HMMs from HMMER. Each protein family domain has adaptive, family-specific thresholds to provide more accurate annotations for different protein families.

In this thesis, both methods are utilized to provide accurate and comprehensive functional annotations. KofamScan assigns KEGG Orthology (KO) IDs and links GTDB genomes to KEGG pathways using family-specific thresholds. Additionally, protein-coding genes from GTDB are mapped to the UniRef100 database, which provides rich annotations such as domains, structures, interactions, and gene ontology (GO) [22]. These annotations, along with InterPro IDs linked through UniRef100, enhance the functional and structural detail of GTDB data. It should be noted that InterPro combines 13 protein signature databases into one central resource: CATH-Gene3D [23], the Conserved Domains Database (CDD) [24], HAMAP [25], PANTHER [26], Pfam [27], PIRSF [28], PRINTS [29], PROSITE Patterns [30], PROSITE Profiles [30], SMART [31], the Structure-Function Linkage Database (SFLD) [32], SUPERFAMILY [33] and TIGR-FAMs [34]. The combination of these member databases, InterPro itself, and the KO annotation from KofamScan results in more comprehensive and higher-coverage functional annotations.

1.3 Overview of Tools for Both Genome-Wide and Window-Based Visualization

Once the genes are annotated and the annotation database is built, a commonly used application is comparative analysis. Such analyses include the exploration of co-occurrence-based or anticorrelation-based phylogenetic patterns and conserved gene neighborhoods. To better utilize the annotation database, two visualization tools previously developed by the Doxey lab, AnnoTree [35] and AnnoView [36], have been updated to better adapt to the new annotation database.

AnnoTree (version 1.0) focuses on phylogenetic pattern exploration and enables visualization of millions of precomputed genome annotations across bacterial and archaeal phylogenies, allowing users to explore gene distributions as well as patterns of gene gain and loss in prokaryotes [35]. It parses tree structure files in NEWICK format from GTDB and creates a user-friendly interactive phylogenetic tree. Users can explore phylogenetic relationships among species and search the function annotation database through the AnnoTree interface to display the distri-

Table 1.1: Comparison between AnnoTree 1.0 and AnnoTree 2.0

Feature	AnnoTree 1.0	AnnoTree 2.0
GTDB version	r95 (31,910 genomes)	r214 (85,205 genomes)
Annotation types	KEGG, Pfam, TIGRFAM	KEGG and InterPro database*
Annotation search support	Single database search	Cross-database search**
Search speed	10 seconds for 7,613 hits	1 second for 15,437 hits***
Tree visualization	Static visualization	Interactive visualization

* The InterPro database contains 13 member databases including CATH-Gene3D, the Conserved Domains Database (CDD), HAMAP, PANTHER, Pfam, PIRSF, PRINTS, PROSITE Patterns, PROSITE Profiles, SMART, the Structure-Function Linkage Database (SFLD), SUPERFAMILY and TIGRFAMs.

** Cross-database search means the search can occur between different databases, for example, K02453 from KEGG and PF19526 from Pfam.

*** The result is tested for the annotation PF04717 from Pfam database.

bution of function annotations across the prokaryotic Tree of Life, showing the genome-wide co-occurrence of annotations of interest.

AnnoView (version 1.0) focuses on visualizing gene neighborhoods surrounding a center gene within a defined window, which is a continuous region of a certain range within the same genome. The center gene is the gene of interest and serves as the anchor for the window. Genes within genomes are drawn based on the coordinates from the FASTA file provided by GTDB and are colored by their annotation. Thus, conserved gene neighborhood compositions are colored the same, providing intuitive visualization. Additionally, AnnoView can perform horizontal alignment for center genes and sort genomes based on the neighborhood composition of the center gene. It also allows users to search for sequences against the GTDB, treat homologous sequences as center genes, and visualize them and their neighborhoods.

Both tools and their corresponding annotations are already linked to GTDB (release r95). However, with the increase in data in GTDB and changes in annotation tools, the databases for AnnoTree and AnnoView have become outdated. This necessitates updates for both tools to use GTDB and the new annotation database. Therefore, this thesis updates both tools, AnnoTree and AnnoView, to version 2.0, increasing the scale of the database and extending functionality to better accommodate the new annotations (see tables 1.1 and 1.2) for updates and comparison between versions 1.0 and 2.0 for AnnoTree and AnnoView).

Table 1.2: Comparison between AnnoView 1.0 and AnnoView 2.0

Feature	AnnoView 1.0	AnnoView 2.0
GTDB version	r95 (31,910 genomes)	r214 (85,205 genomes)
Annotation types	KEGG, Pfam, TIGRFAM	KEGG and InterPro database*
Homologs sequence search	✓	✓
Combination of annotations search	-	✓
Window size for neighbourhood	10kb	5 to 30kb

* The InterPro database contains 13 member databases.

The most important update for both tools is the change in annotation types. Version 1.0 of the tools utilized only three types of annotations: KEGG, Pfam, and TIGRFAM (NCBIfam). The updated annotation types include KEGG and InterPro, with InterPro subdivided into 13 member databases (Table 1.3), increasing the supported annotation types from 3 to 14. Another notable update for AnnoView specifically is the new “combination search” feature. This feature can search for the combination of annotations of interest within a certain window in a short time. Typical questions like “What is the taxonomy distribution containing proteins X and Y in a 10kb region?” or “Are there any gene clusters containing proteins X and Y in a 10kb region?” can now be answered using AnnoView.

Table 1.3: Release version and number of member database signatures integrated into InterPro version 90.0

Member database	Release number	Total signatures	Integrated signatures
CATH-Gene3D	4.3.0	6631	2712 (40.9%)
CDD	3.18	16212	3817 (23.5%)
HAMAP	2021_04	2383	2379 (99.8%)
PANTHER	15.0	139691	10584 (7.6%)
Pfam	35.0	19632	19070 (97.1%)
PIRSF	3.10	3285	3236 (98.5%)
PRINTS	42	2106	1944 (92.3%)
PROSITE patterns	2022_01	1311	1283 (97.9%)
PROSITE profiles	2022_01	1326	1258 (94.2%)
SFLD	4	303	158 (52.1%)
SMART	7.1	1312	1267 (96.6%)
SUPERFAMILY	1.75	2019	1642 (81.3%)
TIGRFAMs	15	4488	4434 (98.8%)

1.4 Language Model for Vectorizing Protein Functional Annotation Based on Genomic Context

Besides the visualization, the annotation database can also support machine learning-based comparative analysis for protein function prediction. One of the techniques employed in this process is Word2Vec [37], a method in natural language processing (NLP) used to create word embeddings. These embeddings are dense vector representations of words, allowing the model to represent words in a continuous vector space where words sharing similar contexts are located close to each other. Word2Vec consists of two main models: Continuous Bag of Words (CBOW) and Skip-Gram. The CBOW model predicts a target word based on its surrounding context, while the Skip-Gram model predicts context words based on a target word (Fig 1.3). Although the training processes for both models are similar, their performance differs. Skip-Gram is slower to train but achieves better overall accuracy for both semantic and syntactic tasks.

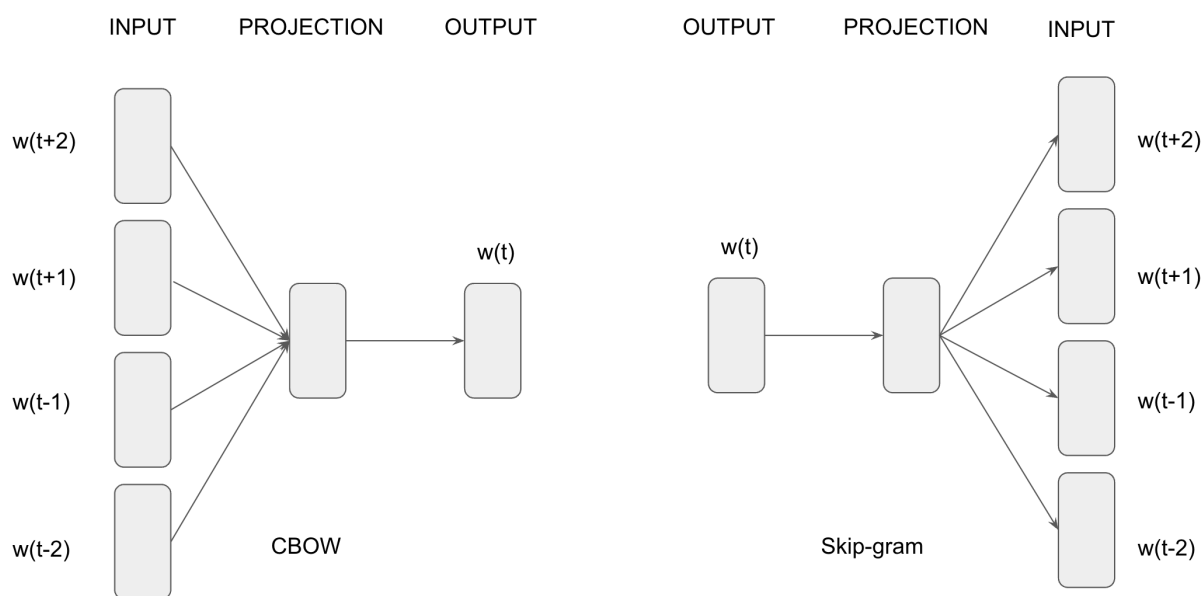


Figure 1.3: Left: The CBOW architecture predicts the current word based on the context. Right: The Skip-gram architecture predicts surrounding words given the current word. w is the window size and t is the position of the current word. The projection layer is the actual word embedding in the middle layer. The figure is adapted from the word2vec paper. [37]

In previous works [38, 39], public genomic databases have been treated as a 'corpus', with contigs of genomes serving as 'sentences' and single annotations of protein domains as 'words'. Annotations within sliding windows are used to generate input-label word pairs for training. The training process for Word2Vec with the protein annotation database follows these steps: First, annotations are encoded as one-hot vectors. Second, a map function is created so that the one-hot vectors can be linearly mapped to the dimension of a user-selected hyperparameter, with the weight matrix (embeddings) initialized randomly. Third, during the forward pass, the input vector is multiplied by the weight matrix to calculate the hidden layer representation, which is also the word embedding. In the CBOW model, context word vectors are averaged to predict the target word, while in the Skip-Gram model, the target word vector is used to predict each context word. Fourth, the difference between the predicted and actual words is measured using a loss function, typically cross-entropy loss. Fifth, this error is backpropagated through the network to update the weight matrix. These steps are repeated for multiple epochs until the model converges. The resulting weight matrix contains the word embeddings, where each row corresponds to the vector representation of a word, capturing semantic relationships such that words with similar meanings have similar vector representations.

In this thesis, annotations are extracted from the Pfam database for the Word2Vec model and are subdivided into three categories: protein domain IDs with known functions like PF04717 for the Type VI secretion system/phage-baseplate injector OB domain, domains of unknown function (DUF) [40] like PF13503 for Domain of Unknown Function index 4123, and customized placeholders consisting of 'GAP' as a prefix and an approximate length as a suffix for proteins without any annotation (e.g., GAP500). A domain of unknown function (DUF) is a protein domain that has no characterized function. These families have been collected together in the Pfam database using the prefix DUF followed by a number, with examples being DUF4123. The DUF designation is tentative, and such families tend to be renamed to a more specific name (or merged into an existing domain) after a function is identified [41]. The last category, 'GAP', helps to separate genes while maintaining the genomic context structure within each contig of the genome (Fig 1.4).

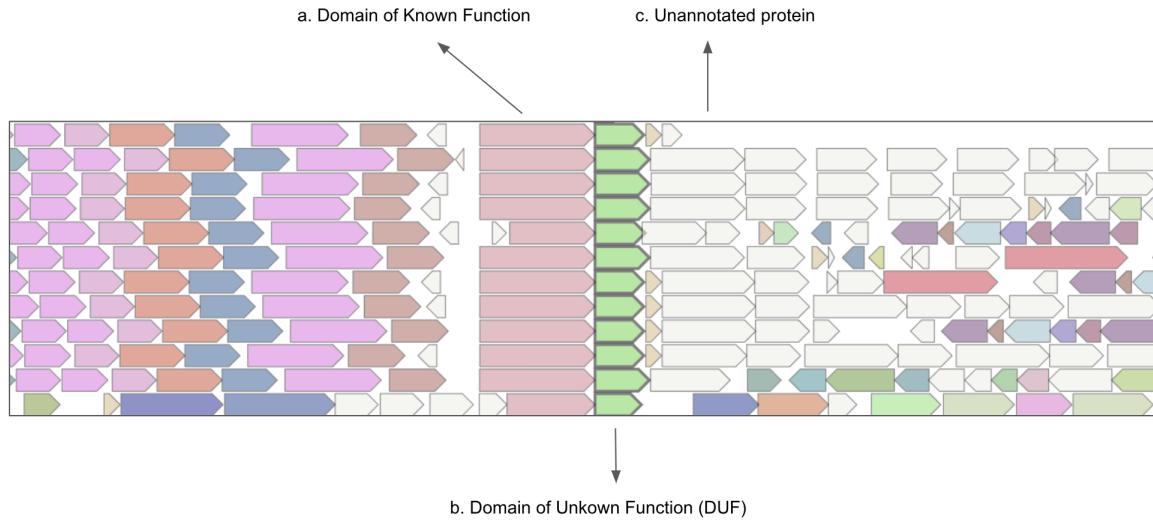


Figure 1.4: AnnoView visualization for 3 types of annotations used in the Word2Vec model:
(a) Domains with well-annotated definitions and short names, PF04717 (Phage base V).
(b) Domains with unknown functions, PF13503 (DUF4123).
(c) Domains without any annotations, GAP1500, where 1500 is the length of the protein.

Once the training process is completed, the model can be used to predict the function of uncharacterized proteins like DUFs by investigating proteins with similar contexts. These proteins with similar contexts may work within the same pathway or system, allowing researchers to infer potential functions based on their genomic neighborhood and associations. This approach leverages the contextual information captured by the Word2Vec model to provide insights into the roles of previously unknown proteins, enhancing our understanding of their biological significance and interactions.

1.5 Methods for Exploring Uncharacterized Proteins Using Visualization Tools and the Language Model

Uncharacterized proteins are proteins whose functions have not been identified or confirmed. These proteins are typically annotated as “hypothetical”, “predicted”, or “uncharacterized” in protein databases. Tools like BLAST and PfamScan are highly effective in identifying sequences that are similar to known proteins. However, if the database lacks functional annotations for similar sequences, BLAST cannot provide functional predictions.

Protein function can be context-dependent, influenced by interactions with other proteins, which shows the effectiveness of gene neighborhood analysis. Analyzing evolutionary relationships can also reveal functional similarities that are not apparent from sequence similarity alone. Therefore, by combining these approaches, researchers can gain a more comprehensive understanding of the roles and mechanisms of uncharacterized proteins, beyond sequence similarity-based methods.

In order to further explore these relationships, AnnoTree and AnnoView can be used to provide more comprehensive correlation analyses. AnnoTree focuses on the phylogenetic distribution of genomes that contain both the output protein and the query protein, while AnnoView examines the window-based co-occurrence of the output protein and the query protein. This means that both tools can not only be used to perform a global inspection of co-occurrence correlations, but also localized, window-based correlation tests using the hypergeometric test. The hypergeometric test calculates the probability of observing a given number of successes (co-occurrences) in a sample drawn from a finite population without replacement. In the context of protein annotation, it helps determine whether the observed overlap between two annotations in a dataset is statistically significant. The result of the cumulative probability calculation gives a p-value. This p-value can then be compared across different combinations of searches to assess whether the protein domains are co-occurring or anti-co-occurring with each other. Typically, if two protein domains are correlated, the p-value will be very small. If the p-value is large, it suggests that the two domains share a similar genomic context, but not all protein domains consistently co-occur with one another, indicating that further investigation is needed in such cases.

Additionally, both AnnoTree and AnnoView can be used to verify the relationship between the query word and its predicted related gene clusters. In the example of T6SS, VgrG is the spike protein and plays a necessary role in the system. By searching for the combination of the target word and the VgrG protein, AnnoTree and AnnoView can provide phylogenetically and gene neighborhood-based visualizations, respectively. Besides the tools discussed in this thesis, the DUF can also be further explored structurally using other tools, including AlphaFold [42] and Foldseek [43]. These tools can provide additional evidence beyond genomic context-based function prediction, offering deeper insights into the structural and functional aspects of DUFs.

1.6 Research Aims

As discussed above, there is a need for a large-scale protein functional annotation database, and it is necessary for the annotation pipeline to keep the data up-to-date to follow the rapid expansion of public genomic sequence databases. Updates to both AnnoTree and AnnoView can ensure that the visualizations are not outdated. These updates include better response speed,

a more comprehensive annotation database, and new features that can support researchers in exploring phylogenetic and gene neighborhood-based comparative analyses. Additionally, the language model Word2Vec, developed for uncharacterized protein function prediction, can be connected with visualization tools to provide insights into protein function exploration. The following chapter, Chapter 2, describes the development of the annotation database, the updates for visualization tools, and the training process and experiments for the language model. In Chapter 3, the methods used for the case analysis will be described. Finally, in the last chapter, the future directions and potential improvements of the AnnoTools pipeline will be discussed.

Chapter 2

The AnnoTools Pipeline

2.1 Background

The AnnoTools suite can be separated into three parts to support genomic analysis: the annotation pipeline, visualization tools, and the language model. The pipeline begins with data from public genomic databases, which are then annotated using the annotation pipeline. This pipeline builds a database on a remote server for fast data retrieval, forming the basis of the overall pipeline and supporting both the visualization tools and the language model. Tools like AnnoTree and AnnoView can directly pull data from this database for visualization purposes, while the language model is trained on the annotation database (Fig 2.1). In this chapter, I will present the detailed architecture and implementations of these three components.

2.2 The Annotation Pipeline

The Genome Taxonomy Database (GTDB) is an initiative to establish a standardized microbial taxonomy based on genome phylogeny. The genomes used to construct this phylogeny are obtained from RefSeq and GenBank, and GTDB releases are indexed to RefSeq releases. Additionally, GTDB provides amino acid FASTA files for all protein-coding gene sequences in archaeal and bacterial microorganisms, predicted with Prodigal. The version of GTDB used in this thesis is r214, which consists of 80,789 bacterial genomes and 4,416 archaeal genomes. The total size of all raw files exceeds 100 GB.¹

¹GTDB website: <https://gtdb.ecogenomic.org/stats/r214>

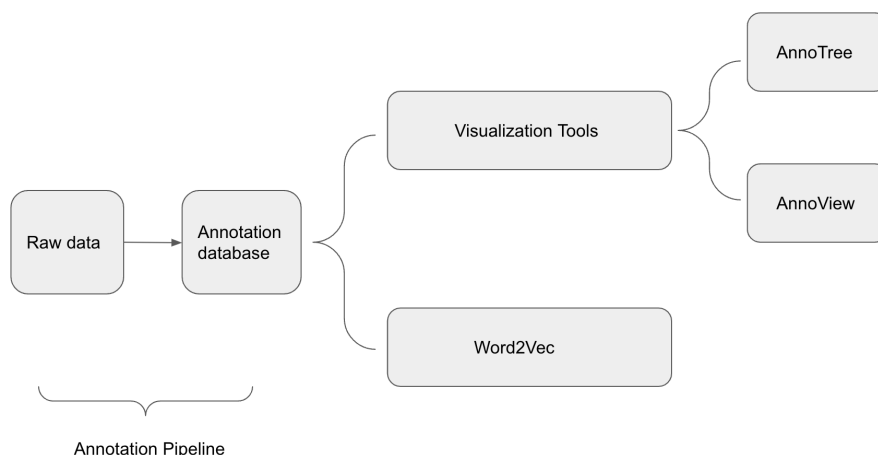


Figure 2.1: The AnnoTool pipeline can be divided into three parts: annotation pipeline, visualization tools, and the language model. It provides a complete workflow for exploring function prediction based on both phylogenetics and gene neighborhoods.

One major challenge in annotating these FASTA files is their large size, which makes it difficult to process on a personal system. This necessitates a distributed cloud-based solution. Compute Canada (now called the Digital Research Alliance of Canada) provides advanced research computing infrastructure and offers many computing instances called nodes to support researchers across Canada. It utilizes SLURM (Simple Linux Utility for Resource Management) [44] as its scheduler to manage and allocate resources on large-scale compute clusters. SLURM is an open-source job scheduler and workload manager designed to efficiently manage and allocate resources in high-performance computing (HPC) environments. Researchers can upload sequencing data to each node and run annotation tools concurrently, allowing many files to be annotated simultaneously instead of applying annotation tools to a single large file. For each computational node, the annotation pipeline first downloads the FASTA files from GTDB, concatenates them into one large FASTA file, and then splits this large file into smaller chunks. These chunks are evenly distributed across different nodes to balance the workload. Each chunk is a separate FASTA file. This approach ensures efficient processing and annotation of the data by leveraging the distributed computing resources.

Two computational tools are required for the annotation: KofamScan [18] and DIAMOND [15]. KofamScan is an HMM-based tool that assigns KO (KEGG Ortholog) IDs to each protein-coding sequence based on pre-computed family-specific adaptive thresholds. DIAMOND, another tool, works similarly to BLAST but with improved speed. It maps each sequence to the

UniRef100 database based on sequence similarity and assigns UniRef100 IDs to each sequence. A cross-reference file needs to be downloaded from InterPro, allowing the UniRef100 IDs to be mapped to InterPro IDs, as both databases are already linked. This mapping also shows a one-to-many relationship between the two databases; that is, given one UniRef100 ID, there are zero or many InterPro IDs corresponding to it, including no domain annotation, single domain annotation, and multiple domain annotations. Since InterPro includes 13 member databases, such as Pfam and Panther, the UniRef100 IDs can also be mapped to IDs from these member databases. Therefore, once DIAMOND completes the search, for a given protein-coding sequence from GTDB, the UniRef100 ID is assigned and represents the closest protein based on sequence similarity. The results will be stored as a TSV file on each computing node. All parameters for both tools are set to their default values. The outputs will be stored on each node on Compute Canada. Once the annotation part is complete, the final results will be aggregated together and filtered to ensure the annotations are reliable and accurate. For KofamScan, it assigns a score to the target query sequence and a threshold score for assigning a target KO family. If the score of the query sequence for a given KO family is higher than the threshold score, the output is highlighted with an asterisk at the beginning of the line. In this thesis, the filter will only retain hits highlighted with the asterisk sign.

For DIAMOND, the filter criteria are the same as those used in AnnoTree [35]. It will choose the UniProt ID with the highest bitscore value that satisfies the following criteria:

- e-value $\leq 1 \times 10^{-5}$
- Percentage of identical matches $\geq 30\%$
- Coverage of both query and subject sequences $\geq 70\%$

The coverage of query sequence is calculated as:

$$\left(\frac{|\text{End of alignment in query} - \text{Start of alignment in query}| + 1}{\text{Query sequence length}} \right) \times 100.0\%$$

Similarly, the coverage of subject sequence is calculated as:

$$\left(\frac{|\text{End of alignment in subject} - \text{Start of alignment in subject}| + 1}{\text{Subject sequence length}} \right) \times 100.0\%$$

After applying the filter, the filtered results will be uploaded to the remote server and loaded into a MySQL database [45]. MySQL is an open-source relational database management system (RDBMS) based on Structured Query Language (SQL). It is widely used for managing and

organizing data in a structured format and is developed, distributed, and supported by Oracle Corporation. MySQL stores data in tables, which are collections of related data organized in rows and columns. Each column in a table represents a specific attribute or key, while each row represents a record. In relational databases, tables can be connected using keys to establish relationships between them. The KEGG annotation table includes two attributes as keys: `gene_id` and `kegg_id`. The UniRef annotation table includes `gene_id`, `uniref_id`, and additional criteria used by the filter for reference. For the InterPro annotation table, additional files need to be downloaded from the InterPro website, including the mapping file between UniRef100 IDs and InterPro IDs, the mapping file between InterPro IDs and its member databases, and other metadata files containing descriptions of each entry. Once the mapping files are loaded into the MySQL database, given a `gene_id`, its corresponding InterPro ID and annotation ID from the member database can be retrieved by joining these tables. To optimize query speed, pre-computed accessory tables will be built based on the joined results. Thus, given a `gene_id`, MySQL can retrieve its corresponding UniRef100 ID, InterPro ID, and member database IDs in a few seconds. In addition, users may want to apply more strict or relaxed filters based on different needs. For example, a lower e-value for more closely related homologs or a higher e-value for more distantly related homologs. Since the MySQL database will not store every hit provided by the tools, users can parse the unfiltered files stored locally according to their needs (Fig 2.2).

The last step is to download additional accessory information from the GTDB and InterPro databases for more comprehensive data and to support the visualization tools AnnoTree and AnnoView. For AnnoTree, the topology of phylogeny and taxonomy are the most important components. This requires the Newick file for the tree structure downloaded from GTDB. The structure will first be parsed using preorder traversal, and node IDs will be assigned to each internode. This information will also be stored in MySQL, where each row contains a key called `parent` to declare the parent node. The taxonomy file for each GTDB ID (the unique ID for the representative genome of the species) and its corresponding GTDB taxonomy is downloaded and connected to the taxonomy mapping between GTDB taxonomy and NCBI taxonomy. This allows users to retrieve GTDB taxonomy and its data by providing NCBI taxonomy. For AnnoView, gene neighborhood visualization requires the coordinates of each gene for comparative analysis. All the raw FASTA files for all protein-coding gene sequences will be imported into the MySQL database, and a table called `coordinates` will be built. The fields of the `coordinates` table include `gene_id`, `gtdb_id` (the name of the file from which the target sequence comes), `contig_id`, `start`, `end`, `strand`, and `sequence`. This allows the gene neighborhood to be extracted by providing the coordinates. The `coordinates` table provides detailed information for each sequence and can connect to other annotation tables using the common field `gene_id`. All the tables are required to build indexes to optimize query speed. In addition, since the files for archaeal and bacterial microorganisms are separated in GTDB, all annotations and accessory

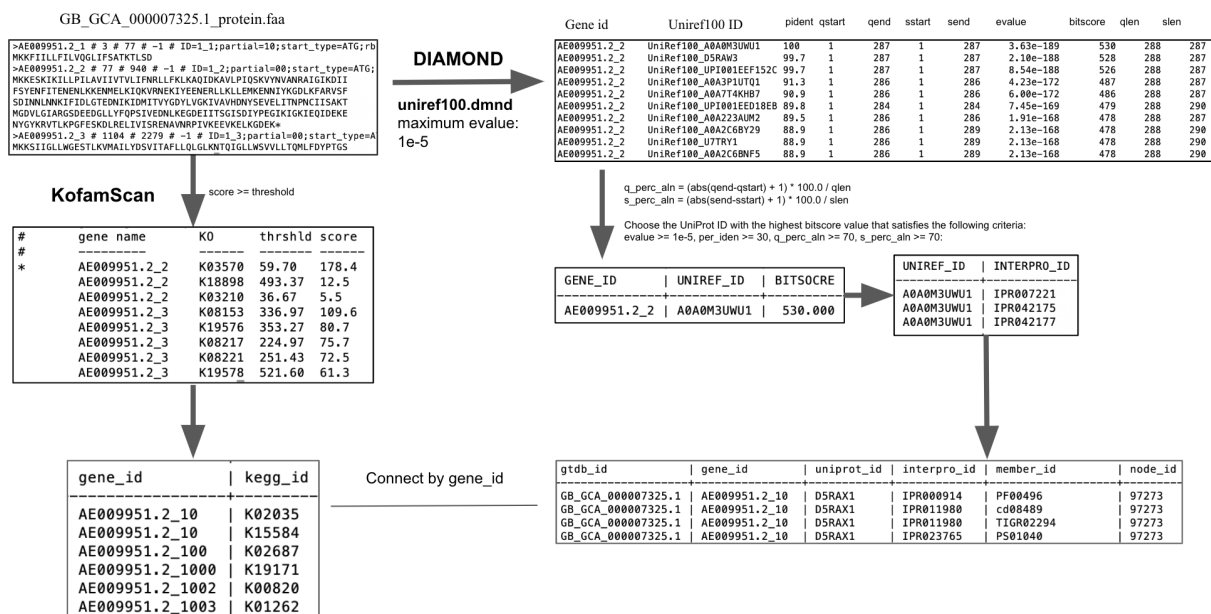


Figure 2.2: Workflow of the annotation pipeline, using the FASTA file for the genome with file ID GB_GCA_000007325.1_protein.faa from GTDB r214 as an example. The pipeline first annotates the protein-coding sequences from the FASTA file using Kofamscan and DIAMOND, and then filters the results. The filtered results are stored in a MySQL database, and the tables can be connected using keys.

files are also separated, but the schema in MySQL remains the same. One technical problem worth mentioning is that importing large files into the MySQL database may be restricted by computing resources such as memory. Therefore, a divide-and-conquer method is required so that large files, like the coordinates file, can be iteratively imported into the MySQL database (Fig 2.3).

2.2.1 Coverage Evaluation

In order to assess the coverage of annotations, all annotations for each genome are extracted, and the coverage for each genome with annotations from database X is calculated as:

$$\text{Coverage} = \frac{\text{Number of genes with annotations from database X}}{\text{Total number of genes}}$$

where X represents the annotation databases, including KEGG, InterPro, and its member databases. The coverage for each genome with annotations from any database is also calculated as 'Final' coverage:

$$\text{Final Coverage} = \frac{\text{Number of genes with annotations from any annotation database}}{\text{Total number of genes}}$$

The coverage evaluations were applied to both the archaea and bacteria databases separately and plotted using violin plots (Fig 2.4 and 2.5).

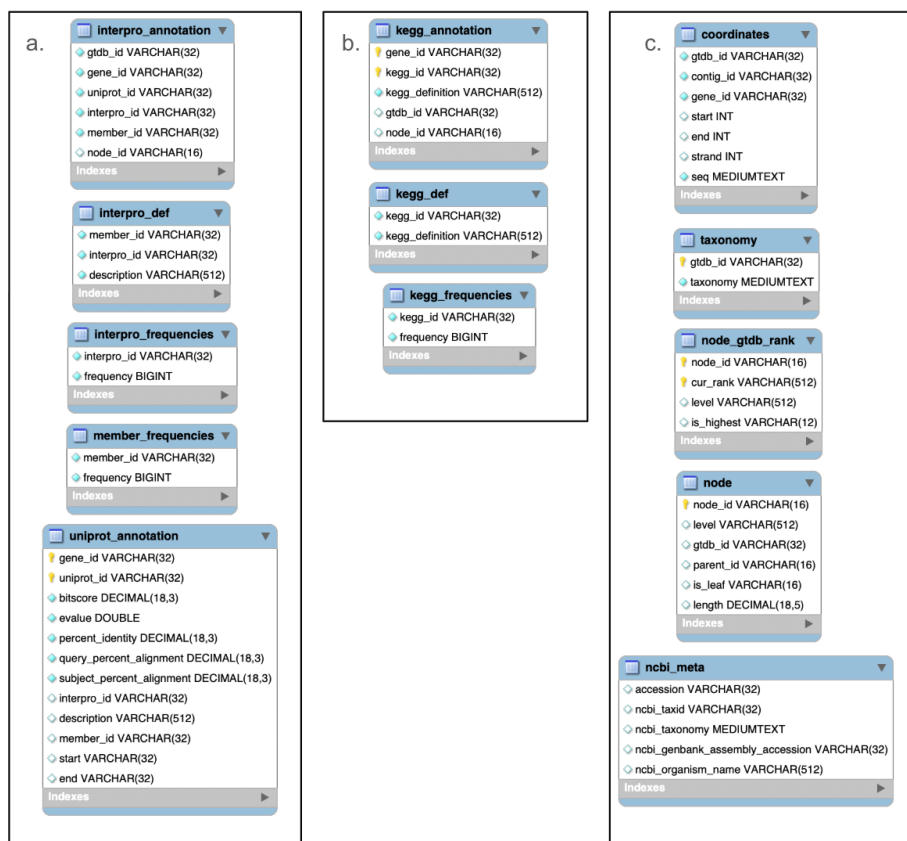


Figure 2.3: The MySQL database schema. Tables are grouped according to the function in which their containing data serves visualization tools and the language model. Each group contains the mapping among annotations and gene IDs, as well as other accessory information to describe different IDs.

(a) DIAMOND search results, including UniRef100 mapping results, InterPro and annotation ID from its member database, and the unique gene IDs for each protein-coding sequence from GTDB.

(b) KofamScan results, including KEGG orthology IDs and their corresponding gene IDs.

(c) Additional tables required for visualization tools and the language model. The coordinates table describes the position of each gene using start, end, and strand information, and also includes the raw sequences. Other tables in this group are used to build the phylogenetic tree from GTDB and link the genomes to their corresponding taxonomy.

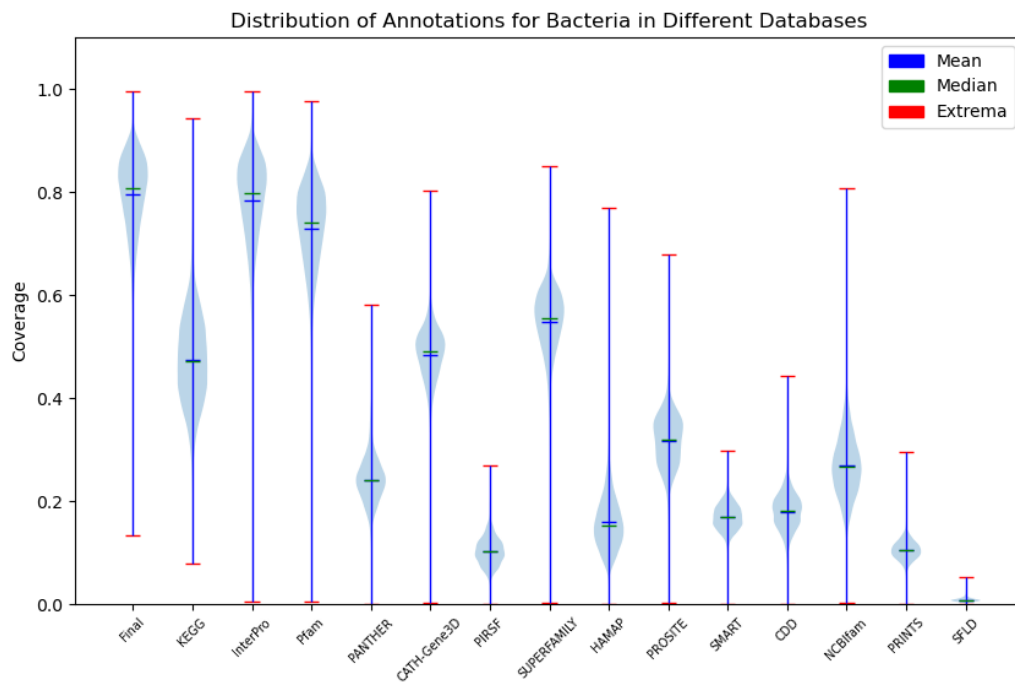


Figure 2.4: Distribution of annotation coverage for bacteria across various databases. The evaluated databases include KEGG, InterPro, and the 13 member databases of InterPro. The violin plots highlight the variability in annotation coverage and the combined metrics for bacteria.

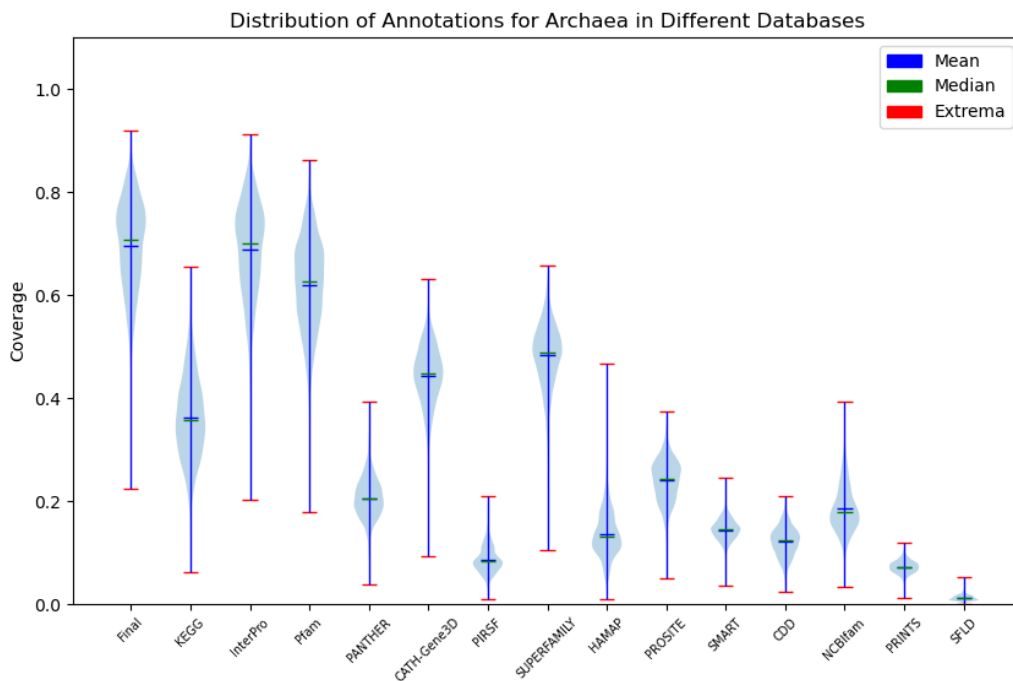


Figure 2.5: Distribution of annotation coverage for archaea across various databases. The evaluated databases include KEGG, InterPro, and the 13 member databases of InterPro. The violin plots highlight the variability in annotation coverage and the combined metrics for archaea.

In conclusion, the mean coverage for the 'Final' annotation database for bacteria is around 79%, and the median coverage is 82%. The distribution is skewed above the mean, meaning that a significant portion of the data has coverage values greater than both the mean and median, indicating a higher concentration of genomes with above-average coverage. The upper part of the violin plot is wider and extends further than the lower part. The mean and median coverage for KEGG annotation are around 50%, while the mean coverage for InterPro annotation is around 77%, and the median coverage for InterPro annotation is around 80%. This shows that most annotations are from the InterPro database, but KEGG can still provide some additional information for genes that are not identified by InterPro annotation.

The member databases within InterPro have varying coverage depending on their focus and definition. For example, Pfam has the highest overall coverage, while SFLD has the lowest among all member databases. This depends on the number of records collected in InterPro, where Pfam contains more than 16,000 entries, and SFLD has only 303 entries. The number of

entries varies based on the annotation methods and update frequency of each member database.

One issue worth noting is that some bacterial genomes may have unexpectedly low coverage. This occurs because, when the sequences from GTDB are mapped to the UniRef100 database, only the UniRef100 ID with the highest bitscore is chosen after applying the filter. The functional domain annotations are then extracted from the InterPro database for that UniRef100 ID. However, the UniRef100 ID may be deleted for several reasons, including the deletion of the corresponding coding sequence (CDS) in the source nucleotide sequence databases (EMBL-Bank/DBJ/GenBank) as requested by the original submitters, the deletion of the sequence prediction from Ensembl or RefSeq, or proteome redundancy reduction [16]. These deletions may be recorded by the InterPro database, resulting in no mapping between the deleted UniRef100 ID and InterPro ID. This discrepancy may not be reflected in the FASTA files provided for users to download, leading to inconsistencies when deletions occur. If most protein-coding sequences within a genome can be mapped with a high bitscore, and then the entire genomes are deleted, all the mappings will be affected due to the inconsistency between the InterPro and UniRef100 databases. For example, if a UniRef ID (e.g., ID A) is used to map a sequence and that ID is later removed from the UniRef100 database, the corresponding mapping in the InterPro database will also become invalid.

Currently, there are 46 genomes in the bacterial database affected by this issue, with an average coverage below 15%. The current bacterial database contains 80,789 genomes, so only 0.05% of the genomes are seriously affected. One way to solve this problem is to check if the corresponding InterPro ID exists before choosing the UniRef100 ID with the highest bitscore. If it does not exist, the next UniRef100 ID with the lower bitscore should be checked until a mapping exists that satisfies the filter requirements. However, this solution requires additional computational cost compared to a single mapping. The number of UniRef100 IDs to be checked needs to be further investigated to find optimized hyperparameters that ensure the speed is acceptable and the low coverage issue is manageable.

2.2.2 Future Work

As of April 24th, 2024, the newest version of GTDB has been released, with the number of bacterial genomes increasing from 80,789 to 107,235, and the number of archaeal genomes increasing from 4,416 to 5,869. This results in an almost 35 percent increase in the total number of genomes. Updating the annotation database regularly ensures its accuracy and maintains its effectiveness over time. Therefore, one major future task is to run the annotation pipeline every year or two, depending on the availability of computational resources.

Currently, the annotations are primarily linked with protein functional domains from databases

such as KEGG and InterPro. However, other databases, such as GO for Gene Ontology, AlphaFold for predicted protein structures, and additional information like isolation types and sample details, can provide more comprehensive data for a given protein-coding sequence. Thus, there are various options for adding other types of annotations in the future, leading to more comprehensive and informative annotations.

2.3 Advancements in AnnoTree and AnnoView: Development of Version 2.0

The two visualization tools, AnnoTree [35] and AnnoView [36], developed by the Doxey lab, primarily rely on the annotation database and can visualize annotations by phylogeny and by neighborhoods, respectively. Both tools originally utilized the annotation database built on GTDB release r95 and supported functional domain annotations from the KEGG, Pfam, and TIGRFAM databases. The new annotation database is significantly expanded in terms of the number of genomes, taxonomy, and types of annotations. This expansion changes the topology of the tree in AnnoTree and alters the gene neighborhoods in AnnoView.

Additionally, with the new annotation database, new features have been developed for both tools to better utilize the expanded data. To accommodate the expansion and changes, necessary updates have been implemented. The details of these implementations are described in the following subsections.

2.3.1 AnnoTree Web Application

AnnoTree is a web application that focuses on the distribution of protein functional domains and the taxonomy of genomes from GTDB. It consists of three parts: the database, frontend, and backend. In the original version (version 1.0), the database was built on GTDB release 95, which consisted of 30,238 bacterial genomes and 1,672 archaeal genomes. Each leaf of the corresponding phylogenetic tree is a representative genome of the species in GTDB. The protein functional annotations are sourced from the KEGG, Pfam, and TIGRFAM databases, all stored in a MySQL database on a remote server. Since the number of genomes has significantly increased compared to the old database, the structure of the phylogenetic tree has also changed accordingly. The nodes in the phylogenetic tree have increased from 60,475 to 161,577 for bacteria and from 3,343 to 8,831 for archaea (Fig 2.6 and 2.7).

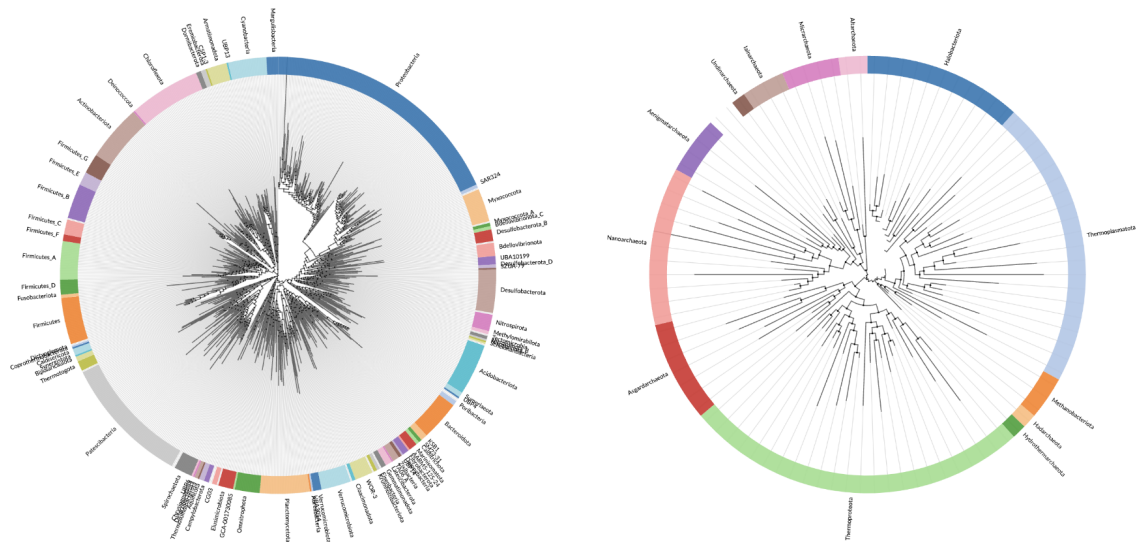


Figure 2.6: Left: Phylogenetic tree visualization for the GTDB r95 bacteria database using the AnnoTree application.
 Right: Phylogenetic tree visualization for the GTDB r95 archaea database using the AnnoTree application.

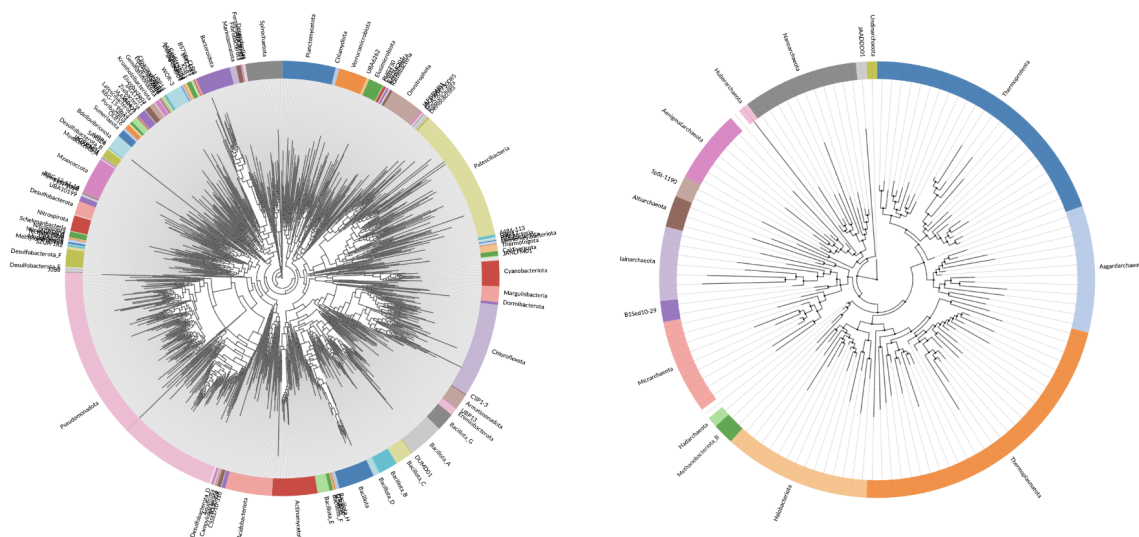


Figure 2.7: Left: Phylogenetic tree visualization for the GTDB r214 bacteria database using the AnnoTree application. Right: Phylogenetic tree visualization for the GTDB r214 archaea database using the AnnoTree application. The number of internal nodes has significantly increased from GTDB r95.

One common use case involves providing a combination of protein annotations and then finding all genomes that contain this combination. This is followed by highlighting the path from the root node to the leaf node on the phylogenetic tree. The original implementation to achieve this operation on the MySQL database involves, for each genome, finding all annotations in the combination, grouping the hits by the occurrence of annotations, and then counting the number of distinct hits within the genome where the count is greater than or equal to the number of annotations in the search. This operation is effective and common when the table size is small, but it becomes slow on tables with millions of rows. While it showed results correctly in AnnoTree 1.0, the increased size of the database in the updated version has slowed down this process, affecting the user experience.

In order to optimize the user experience, an improved solution has been implemented. For preparation, indexes on each table and pre-computed mappings between the genome ID and annotations are built. For the query, it first searches for the genomes containing each annotation in the combination separately and then finds the intersections among all the resulting genomes. This method can significantly accelerate the query speed since the single annotation query is indexed and can be achieved within several seconds on average.

In the worst-case scenario, the InterPro ID IPR027417 (P-loop containing nucleoside triphos-

phate hydrolase) has the highest occurrence, with 27,280,335 entries in the database. The single annotation query for this annotation takes around 30 seconds. The InterPro ID IPR001789 (Response regulator receiver domain) has the second highest occurrence, with 8,307,022 entries, and it takes around 10 seconds for this query. Other annotation IDs take within 10 seconds depending on the count of the annotation ID in the annotation database.

Therefore, the approximate time for querying the bacterial database, excluding IPR027417, can be calculated as:

$$\text{Searching Time} = N \times T + t$$

where:

- N is the number of annotations in the combination,
- T is the time for a single annotation query,
- t is the time for finding the intersection among all genomes.

The upper limit for a single annotation query is within 10 seconds (except for IPR027417), and the upper limit for the resulting genomes for each single annotation query is 80,789, which is the total number of genomes in the bacterial database. The time cost increases linearly with the increase in the number of genomes, so this method can also handle future updates of the database.

The time for running a single annotation query for the 10 highest occurrence InterPro IDs can be plotted (Fig 2.8).

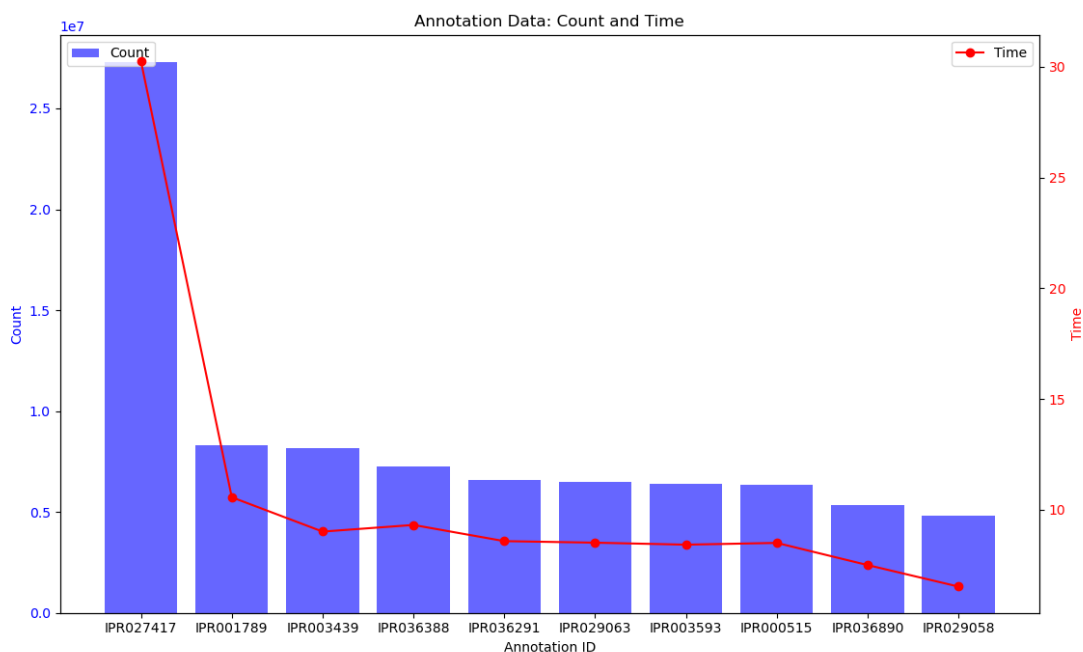


Figure 2.8: Top 10 occurrences of annotation IDs and their corresponding search times in seconds in the database. Most annotation IDs can be retrieved in less than 10 seconds.

Therefore, by using the intersection of single annotation queries, the query does not need to group the hits by occurrence, which was the major time-consuming step in version 1.0.

As a comparison, the new method for finding “which genomes contain PF04717 (Phage base V) and PF13503 (Domain of unknown function DUF4123)” takes only 1 second for 2,799 hits in the GTDB r214 database. In contrast, it takes 21 seconds for 1,759 hits in the GTDB r95 database and 95 seconds for 1,759 hits in the GTDB r214 database using the old method. Both queries were run on a remote server with the same configuration: 16 GB memory and 2 CPUs (Fig 2.9 and 2.10).

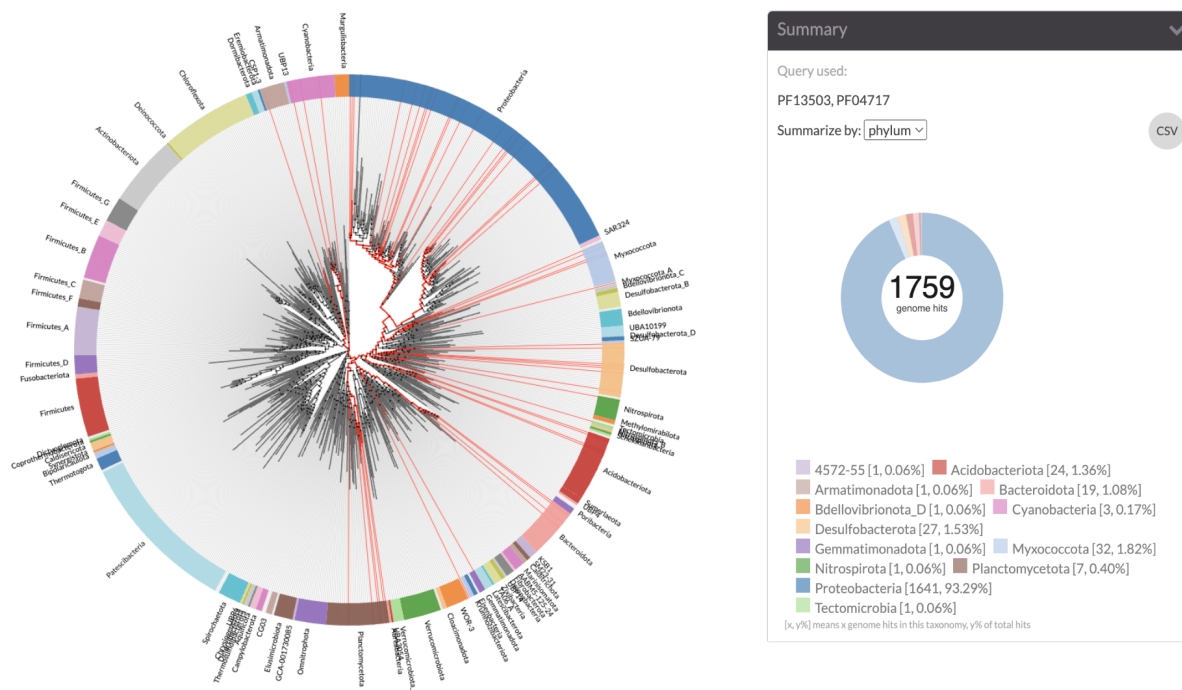


Figure 2.9: AnnoTree 1.0 Application Results: For the search query of PF04717 (Phage base V) AND PF13503 (Domain of Unknown Function DUF4123) in the GTDB r95 bacteria database, the results indicate that 1,759 bacterial genomes contain this combination. The query took 21 seconds to execute.

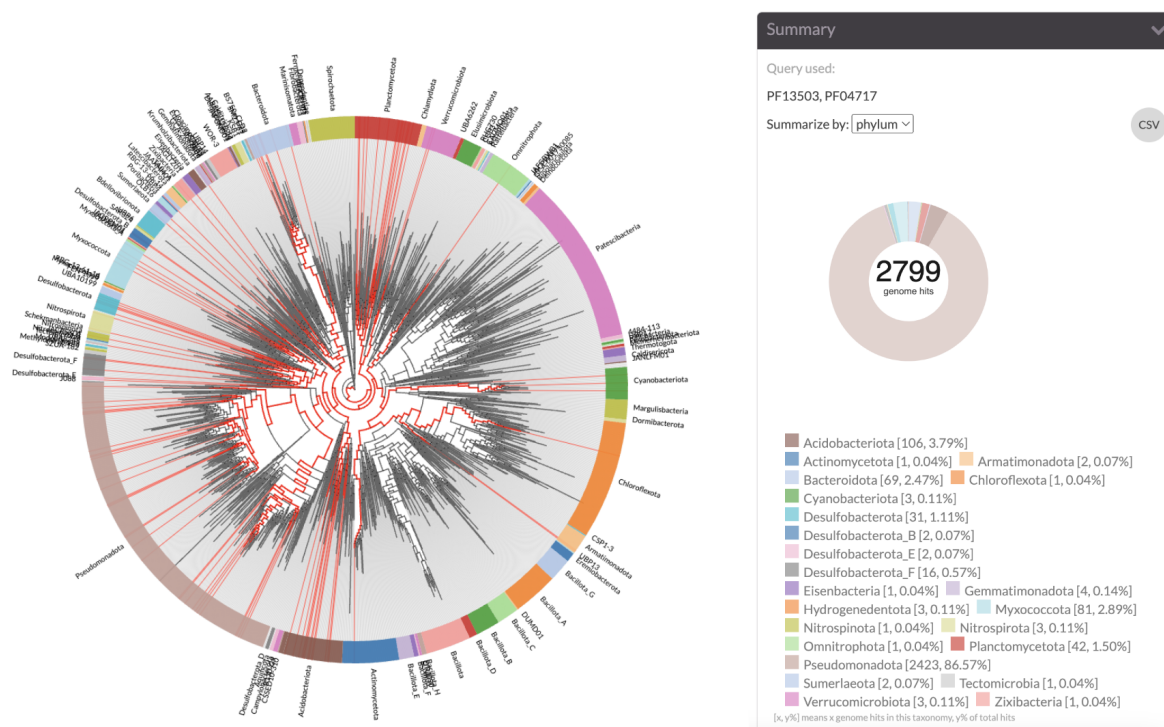


Figure 2.10: AnnoTree 2.0 Application Results: In the GTDB r214 bacteria database, the same search query for PF04717 (Phage base V) AND PF13503 (Domain of Unknown Function DUF4123) identified 2,799 bacterial genomes containing this combination. The query execution time was reduced to just 1 second.

Another major update is the search options. In version 1.0, users could only search for annotations of the same type. For example, users could only search for the distribution of Pfam IDs but not for the distribution of Pfam IDs and KEGG IDs together. This limitation existed because, in version 1.0, the tables were stored separately for each annotation database. In version 2.0, most annotation databases are member databases of InterPro, and they can all be mapped to InterPro IDs within the same table. This method allows for searching annotations against the InterPro database and its 13 member databases simultaneously. If users have a query for annotations, it first checks the distribution within the InterPro database and then finds the intersection with the KEGG database since these two databases are not linked together. Using this method, AnnoTree 2.0 can correctly parse annotations and find results in a very short time. For example, in version 1.0, finding “which genomes contain PF04717 (Phage base V) and K02406 (flagellin)” was not allowed since the domains are from different databases. However, it can be achieved in version

2.0 without any additional time cost to handle the different annotation data sources.

Other user-interface updates include interactive selection for internal nodes in the tree, which highlights the subtree of the given taxonomy, and text highlighting when the mouse hovers over. These changes and other improvements have been implemented to support a better user experience.

2.3.2 AnnoView Web Application

AnnoView is a web application that visualizes gene neighborhoods, which are colored based on the composition of their functional domain annotations. In AnnoView version 1.0, there are two major features:

1. Customized file upload and visualization, which supports .gbk, .gff, or .csv files, allowing genes to be sorted by those with the same annotations.
2. Given a sequence, search for its homologous sequences against GTDB r95 using DIAMOND, and visualize the gene neighborhoods around the sequences in a ± 10 kilobases range.

In AnnoView 2.0, the functionality of searching sequences against GTDB has been improved, relying on GTDB r214 and supporting protein functional domain annotations from KEGG, Pfam, TIGRFAM, InterPro, and its 13 member databases. This means the data in version 2.0 is a superset of the data in version 1.0. In addition to functional domains, since each gene from GTDB has been mapped to the UniRef100 database and assigned a UniRef100 ID, those IDs can then be mapped to the AlphaFold database for protein structure prediction (if the mapping exists). This allows users to visualize the AlphaFold prediction of the closest homologous sequence from the UniRef100 database for a given gene. Mapping details, including e-values and identities, are also provided for reference. Additionally, users can now choose the range of interest surrounding the target gene (from ± 5 kilobases to ± 30 kilobases), instead of the fixed ± 10 kilobase range in version 1.0. The intermediate page in the GTDB search, which is used to parse the DIAMOND results and allow users to select the genome of interest to visualize before the actual gene neighborhood visualization page, is now linked to AnnoTree 2.0. This linkage allows users to visualize the distribution of genomes containing homologous genes across the tree of life.

Another major update for the AnnoView application is the new feature called combination search. The combination search requires one or more protein functional annotations as input, called a combination, and then uses the first ID in the combination as the center or anchor to

check if the combination exists within the specified range surrounding the center gene. This is useful for users interested in specific gene clusters or pathways. For instance, if users are interested in gene clusters like the type VI secretion system (T6SS), which is composed of 13 conserved, essential core components and several accessory components, they can choose any proteins of interest belonging to the core components of T6SS and then search for the combination using AnnoView 2.0. Questions like “What are the gene neighborhoods around PF04717 (Phage base V)?” or “What is the gene context for gene clusters that contain the domains PF04717 (Phage base V) and PF20249 (VasX toxin N-terminal region)?” can now be answered and visualized.

The method to achieve the combination search uses the ‘bedtools intersect’ command [46], which is a command-line tool from the BEDTools suite widely used for genomic analysis. Given intervals in two files, A and B, with coordinates ‘start’ and ‘end’ for each interval, where the ‘start’ and ‘end’ are from the coordinates table in the MySQL database and processed from the FASTA file provided by GTDB, it can find the intersection or overlap of all the intervals using the Sweep Line Algorithm [47], with a time complexity of $O(n \log n + m \log m)$, where n and m are the number of intervals in files A and B, respectively. In the case of combination search, each annotation has its own start and end, so it can be treated as an interval. Finding if the combination exists within the window surrounding the center gene is equivalent to finding if the intersection or overlap between the interval of the window and all the intervals of each ID exists (Fig 2.11 and 2.12).

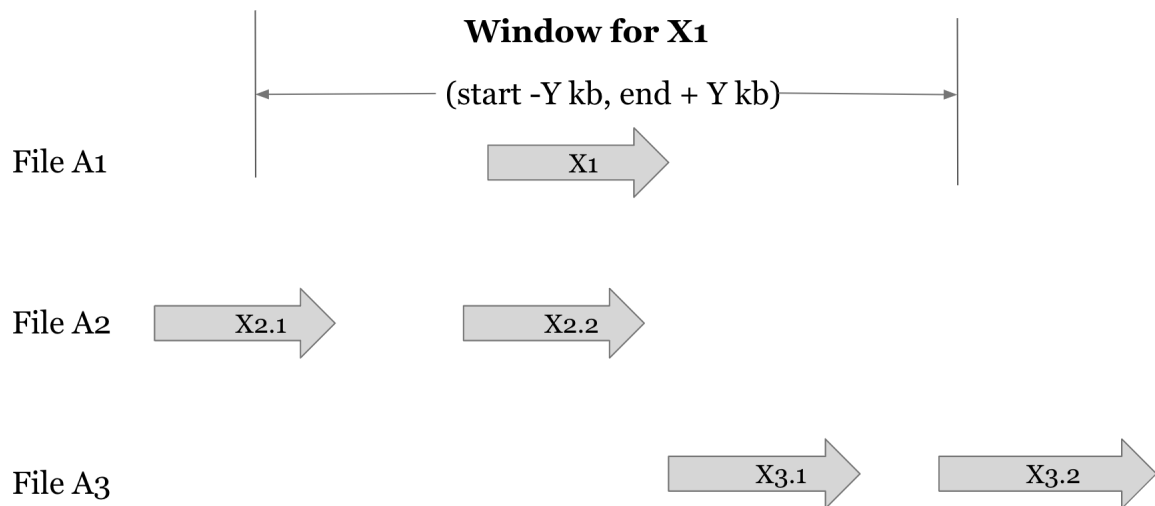


Figure 2.11: In this workflow, the `bedtools intersect` command is used to identify intervals that contain specific annotation IDs. The intervals X_1 , $X_{2.1}$, $X_{2.2}$, $X_{3.1}$, and $X_{3.2}$ represent the coordinates of the corresponding annotation IDs. Specifically, $X_{2.1}$ and $X_{2.2}$ are two intervals with annotation ID X_2 , while $X_{3.1}$ and $X_{3.2}$ are two intervals with annotation ID X_3 . The search checks whether intervals with annotation IDs X_2 and X_3 are subintervals within the window centered on X_1 . In this case, $X_{2.2}$ and $X_{3.1}$ are subintervals of the X_1 -centered window, so X_1 is considered a valid candidate that contains the target combinations.

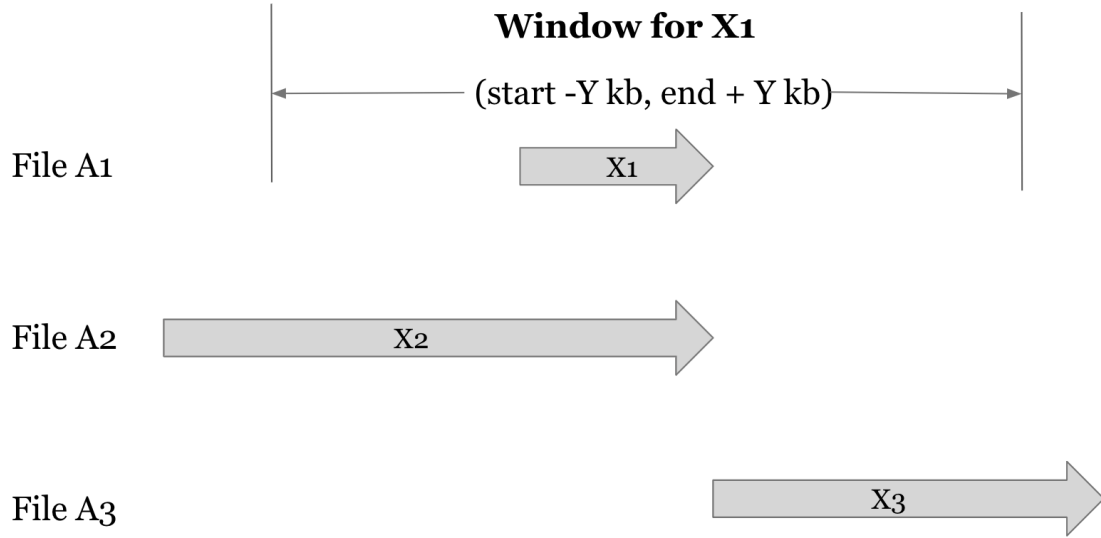


Figure 2.12: Similar to figure 2.11, if X_2 and X_3 are not subintervals within the window of X_1 , then there are no valid candidates containing the target combinations.

The detailed steps are as follows. Assume there are n annotation IDs in the combination search from X_1 to X_n , and the window length is Y kilobases:

1. Assume the first annotation ID in the combination search is X_1 . The combination search will first locate all genes with annotation X_1 , and then use the 'start - Y ' and 'end + Y ' of each gene as intervals, which are also the windows that need to be checked for the combination. All intervals are saved in file A_1 .
2. For the following annotation IDs from X_2 to X_n , search all genes with the corresponding ID, and then save the 'start' and 'end' of each gene in file A_i as intervals, where i is the index of the ID in the combination. Use the 'bedtools intersect' to find all windows in file A_1 that are supersets of one or more intervals in file A_i .
3. Find the intersection of all the outputs, which represents the windows containing all IDs in the combination search. The result is the gene ID with the annotation X_1 .

Once the search is finished, the genes within the windows can be extracted and parsed as a tree structure for users to select the taxonomy of interest to visualize.

The speed of ‘bedtools’ in processing the intersection is fast and can normally be done within seconds. However, some IDs in the combination search may correspond to millions of genes. A typical example is IPR029058 with 4,802,892 hits in the database. If this ID is the only annotation to search, then all 4 million genes with their corresponding gene IDs will be extracted for the taxonomy tree, which could overload the application. One possible solution is to limit the size of the data when users search. For example, limit the search by taxonomy, or set an upper limit for the number of hits for the combination, and force users to add more IDs in the combination so that the final number of output gene IDs is below the upper limit.

2.4 Language Model for Protein Function Prediction

NOTE: The work described in this section appears in the following manuscript: Huagang Tan, Gabriel Moreno-Hagelsieb, Andrew C Doxey. (2024) Gene function prediction using an AnnoTree-based genomic language model. 2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) [48].

Context is fundamental in natural language processing (NLP), where the context of words in a sentence is used to understand their meanings. Modern NLP techniques train deep learning or machine learning-based models on vast text corpora, including sources like Wikipedia and news articles, to develop numerical representations of words, known as “embeddings.” These embeddings are crucial for various applications, such as text classification and conversational agents [49]. They can cluster words with similar contexts in high-dimensional space; the closer the embeddings, the more similar the contexts of the words. Recently, similar NLP methods have been applied to “protein languages”, predicting amino acid properties based on their context within protein sequences [50, 51, 52]. This approach has been employed to model protein characteristics. Additionally, previous work [38, 39] describes language models using protein functional domains instead of amino acids to classify biosynthetic gene clusters and predict protein function. These models focus on domain-level language models instead of protein sequence-level language models, providing a higher-level perspective.

2.4.1 Word2Vec Model

The Word2Vec model [37] is a popular neural network-based model developed by researchers at Google that used natural language processing (NLP) to generate word embeddings—numerical representations of words that capture their meanings and semantic relationships. Word2Vec transforms words into vectors of fixed size, enabling the capture of syntactic and semantic

similarities between words. It comprises two primary architectures: Continuous Bag of Words (CBOW) and Skip-gram (Fig 1.3). In the CBOW model, the objective is to predict a target word based on its surrounding context words within a fixed window size, effectively capturing the semantic relationship by averaging the context word vectors. Conversely, the Skip-gram model, which is used in this thesis, aims to predict the context words given a target word, optimizing the probability of context words occurring around a given word. Both architectures utilize shallow, two-layer neural networks to transform words into dense, low-dimensional vectors known as word embeddings. These embeddings encode semantic similarities and syntactic relationships, facilitating a wide range of natural language processing tasks. The training process involves adjusting the weights of the neural network to minimize the prediction error, thereby creating a vector space where words with similar meanings are positioned closely together.

In this thesis, the genomic corpus, which is processed from the annotation database built on GTDB r214, will be used to train the Word2Vec model. Additionally, only Pfam IDs [27] are considered for training the model because:

1. Pfam has the highest coverage percentage among the member databases of InterPro, and the coverage is also much higher than the KEGG database.
2. Each annotation ID from the Pfam database does not overlap, so the words are naturally separated. In contrast, annotation IDs from the InterPro database may overlap because they may link to different member databases.
3. The Pfam database contains annotation IDs labeled as domains of unknown function (DUF) [40]. DUFs represent protein domains for which no clear functional information is available. These domains are identified based on sequence similarity and structural features, but their specific biological roles or functions remain unclear. By applying the Word2Vec model to Pfam annotations, the functions of DUFs may be revealed based on genomic context.

A previous word2vec based genomic language model treats each functional domain as an individual word and focuses on domain-level functional prediction. Consequently, every functional domain has its individual and fixed embedding after training. However, the function of a domain may not be unique, and the fixed vector embedding may ignore the potential functions of the domain, leading to inaccurate representation. This also means that the genomic context of a given functional domain may vary significantly, causing the embedding to be biased.

An example in natural language includes the English word “park,” which can be a verb in the sentence “I park the car” or a noun in the sentence “there is a park.” The context is different, and the encoding relies on the predominant scenario in the corpus.

In the example of protein functional domains, consider the domain PF04717, which is a domain of bacterial and phage baseplate assembly proteins responsible for forming the small spike at the end of the tail or bacterial pathogenic needle-shaft [53]. It can belong to VgrG (valine/glycine-repeat protein G), the ortholog of the phage T4 cell-puncturing device found in the T6SS, commonly co-occurring with PF13296 (T6SS Vgr), but it can also co-occur with PF18946 (Apex) [54], a short motif found at the C-terminus of Phage gpV proteins. These combinations do not share similar genomic contexts, meaning the Pfam ID PF04717 has multiple semantic meanings depending on its context. In this case, no T6SS cluster is found around the combination of PF04717 and PF18946 in a 30 kilobases window. If the Word2Vec model treats PF04717 as a single word, it may ignore the lower frequency occurrences and mislead the interpretation of the word embedding of this protein functional domains.

Here, the solution provided in this thesis is, instead of treating the proteins of multi-domains as a list of individual words, to concatenate the domains within one protein as a single word and then train the Word2Vec model. In this case, the model will focus more on the protein level instead of the domain level, and can disambiguate to some extent. For example, if a protein is annotated as PF04717 and PF18946, the original Word2Vec would treat it as two separate words for domain-level splitting. The modified version of the Word2Vec model would concatenate both annotation IDs with a comma and use “PF04717,PF18946” as a single input word to train the model, focusing more on the protein level.

The cost of this modification is that the dictionary size (total number of words in the model) increases, leading to an increase in the model’s parameters and training time.

The annotation IDs are then mapped to the genes so that each contig can be treated as a sentence similar to natural language. Some genes may not have corresponding domain annotations. To maintain the genomic structure, genes without any annotations will be labeled as ‘GAP’ with the length as the prefix. The lengths are discretized by a bin size of 500. Therefore, for genes without any annotations, the words in the genomic corpus will be ‘GAP500’, ‘GAP1000’, ‘GAP1500’, corresponding to genes with lengths under 500 bases, between 500 and 1000 bases, and between 1000 and 1500 bases, respectively. This method was used in previous studies that used InterPro annotations as words instead of Pfam [38].

By using the annotation database as a corpus, which is also the training dataset, the model is trained on 11 million contigs and over 240 million genes with annotations. Thus, one gene is represented as a list of single protein domains. The Skip-gram architecture is used because it can better capture the semantic meaning.

The Skip-gram architecture of the Word2Vec model finds vector representations of protein domains that maximize the probability of context protein domains given a target protein domain. Given a sequence of training protein domains (words) $w_1, w_2, w_3, \dots, w_T$, the objective is to

maximize the average log probability

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log P(w_{t+j}|w_t),$$

where c is the context window size. The conditional probability $P(w_{t+j}|w_t)$ is defined using softmax:

$$P(w_O|w_I) = \frac{\exp(v_{w_O} \cdot v_{w_I})}{\sum_{w \in V} \exp(v_w \cdot v_{w_I})},$$

where v_{w_I} and v_{w_O} are the vector representations of the input (target) and output (context) protein domains, respectively, and V is the vocabulary. Direct computation of the softmax is computationally expensive, so efficient approximation techniques like Negative Sampling and Hierarchical Softmax are used. Negative Sampling approximates the softmax by updating vectors for only a small sample of negative examples, with the objective:

$$\log \sigma(v_{w_O} \cdot v_{w_I}) + \sum_{i=1}^k \mathbb{E}_{w_i \sim P_n(w)} [\log \sigma(-v_{w_i} \cdot v_{w_I})],$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function, k is the number of negative samples, and $P_n(w)$ is the noise distribution. Hierarchical Softmax uses a binary tree representation to reduce the computational complexity. The model is trained using stochastic gradient descent (SGD), where the gradient of the log probability is computed for the target protein domains and their context protein domains, and the vectors are updated iteratively [37].

In the actual implementation, the training process and inference of the model use Python 3 and the Gensim library [55]. The window size is set to 6 (meaning the model processes 6 functional annotations at a time during training), the dimension of the vector is 200, and the weight decay is 0.1. All other parameters are set to default values.

The model is now hosted in AnnoView application, and can infer the result within seconds. (Fig 2.13).

A

Please enter single Pfam ID or Pfam IDs

PF20455 --- Domain of unknown function DUF6708 [Pfam] ✕

Database: Bacteria

Top similar words (between 1 and 50): 30

Model: Word2vec

Search

B

Proteins with similar context

Proteins that contains target domain(s)	Score
PF20249 (VaaX_N)	0.838650107383728
PF04717 (Phage_base_V) PF10106 (DUF2345) PF13296 (T6SS_Vgr)	0.7128921747207642
PF13503 (DUF4123)	0.6856477856636047
PF05488 (PAAR_motif)	0.6688161492347717
PF11745 (DUF3304)	0.6498157382011414
PF05638 (T6SS_HCP)	0.6395726203918457
PF06744 (lcmF_C) PF06761 (lcmF-related)	0.5912566184997559
PF10106 (DUF2345) PF13296 (T6SS_Vgr)	0.5912296175956726
PF13296 (T6SS_Vgr)	0.5690950155258179
PF05488 (PAAR_motif) PF11187 (Mbeg1-like)	0.562502384185791

1 2 3

Figure 2.13: (A). Search bar in AnnoView application. (B). Word2vec prediction for top semantically similar words.

2.4.2 Application

After training, all Pfam annotation IDs within genes are encoded as vectors of 200 continuous dimensions, enabling the calculation of differences between vectors. The dictionary of annotations can be treated as a 200-dimensional space, where the annotations are vectors within this space. Each vector represents a gene with functional domain, so genes with similar genomic contexts are close to each other in the space. The space can also be visualized using dimensional reduction techniques like PCA (Principal component analysis) [56], where functionally related domains are close to each other. Examples include gene clusters for secretion systems (Fig 2.14). For DUFs in gene clusters of known functions, they may play potential roles in the gene cluster and interact with other protein domains within the same cluster. They can have either a co-occurrence or anti-co-occurrence relationship in the gene cluster because they share similar genomic contexts.

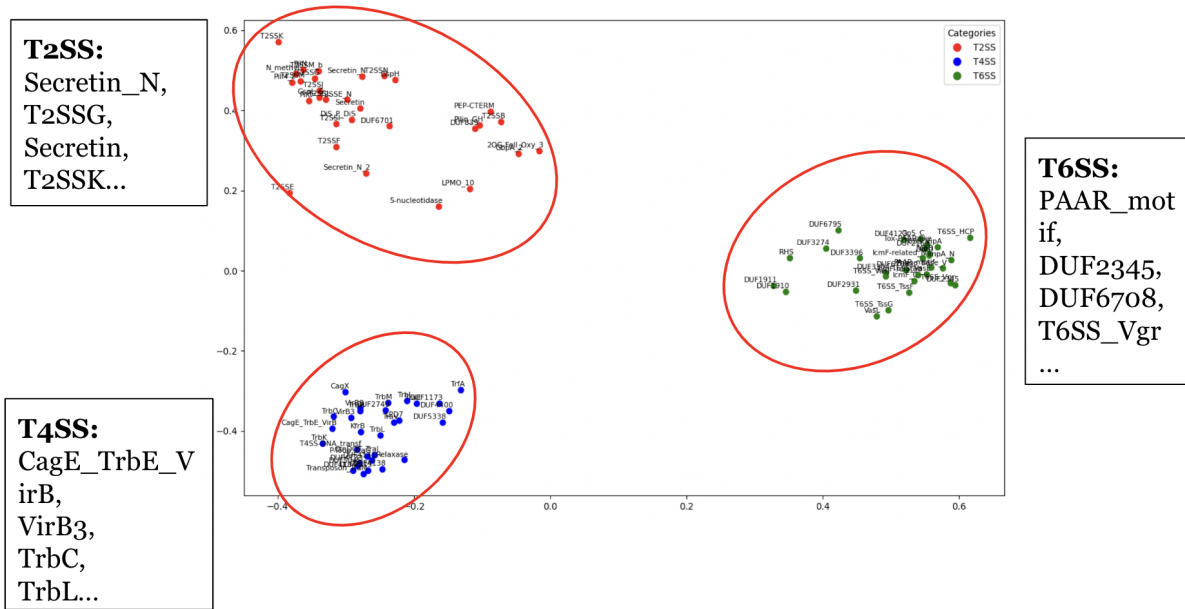


Figure 2.14: PCA visualization of the protein global domain embedding space, with the dimension reduced from 200 to 2. The red circles highlight clusters including the type II secretion system (T2SS), type IV secretion system (T4SS), and type VI secretion system (T6SS). Each dot within the cluster represents an individual Pfam domain, and 30 dots from each gene cluster are plotted randomly, the short names of the domains are labeled in the figure.

As an example, consider one of the effector proteins within T6SS, whose signature domain is assigned the Pfam ID PF20249. This domain represents the N-terminal region of the VasX toxin from *Vibrio cholerae* and acts as the domain of effector proteins in T6SS that can be injected into other organisms (Fig 2.15). The ID is encoded as a vector of 200 dimensions after the model is trained.

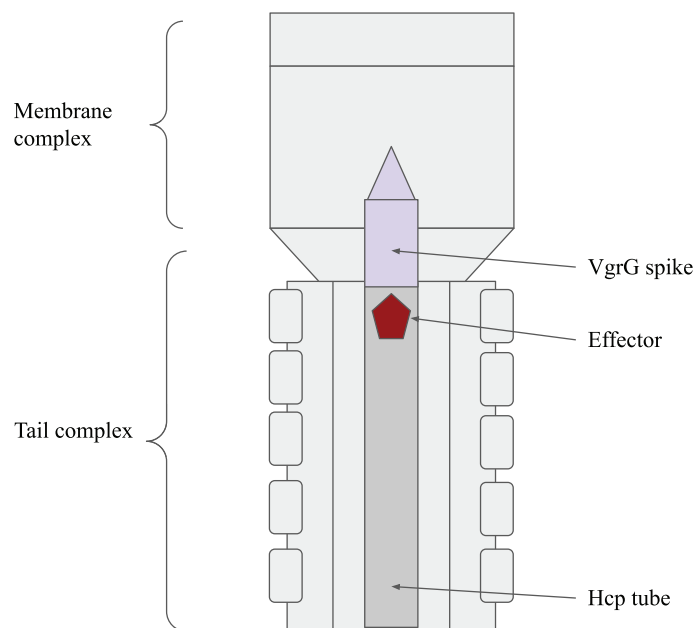


Figure 2.15: The T6SS tail complex attaches to the membrane complex. Toxic effectors are recruited to the spike-tube complex through the extension domains of VgrG and/or PAAR-repeat proteins, or by incorporating into the haemolysin co-regulated protein (Hcp) tube. Immunity proteins are produced alongside effector proteins. Their primary function is to neutralize or inhibit the activity of the effector proteins that are secreted by the same bacterial cell. This prevents self-damage or autotoxicity [57]. Figure modified from [58].

The table below presents the results from the trained model when using Pfam ID PF20249 as the target word. It lists the top 50 functional domain IDs with the most similar genomic contexts. The table includes the following columns: annotation (domain) ID, similarity score, description, and relevant links to related articles. Domain IDs are sourced from the Pfam database, and the similarity scores are computed using cosine similarity between vectors. Links are provided only for Domain of Unknown Function (DUF) entries, as the functions of other domain IDs are already known.

The annotation IDs are categorized into three classes:

1. Domains within the necessary proteins in T6SS, including proteins in the membrane complex, VgrG spike protein, tail complex, and Hcp tube. They commonly co-occur with the effector protein because they translocate the effector protein into target cells.
2. DUFs for which the functions have been predicted in prior studies, as documented in related articles. These include domains in different effector/immunity protein pairs that interact with necessary proteins from T6SS. These proteins pairs often co-occur with other necessary proteins in T6SS but anti-co-occur with other effector/immunity protein pairs.
3. DUFs with few or no associated articles, indicating unknown functions.

Annotation ID	Score	Description	Reference
PF20455 (DUF6708)	0.9123	Immunity protein	[59]
PF13296 (T6SS Vgr)	0.8399	VgrG related	
PF10106 (DUF2345)	0.8395	VgrG related	[60]
PF11745 (DUF3304)	0.8239	Immunity protein tliV1	[61]
PF13503 (DUF4123)	0.8121	Chaperone proteins	[62]
PF05638 (T6SS HCP)	0.7985	T6SS component	
PF05488 (PAAR motif)	0.7880	T6SS component	
PF04717 (Phage base V)	0.7501	VgrG related	
PF06812 (ImpA N)	0.7377	T6SS component, TssA	
PF16989 (T6SS VasJ)	0.7291	T6SS component, TssA	
PF05591 (T6SS VipA)	0.7164	T6SS component, TssB	
PF06761 (IcmF-related)	0.7072	T6SS component, TssM	
PF11153 (DUF2931)	0.7067	Immunity protein tli1, tli2	[63]
PF05943 (VipB)	0.7055	T6SS component, TssC	
PF06715 (Gp5 C)	0.7051	VgrG related	
PF06744 (IcmF C)	0.7050	T6SS component, TssM	
PF08929 (DUF1911)	0.7025	Immunity protein PoNi	[64]
PF05947 (T6SS TssF)	0.6994	T6SS component, TssF	
PF11876 (DUF3396)	0.6992	Immunity protein TsiV1-4	[65]
PF09850 (DotU)	0.6983	T6SS component, TssL	
PF05936 (T6SS VasE)	0.6925	T6SS component, TssK	
PF12790 (T6SS-SciN)	0.6923	T6SS component, TssJ	
PF14331 (IcmF-related N)	0.6883	T6SS component, TssM	
PF13665 (Tox-PAAR-like)	0.6797	T6SS component	
PF18945 (VipB 2)	0.6793	T6SS component, TssC	
PF09937 (DUF2169)	0.6791	Adaptor protein	[66]
PF19940 (DUF6402)	0.6791	UNKNOWN	
PF03527 (RHS)	0.6764	Effector protein domain	
PF08928 (DUF1910)	0.6746	Immunity protein PoNi	[64]
PF06996 (T6SS TssG)	0.6690	T6SS component, TssG	
PF14113 (Tae4)	0.6684	Effector protein	
PF11678 (DUF3274)	0.6566	Effector protein domain, tle3	[60]
PF09994 (DUF2235)	0.6481	Effector protein domain, tle1	[67]

Annotation ID	Score	Description	Reference
PF09867 (TagF N)	0.6473	T6SS related	[68]
PF12486 (VasL)	0.6454	T6SS component, TssA	
PF19933 (DUF6396)	0.6288	UNKNOWN	
PF04965 (GPW gp25)	0.6285	T6SS component, TssE	
PF01764 (Lipase 3)	0.6235	Effector protein domain, tseL	
PF20093 (DUF6484)	0.6229	VgrG related	
PF20598 (DUF6795)	0.6190	UNKNOWN	
PF18426 (Tli4 C)	0.6165	Immunity protein Tli4	
PF20232 (T6SS FHA C)	0.6138	T6SS related, FHA	
PF15607 (Ntox44)	0.6013	Effector protein	
PF11319 (VasI)	0.5958	Effector protein	[69]
PF20148 (DUF6531)	0.5955	Effector protein domain	
PF15579 (Imm52)	0.5950	Immunity protein	
PF15604 (Ntox15)	0.5943	Effector protein	[70]
PF17642 (TssD)	0.5942	T6SS component, TssD	
PF08887 (GAD-like)	0.5940	Immunity protein	
PF12059 (DUF3540)	0.5940	UNKNOWN	

The score is the cosine similarity between two words, which represents a metric used to measure how similar two vectors are in an inner product space.

The Word2Vec model can also be connected with the AnnoTree and AnnoView applications to further test the relationships between two functional domains. Both applications are used for co-occurrence analysis: AnnoTree focuses on genome-wide co-occurrence analysis, while AnnoView focuses on window-based co-occurrence analysis. The hypergeometric test can be applied to check the potential relationships between two target annotation domains that share similar genomic contexts.

The hypergeometric test is a statistical method used to determine if the overlap between two sets is significant within a given dataset. In the context of a biological database, where the two sets represent the distributions of two different functional domains, the hypergeometric test evaluates whether the observed overlap between these domains is greater than would be expected by chance. This test calculates the probability of obtaining the observed number of overlapping items (or more extreme) under the null hypothesis that the two sets are independent. The test uses the hypergeometric distribution, which is defined by the formula:

$$P(X = k) = \frac{\binom{K}{k} \binom{N-K}{n-k}}{\binom{N}{n}}$$

where N is the total number of items in the dataset, K is the number of items in set 1, n is the number of items in set 2, and k is the number of overlapping items.

For example, consider a dataset with 1000 items that have one or more labels, where 200 items (set 1) have label A and 150 items (set 2) have label B. If 50 items have both labels, which also represents the overlap of the two sets, we can use the hypergeometric test to determine if this overlap is significant. Here, $N = 1000$, $K = 200$, $n = 150$, and $k = 50$.

In the case of AnnoTree, the genomes that contain both functional domains are considered the overlap of the two sets. In the case of AnnoView, the windows within the genomes that contain both functional domains are considered the overlap of the two sets. Thus, AnnoTree can be used for the initial hypergeometric test within a short time, followed by the hypergeometric test in AnnoView for further verification, which takes longer than AnnoTree.

By substituting these values into the formula, we can calculate the p-value, which indicates the significance of the overlap:

$$P(X \geq 50) = \sum_{i=50}^{\min(K,n)} \frac{\binom{K}{i} \binom{N-K}{n-i}}{\binom{N}{n}}$$

A small p-value suggests that the overlap is unlikely to have occurred by chance, implying a significant association between the two functional domains. This test is particularly useful for identifying meaningful relationships in genomic data, where understanding the co-occurrence of functional domains can provide insights into protein function and interactions.

Therefore, given a pair of functional domains that share a similar genomic context, a small p-value from the hypergeometric test indicates that the two functional domains phylogenetically co-occur. Additionally, if a DUF is co-occurring with another functional domain, it indicates that there is a potential interaction between those two domains, and if a DUF is anti-co-occurring with another functional domain but shares a similar genomic context, it suggests that the roles of these two domains are similar in the cluster. This allows for functional prediction for uncharacterized proteins.

2.4.3 Evaluation

In this thesis, KO IDs and their corresponding KEGG pathways will be used to assess the performance of the model. The underlying assumption is that, in the high-dimensional embedding space, closer embeddings represent genes that share a similar genomic context and may belong to conserved gene clusters. This suggests potential interaction between the genes, indicating they might be part of the same pathway and thus share the same KEGG pathway ID. An example is the Type VI Secretion System (T6SS), where the necessary components, such as K11903

(Hcp) and K11904 (VgrG), are both assigned to the KEGG pathway: Bacterial secretion system [PATH:ko03070]. These components frequently co-occur with each other.

The evaluation pipeline consists of the following steps:

1. Find the word dictionary of the model, which consists of 51,671 unique representations of genes. For example, the gene ID 'NZ_JYKS01000015.1_309' has the annotations PF05943 (VipB) and PF18945 (VipB_2). This is considered one word in the dictionary in the format 'PF05943 (VipB), PF18945 (VipB_2)'.
2. For each word, identify its corresponding KEGG ID across the database. Some words may correspond to multiple KEGG IDs, and only the KEGG ID with the highest occurrence is used as the label for the current word. The KEGG pathway of this KEGG ID is then used as the ground truth. Words without a corresponding KEGG ID are discarded, reducing the size of the dictionary to 39,167 words.
3. Once each word in the dictionary has its KEGG ID as ground truth, the predicted label can be inferred using the K-nearest neighbors (KNN) algorithm. The KEGG pathway ID of the given word is determined by voting based on the top similar words based on their genomic context. For example, if the most frequent KEGG pathway ID among the top 3 similar words is PATH:ko03070, then the predicted pathway ID is PATH:ko03070. Additionally, some KEGG IDs correspond to multiple KEGG pathway IDs. For instance, K18138 belongs to both beta-Lactam resistance [PATH:ko01501] and penicillin and cephalosporin biosynthesis [PATH:ko01502]. Therefore, both the ground truth and prediction are considered sets of KEGG pathways rather than single terms.

The pipeline will be applied to every single word in the dictionary. The precision, recall, and F1-score will be used to calculate the performance metrics.

Precision is the proportion of correctly predicted items (true positives) out of all items predicted by the model.

$$\text{Precision} = \frac{|\text{True Positives}|}{|\text{Predicted Set}|}$$

True Positives (TP): The items that are both in the predicted set and in the ground truth set. For example, if the predicted set is {A, B, C} and the ground truth is {A, B}, the true positives are {A, B}. **Predicted Set:** The total number of items in the set predicted by the model. In this case, it's {A, B, C}.

Recall is the proportion of correctly predicted items (true positives) out of all the items in the ground truth.

$$\text{Recall} = \frac{|\text{True Positives}|}{|\text{Ground Truth Set}|}$$

Ground Truth Set: The number of items in the actual ground truth set. In this case, it's $\{A, B\}$.

So, for the same example: - True Positives: $\{A, B\}$ - Recall:

$$\frac{|\{A, B\}|}{|\{A, B\}|} = \frac{2}{2} = 1$$

The F1 score is calculated as the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

In this case:

$$F1 = 2 \times \frac{0.67 \times 1}{0.67 + 1} \approx 0.8$$

For the model in this thesis, the final average precision is 0.402, recall is 0.468, and F1-score is 0.43 across all classes. However, these metrics are affected by class imbalance, as the occurrence of some classes is higher than others. The figures below (Fig 2.16) show the precision, recall, and F1-score for individual classes. For example, the KEGG pathway, Two-component system [PATH:ko02020], has the highest occurrence, with a count of 3,592, a precision of 0.63, a recall of 0.80, and an F1-score of 0.71. For the Bacterial secretion system [PATH:ko03070]', the precision is 0.60, recall is 0.71, and F1-score is 0.56, with a count of 247. In addition, different classes exhibit different performance levels. After filtering irrelevant KEGG pathways, such as 'Autophagy - animal [PATH:ko04140]', there are 177 KEGG pathway classes remaining. The metrics demonstrate that the performance is much better than random, where the F1-score in a uniform distribution would be approximately $\frac{1}{177} = 0.0056$.

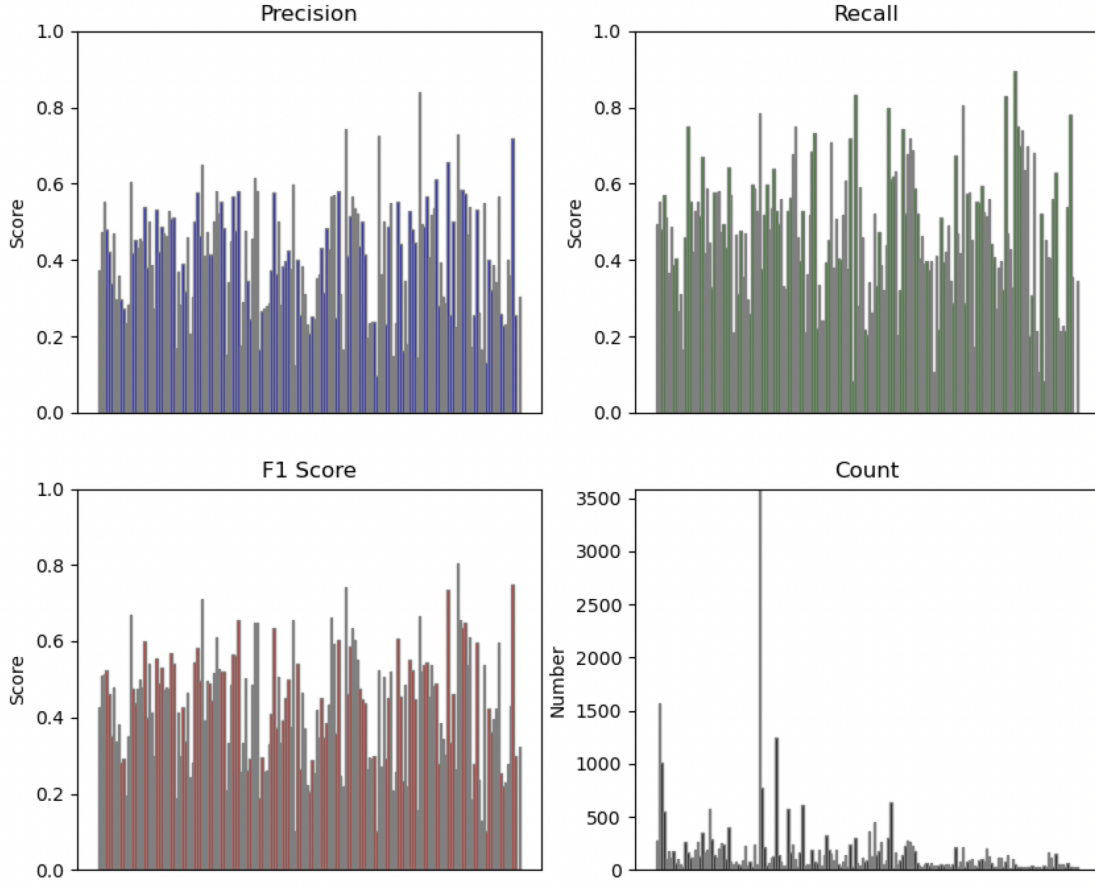


Figure 2.16: Metrics for the Word2Vec model are shown in the figure, with the x-axis in all figures representing KEGG pathway classes, arranged in the same order across all figures.

Since the dataset is constrained by class imbalance across different KEGG pathways, the weighted precision was also calculated using the following formula:

$$\text{Weighted Precision} = \frac{\sum_{i=1}^n w_i \cdot P_i}{\sum_{i=1}^n w_i},$$

where w_i represents the weight (e.g., the number of samples) for class i , and P_i is the precision for class i .

The weighted precision of this method was 0.48, and the weighted sensitivity was 0.56. Al-

though low, these accuracies are significantly higher than those of a random model. The average precision for pathway prediction was 192-fold higher than that obtained using a random model.

In addition, some pathways could be predicted with much higher accuracy than others (Table 2.2). For example, two-component systems [PATH:ko02020], nitrogen metabolism [PATH:ko00910], flagellar assembly [PATH:ko02040], cobalamin transport and metabolism [PATH:ko04980], and carbon fixation pathways in prokaryotes [PATH:ko00720] all had precision and sensitivity ranging from 60–90%. In contrast, other functions, such as pancreatic secretion [PATH:ko04972] and lysosome [PATH:ko04142], had values below 40%. Further analysis revealed that low-performing pathways were highly associated with eukaryotic organisms, which are not well represented in our AnnoTree database and not in GTDB at ll.

Table 2.2: Accuracy in predicting different KEGG pathways in prokaryotes

Class	Precision	Recall	Count
RNA polymerase	0.77	0.89	80
Phosphotransferase system	0.80	0.85	55
Flagellar assembly	0.68	0.77	216
Citrate cycle (TCA cycle)	0.68	0.75	195
Oxidative phosphorylation	0.65	0.78	302
Two-component system	0.63	0.80	3592
Nitrogen metabolism	0.68	0.72	161
Carbon fixation in prokaryotes	0.65	0.65	242
Cobalamin transport and metabolism	0.62	0.69	26
Bacterial secretion system	0.60	0.71	247

2.4.4 Future Work

The Word2Vec model is a machine learning-based language model known for its effectiveness in word embedding and its interpretability. However, challenges still remain. There is currently no standard metric to test the semantic and syntactic accuracy of the target protein domains. In the original paper, the accuracy of the Word2Vec models was tested using various tasks designed to evaluate the quality of word embeddings. These tasks included syntactic and semantic word relationship tests, as well as analogy tasks. The authors used word pairs to test the model’s ability to capture both syntactic and semantic relationships. The pairs were labeled manually, and the accuracy was calculated by counting the number of correct answers.

However, in the case of protein functional domains, manually creating such a training dataset to answer these analogy questions and measure accuracy is challenging. While precision and

recall methods can demonstrate that the model outperforms a random uniform distribution, the evaluation method still requires improvement.

Firstly, the coverage of the annotation affects the evaluation, as the neighbors of the target word may be unannotated, which leads to the loss of semantic structure in the gene cluster. Secondly, a word may have multiple meanings, and its function is highly dependent on its genomic context. For example, the ground truth pathway for a given KEGG ID might be both A and B, but since the ID is found in different gene clusters, the model may predict only A or B, rather than both, which leads to an underestimation of the model's performance.

In the future, one possible improvement is to use clustering algorithms to group all unannotated genes and assign customized labels to them. This would allow unannotated genes to be classified into more specific categories instead of using the generic GAP prefix. This approach could better capture the semantic meaning of conserved gene clusters and provide deeper insights into the genomic language model.

Chapter 3

Case Analysis Using AnnoTools Pipeline

3.1 Background

Function prediction for uncharacterized proteins involves inferring the biological roles and functions of proteins whose functions have not yet been experimentally determined. This task is crucial for understanding the roles these proteins play in various biological processes and pathways. Traditional methods like BLAST often rely on sequence-based similarity searches against known protein databases, but these approaches may fail for novel proteins with no close homologs. To address this limitation, the Word2Vec-based prediction model can predict functional categories for a substantial number of uncharacterized genes by contextualizing them within their genomic neighborhoods.

In this thesis, the Word2Vec model is applied to several case studies, demonstrating its ability to predict novel Type VI secretion system (T6SS) effectors that are currently classified as domains of unknown function (DUFs). The type VI secretion system (T6SS) is an excellent case study for gene function prediction using phylogenomic machine learning due to its complexity, diversity, and incomplete characterization across bacterial species. T6SS is a versatile nanomachine used by many Gram-negative bacteria to inject effector proteins into target cells, facilitating interbacterial competition, host interactions, and niche colonization. The system is composed of a conserved structural core, but its functionality is largely dictated by the diversity of effector proteins it delivers. These effectors, which can target cell walls, membranes, or intracellular components, vary widely among bacterial species, reflecting adaptation to specific ecological niches and competitive pressures [57, 58]. Despite its ubiquity and ecological significance, the T6SS remains poorly understood in many bacterial taxa. While some core components have been well characterized, the diversity of accessory genes and effectors, as well as

the evolutionary trajectories that give rise to this diversity, remain an open field of study. This lack of comprehensive annotation makes T6SS an ideal target for phylogenomic approaches, as machine learning can leverage genomic and phylogenetic patterns to predict unknown gene functions and uncover novel effector proteins. Some studies have also demonstrated the effectiveness of this combinational approach [71]. By focusing on T6SS, our study not only addresses a system of high biological relevance but also highlights the power of computational tools in advancing our understanding of complex and variable systems in microbial genomes. Additionally, the updated applications, AnnoTree and AnnoView, are utilized to interpret the prediction results of the Word2Vec model. AnnoTree provides taxonomy distribution, while AnnoView enables further verification through gene neighborhood visualization.

3.2 Language Model Prediction and Analysis

After training, given an annotated gene with one or more functional domains, the model can identify other annotated genes that share a similar genomic context. Consider the protein domain for the VasX toxin mentioned previously. Table 3.1 presents the results from the trained model when using the domain (PF20249) as the query word. To focus on potentially novel predictions, the results are filtered, and only the top-scoring proteins with single Domains of Unknown Function (DUFs) are listed. Twelve out of the top 15 (80%) identified DUFs have been previously associated with the Type VI Secretion System (T6SS), with supporting literature shown in the table. The remaining three DUFs—PF19940 (DUF6402), PF19933 (DUF6396), and PF20598 (DUF6795)—have not been previously associated with T6SS to our knowledge, and therefore represent potentially novel T6SS protein predictions. The scores are calculated using cosine similarity.

Further analysis of these Pfam IDs is needed to determine whether additional evidence supports a T6SS-related function. A hypergeometric test can be used for this analysis. One of the necessary proteins for T6SS is the spike protein VgrG, with the Pfam domain ID PF04717. The hypergeometric test can assess whether both Pfam IDs co-occur, with significance measured by a p-value. In this case, if the p-value is small, it suggests that the target DUF ID frequently co-occurs with PF04717, indicating that the DUF may be involved in T6SS and perform a T6SS-related function.

Table 3.1: Top-scoring DUFs with semantic similarity to the T6SS protein VasX (PF20249)

Annotation ID	Score	Description	Ref
PF20455 (DUF6708)	0.9123	Immunity protein	[59]
PF10106 (DUF2345)	0.8395	VgrG related	[60]
PF11745 (DUF3304)	0.8239	Immunity protein tliv1	[61]
PF13503 (DUF4123)	0.8121	Chaperone proteins	[62]
PF11153 (DUF2931)	0.7067	Immunity protein tli1, tli2	[63]
PF08929 (DUF1911)	0.7025	Immunity protein PoNi	[64]
PF11876 (DUF3396)	0.6992	Immunity protein TsiV1–4	[65]
PF09937 (DUF2169)	0.6791	Adaptor protein	[66]
PF19940 (DUF6402)	0.6791	UNKNOWN	
PF08928 (DUF1910)	0.6746	Immunity protein PoNi	[64]
PF11678 (DUF3274)	0.6566	Effector protein domain, tle3	[60]
PF09994 (DUF2235)	0.6481	Effector protein domain, tle1	[67]
PF19933 (DUF6396)	0.6288	UNKNOWN	
PF20093 (DUF6484)	0.6229	VgrG related	[68]
PF20598 (DUF6795)	0.6190	UNKNOWN	

3.2.1 AnnoTree-Based Analysis

The AnnoTree application utilizes the updated database to provide detailed statistics on the distribution of genomes, making it a valuable tool for supporting hypergeometric tests. The general pipeline for AnnoTree-based analysis includes the following steps:

1. Run the combination search function from AnnoView and visualize the distribution of hits on AnnoTree. This restricts the search to a fixed window length (e.g., 30 kilobases) and ensures that the IDs are close to each other, forming a conserved gene cluster.
2. Perform a hypergeometric test to determine whether the co-occurrence is statistically significant [72]. The p-value is calculated for reference: a small p-value indicates a strong co-occurrence relationship, while a larger p-value suggests further exploration of the query distribution may be necessary.
3. The default setting of AnnoTree focuses on the rank of bacteria. If the p-value is not significant, users can adjust the taxonomy based on the distribution of the query to a lower taxonomic rank. Co-occurrence may only be significant within specific taxonomies. For instance, the co-occurrence may be significant within *Pseudomonadaceae* (family) but not *Pseudomonadota* (phylum).

In the case of the Type VI Secretion System (T6SS), the combination search was limited to one DUF of interest and the Pfam ID PF04717 (phage base V). If the taxonomy is not restricted (i.e., the universal set is the entire bacterial database), there are 80,789 genomes, and 15,437 of these genomes contain PF04717 (phage base V).

One of the previously mentioned DUFs is PF20598 (DUF6795). By conducting a combination search for PF20598 and visualizing the results in AnnoTree, the analysis reveals that 454 genomes contain PF20598 (DUF6795), and 98 windows of 30 kb length in these genomes contain both PF20598 (DUF6795) and PF04717 (phage base V) (Fig 3.1). Applying the hypergeometric test yields a p-value of 0.1, suggesting that the overlap between the two sets is not statistically significant and is likely due to random chance. This indicates that there is no strong evidence for a co-occurrence association between PF20598 (DUF6795) and PF04717 (phage base V).

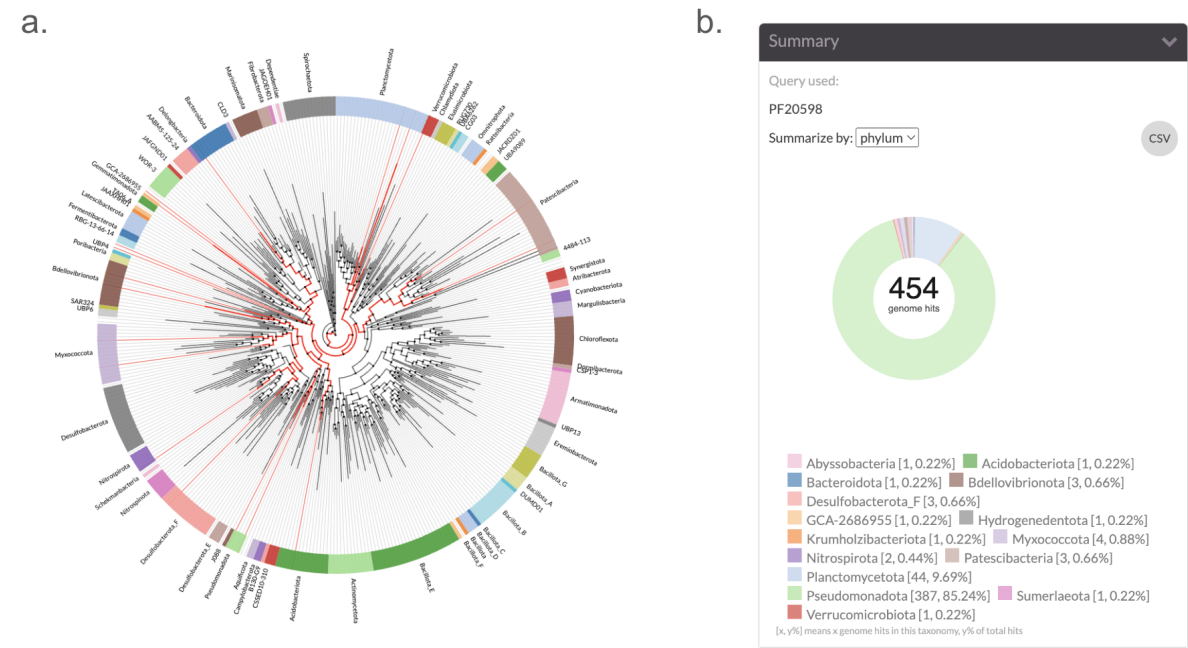


Figure 3.1: a. AnnoTree visualization of domain ID PF20598 (DUF6795), with the genomes containing the ID highlighted in red. b. Detailed description of the distribution.

The analyses for the three DUFs are presented in the following table:

Table 3.2: Hypergeometric Test for Co-Occurrence Significance across the Bacterial Database

Annotation ID	# of Genomes Containing Target ID	# of Genomes Containing the Combination	P-value	T6SS Found?
PF20598 (DUF6795)	454	98	0.1	Yes
PF19940 (DUF6402)	350	100	1.15×10^{-5}	Yes
PF19933 (DUF6396)	386	251	6.31×10^{-87}	Yes

It is important to note that the p-value for PF20598 (DUF6795) is not significant across the bacterial database (p-value > 0.05). This is because the combination may be taxonomy-dependent, and only a subset of genes annotated as PF20598 (DUF6795) function within the T6SS. In this case, users can further investigate the phylogeny of the ID to find the taxonomic level at which the combination occurs within the T6SS.

For example, at the order level of *Pseudomonadales*, the total number of genomes is 27,546. The number of genomes that contain PF04717 is 1,512, the number of genomes that contain PF20598 is 128, and the number of genomes that contain both PF20598 and PF04717 within the specified window is 51. This gives a p-value of 6.41×10^{-31} , indicating a strong co-occurrence association between PF20598 (DUF6795) and PF04717 (Phage base V) within *Pseudomonadales* (Fig 3.2).

From the table above, it is evident that there is a statistical significance of co-occurrence between the domain IDs and the spike protein VgrG.

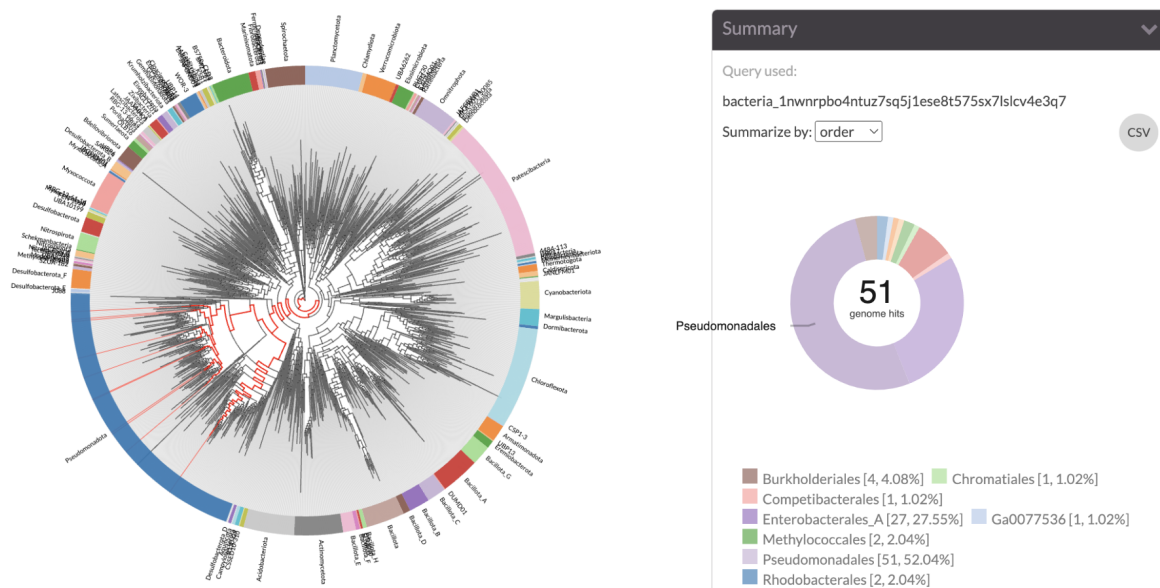


Figure 3.2: AnnoTree visualization with the taxonomy limited to the order level.

3.2.2 AnnoView-Based Analysis

The AnnoView application is used for gene neighborhood visualization, supporting the interpretation of predictions based on gene coordinates. It provides detailed information and direct evidence for the predicted functions.

The figures (Fig 3.3) display the gene neighborhood visualization around the target domain ID, where the target domain ID is positioned in the center and highlighted in bold black. Surrounding genes are bolded in red, and genes annotated with the same domain IDs are color-coded consistently. On the left, the taxonomy of each genome is shown, and the visualization spans a 30 kb range around the target domain ID.

Thus, by using AnnoView, the prediction can be visually verified, providing evidence that the domain is related to the T6SS.

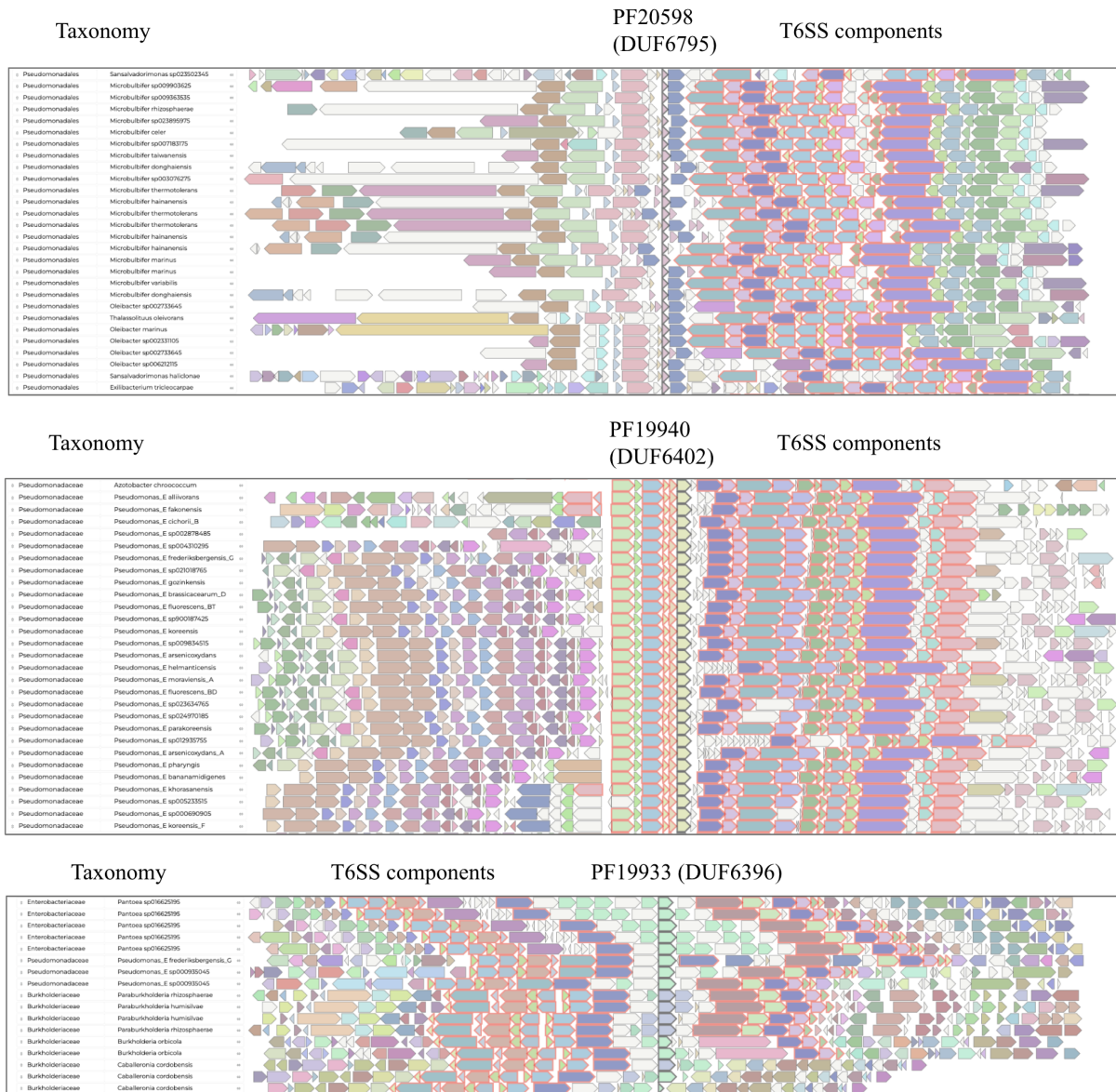


Figure 3.3: Visualization of case studies using AnnoView for different protein families. The colors remain consistent across all subplots if the domains within the proteins are the same. (a) AnnoView 2.0 visualization for PF20598(DUF6795). (b) AnnoView 2.0 visualization for PF19940(DUF6402). (c) AnnoView 2.0 visualization for PF19933(DUF6396).

3.3 Follow-Up Analysis Using Foldseek

Foldseek [43] is a fast and efficient search tool used to compare protein structures. It allows users to identify similar protein structures in large databases by aligning 3D models. When combined with AlphaFold predictions and other structure databases like PDB, Foldseek can compare the structure of an unknown protein with those of known functions, offering potential functional annotations based on structural similarities. This structural comparison can reveal potential functions and further support model predictions.

For genes located near T6SS components, their UniRef100 IDs are identified and collected. These IDs are then mapped to AlphaFold predictions where the structure with the highest number of votes is selected. This ensures that the majority of the structures align with the same domain annotations. The AlphaFold predictions [42] are subsequently used as input to search against other structure databases using Foldseek. The results for three DUFs are listed in the following table:

Table 3.3: Foldseek Results for Three DUFs

Annotation ID	Example UniProt ID Mapping	Taxonomy in GTDB	Mapping E-Value	Mapping Description
PF20598 (DUF6795)	S2KPG5	Microbulbifer taiwanensis	1.93e-3	TransThyretin-Related Family Domain
PF19940 (DUF6402)	A0A1Q9XU50	Pseudomonas_E sp009764645		Uncharacterized Protein
PF19933 (DUF6396)	A0A423KER3	Pseudomonas_E frederiksbergensis.G	1.18e-7	Crystal Structure of Immunity Protein Pa5087 from <i>Pseudomonas aeruginosa</i>

Foldseek results provide additional evidence for the functional roles of these domain IDs in T6SS. For PF20598 (DUF6795), most Foldseek results indicate that the structure is related to the TransThyretin-Related (TTR) family domain. TTR is an immunoglobulin-like domain known to form oligomeric structures in which these domains interact with each other or with other partners [73]. Based on the structural similarity identified through Foldseek, the TTR domain in PF20598 may function as an adapter in the T6SS. PAAR-associated TTR domains could potentially facilitate the decoration of the VgrG tip with effectors that display TTR domains or serve to bind the spike to other TTR domain-containing proteins [74, 75]. This suggests that PF20598 (DUF6795) plays a role in T6SS by mediating protein-protein interactions within the secretion system (Fig 3.4). AnnoView visualization also shows that this domain is located downstream of the PAAR motif, indirectly supporting this hypothesis.

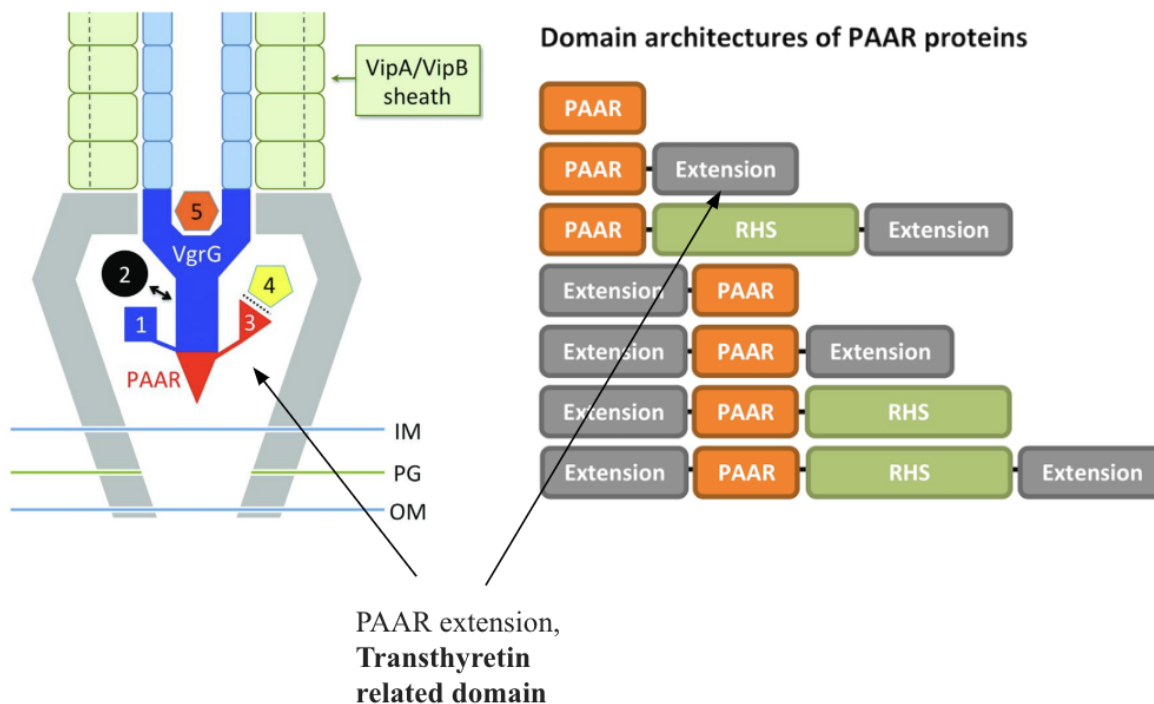


Figure 3.4: The TransThyretin-Related (TTR) domain functions as an extension with the PAAR protein. Figure adapted from ref [74]

For PF19940 (DUF6402), based on genomic context, DUF6396 was found to occur nearby T6SS gene clusters, consistent with its gLM-based association with T6SS genes (Fig 3.5a). AlphaFold [42] and Foldseek [43] were utilized to predict the structure of DUF6396 and identify similar structures of known function in the Protein Data Bank. The AlphaFold-predicted structure of DUF6396 (Fig 3.5b) was found to be significantly similar to a crystal structure of the immunity protein Pa5087 from *Pseudomonas aeruginosa*, which is a known T6SS immunity protein [76]. Together, the gene neighborhood information, predicted structure, and Word2vec model all points to DUF6396 as a novel T6SS protein with a likely function as an immunity protein.

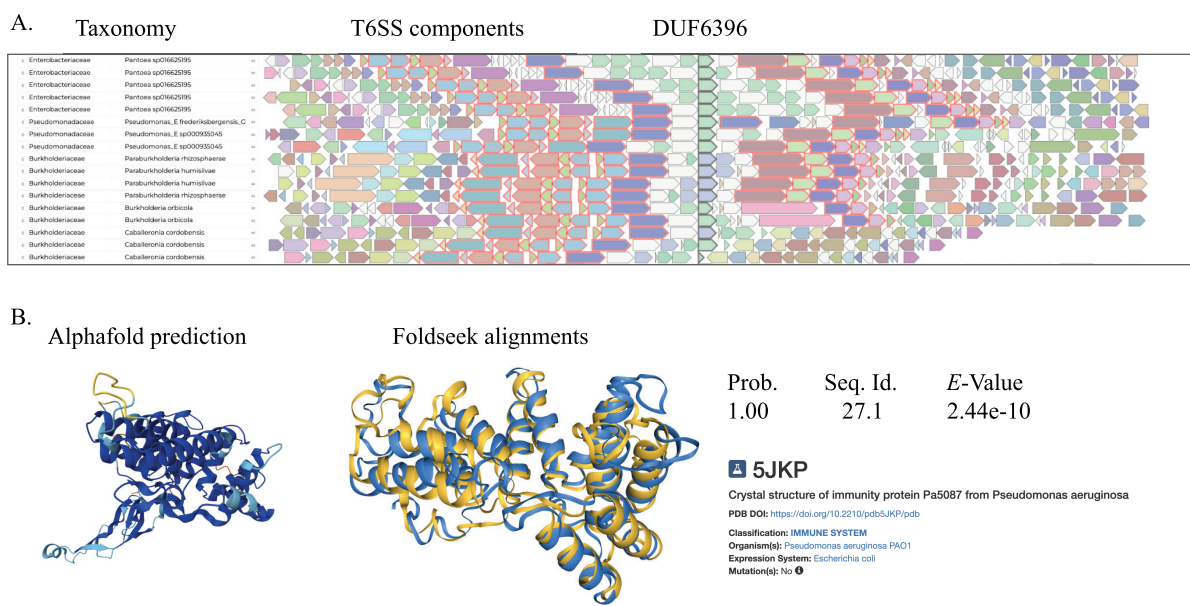


Figure 3.5: (A) AnnoView visualization for the gene neighborhood surrounding PF19933 (DUF6396) highlights this domain in black, while the necessary components of the T6SS are highlighted in red. (B) AlphaFold prediction and Foldseek search result.

Chapter 4

Conclusion and Future Directions

4.1 Contribution

In conclusion, this thesis presents a comprehensive pipeline, AnnoTools, which integrates data annotation, visualization, and machine learning for comparative genomic analysis.

In the data annotation phase, Kofamscan and DIAMOND were employed to assign identifiers from 14 different protein families to genomic data from the Genome Taxonomy Database (GTDB). The data were then post-processed and stored remotely to enable fast and efficient data retrieval.

For the visualization component, the tools AnnoTree and AnnoView, previously developed by our lab, were updated with new databases and enhanced functionalities. These updates provide more comprehensive annotations and intuitive data exploration. A notable feature is the combination search, which allows users to search the entire database for specific combinations of annotation IDs within a fixed-length window. This is particularly useful for identifying conserved gene clusters.

In the machine learning part, the Word2Vec model was trained using the updated database as a genomic corpus. This model can predict the potential function of a given domain ID based on its genomic context, offering new insights into uncharacterized domains.

By leveraging large-scale protein function annotations, enhancing visualization tools, and applying the Word2Vec language model, this pipeline significantly improves the coverage and functionality of AnnoTree and AnnoView. These enhancements enable more efficient data exploration and interpretation, offering novel insights into protein functions and their genomic contexts.

4.2 AnnoTools Applications and Language Model

The Word2Vec model demonstrates that the updated database can be effectively trained as a genomic corpus, where each word represents a domain ID, and each contig functions as a sentence. Through this approach, domain IDs are encoded as high-dimensional vectors by the model, with domain IDs sharing similar gene neighborhoods clustering together in hypergeometric space. The case study on the Type VI Secretion System (T6SS) highlights the pipeline’s potential in identifying the roles of Domains of Unknown Function (DUFs), serving as a complementary method to traditional sequence similarity-based searches. The results can be verified using both AnnoTree and AnnoView, enabling the exploration of distribution patterns through hypergeometric tests or visualization of gene neighborhoods to examine clusters near the target domain ID. Additionally, the model can be connected to structure-based tools such as Foldseek, providing further evidence based on predicted protein structures.

4.3 Limitations

The “AnnoTools” pipeline represents a significant advancement in genome annotation and visualization; however, several limitations must be considered to contextualize its capabilities and identify areas for improvement. A primary limitation lies in its dependency on external databases such as KEGG, InterPro, and their associated repositories. The quality and completeness of functional annotations are inherently tied to the coverage and accuracy of these resources. Consequently, certain genes, particularly those from less-studied or novel genomes, may remain unannotated or inaccurately characterized due to gaps or inconsistencies in the underlying databases. Moreover, updates or modifications to these databases, such as changes to UniRef100 or InterPro, could lead to discrepancies in annotations or missing data, potentially impacting the reproducibility of the pipeline’s results. A potential solution to mitigate this issue is to establish a standardized update schedule for these databases, ensuring greater consistency across external resources and enhancing annotation reliability.

Another notable limitation lies in the application of the Word2Vec language model within the pipeline. While this approach provides novel insights into functional prediction, its reliance on genomic context introduces potential biases, particularly for protein domains that lack annotations. Such biases can result in noisy data, potentially misleading the model by establishing connections between unrelated proteins. This issue is especially problematic for rare protein domains within the dataset, which may lead to incorrect embeddings. Additionally, Word2Vec is a relatively traditional language model and is considered simplistic compared to recent advancements in machine learning. Incorporating more sophisticated neural network-based models

capable of handling high-dimensional contextual data may yield improved results in the future. Furthermore, the functional predictions, especially for domains of unknown function (DUFs), remain computationally derived and require experimental validation to confirm their biological roles and interactions.

4.4 Future Directions

In the future, the pipeline can be improved in several ways:

1. **Annotation Pipeline Enhancements:** As the Genome Taxonomy Database (GTDB) updates annually with approximately a 30% increase in size, it is essential to update the databases used by AnnoTree and AnnoView accordingly. Without regular updates, the data may become outdated, leading to incomplete or less accurate annotations. The increasing database size may also slow down the applications due to the larger datasets being processed. To address this, the development of secondary data indices and the use of additional pre-computed tables on the backend could support faster data retrieval. Furthermore, annotation tools and their reference databases, such as UniRef100 and the mapping between UniRef100 and InterPro, should be updated regularly to improve the coverage and accuracy of annotations. In addition, additional information will be incorporated into the database, including the coordinates of domains and motifs within sequences, to enhance the transfer of functional annotations from UniProt to proteins in those genomes. This is achieved by collecting metadata from the InterPro database and building cross-reference mappings within a MySQL database.
2. **Machine Learning Model Improvement:** The current Word2Vec model is trained solely on annotated genomic corpora, while unannotated data are classified by their length, which does not provide an accurate representation of these unannotated sequences and may introduce noise into the training corpus. To improve the model, one possible enhancement is to cluster the unannotated data and assign customized identifiers. This would ensure that the genomic corpus is more comprehensive and better reflects the diversity of gene sequences, potentially improving the model's overall performance.

References

- [1] Pedro Queirós, Polina Novikova, Paul Wilmes, and Patrick May. Unification of functional annotation descriptions using text mining. *Biological Chemistry*, 402(8):983–990, 2021.
- [2] Lincoln Stein. Genome annotation: from sequence to biology. *Nature Reviews Genetics*, 2:493–503, 2001.
- [3] C. Yu, N. Zavaljevski, V. Desai, and et al. The development of pipa: an integrated and automated pipeline for genome-wide protein function annotation. *BMC Bioinformatics*, 9:52, 2008.
- [4] C.A. Ruiz-Perez, R.E. Conrad, and K.T. Konstantinidis. Microbeannotator: a user-friendly, comprehensive functional annotation pipeline for microbial genomes. *BMC Bioinformatics*, 22:11, 2021.
- [5] J.O. Korbelt, L.J. Jensen, C. von Mering, and P. Bork. Analysis of genomic context: prediction of functional associations from conserved bidirectionally transcribed gene pairs. *Nature Biotechnology*, 22(7):911–917, 2004.
- [6] E.W. Sayers, E.E. Bolton, J.R. Brister, K. Canese, J. Chan, D.C. Comeau, R. Connor, K. Funk, C. Kelly, S. Kim, T. Madej, A. Marchler-Bauer, C. Lanczycki, S. Lathrop, Z. Lu, F. Thibaud-Nissen, T. Murphy, L. Phan, Y. Skripchenko, T. Tse, J. Wang, R. Williams, B.W. Trawick, K.D. Pruitt, and S.T. Sherry. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 50(D1):D20–D26, 2022.
- [7] N.A. O’Leary, M.W. Wright, J.R. Brister, S. Ciuffo, D. Haddad, R. McVeigh, B. Rajput, B. Robbertse, B. Smith-White, D. Ako-Adjei, A. Astashyn, A. Badretdin, Y. Bao, O. Blinkova, V. Brover, V. Chetvernin, J. Choi, E. Cox, O. Ermolaeva, C.M. Farrell, T. Goldfarb, T. Gupta, D. Haft, E. Hatcher, W. Hlavina, V.S. Joardar, V.K. Kodali, W. Li,

- D. Maglott, P. Masterson, K.M. McGarvey, M.R. Murphy, K. O'Neill, S. Pujar, S.H. Rangwala, D. Rausch, L.D. Riddick, C. Schoch, A. Shkeda, S.S. Storz, H. Sun, F. Thibaud-Nissen, I. Tolstoy, R.E. Tully, A.R. Vatsan, C. Wallin, D. Webb, W. Wu, M.J. Landrum, A. Kimchi, T. Tatusova, M. DiCuccio, P. Kitts, T.D. Murphy, and K.D. Pruitt. Reference sequence (refseq) database at ncbi: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research*, 44(D1):D733–D745, 2016.
- [8] C.L. Schoch, S. Ciufo, M. Domrachev, C.L. Hutton, S. Kannan, R. Khovanskaya, D. Leipe, R. McVeigh, K. O'Neill, B. Robertse, S. Sharma, V. Soussov, J.P. Sullivan, L. Sun, S. Turner, and I. Karsch-Mizrachi. Ncbi taxonomy: a comprehensive update on curation, resources and tools. *Database*, 2020:baaa062, 2020.
- [9] D.H. Parks, M. Chuvochina, D.W. Waite, C. Rinke, A. Skarshewski, P.-A. Chaumeil, and P. Hugenholtz. A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. *Nature Biotechnology*, 36(10):996–1004, 2018.
- [10] D.H. Parks, M. Imelfort, C.T. Skennerton, P. Hugenholtz, and G.W. Tyson. Checkm: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Research*, 25:1043–1055, 2015.
- [11] D. Hyatt, G.L. Chen, P.F. LoCascio, and et al. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, 11:119, 2010.
- [12] W.R. Pearson. An introduction to sequence similarity ('homology') searching. *Current Protocols in Bioinformatics*, 2013 Jun; Chapter 3:3.1.1-3.1.8, 2013.
- [13] D. Lee, O. Redfern, and C. Orengo. Predicting protein function from sequence and structure. *Nature Reviews Molecular Cell Biology*, 8:995–1005, 2007.
- [14] C. Camacho, G. Coulouris, V. Avagyan, and et al. Blast+: architecture and applications. *BMC Bioinformatics*, 10:421, 2009.
- [15] B. Buchfink, K. Reuter, and H.G. Drost. Sensitive protein alignments at tree-of-life scale using diamond. *Nature Methods*, 18:366–368, 2021.
- [16] B.E. Suzek, Y. Wang, H. Huang, P.B. McGarvey, and C.H. Wu; UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.
- [17] S.R. Eddy. Hidden markov models. *Current Opinion in Structural Biology*, 6(3):361–365, 1996.

- [18] T. Aramaki, R. Blanc-Mathieu, H. Endo, K. Ohkubo, M. Kanehisa, S. Goto, and H. Ogata. Kofamkoala: Kegg ortholog assignment based on profile hmm and adaptive score threshold. *Bioinformatics*, 36(7):2251–2252, April 2020.
- [19] M. Kanehisa and S. Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.
- [20] P. Jones, D. Binns, H.-Y. Chang, M. Fraser, W. Li, C. McAnulla, H. McWilliam, J. Maslen, A. Mitchell, G. Nuka, S. Pesseat, A.F. Quinn, A. Sangrador-Vegas, M. Scheremetjew, S.-Y. Yong, R. Lopez, and S. Hunter. Interproscan 5: genome-scale protein function classification. *Bioinformatics*, 30(9):1236–1240, 2014.
- [21] M. Blum, H.-Y. Chang, S. Chuguransky, T. Grego, S. Kandasaamy, A. Mitchell, G. Nuka, T. Paysan-Lafosse, M. Qureshi, S. Raj, L. Richardson, G.A. Salazar, L. Williams, P. Bork, A. Bridge, J. Gough, D.H. Haft, I. Letunic, A. Marchler-Bauer, H. Mi, D.A. Natale, M. Necci, C.A. Orengo, A.P. Pandurangan, C. Rivoire, C.J.A. Sigrist, I. Sillitoe, N. Thanki, P.D. Thomas, S.C.E. Tosatto, C.H. Wu, A. Bateman, and R.D. Finn. The interpro protein families and domains database: 20 years on. *Nucleic Acids Research*, 49(D1):D344–D354, 2021.
- [22] M.A. Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, S. Lewis, B. Marshall, C. Mungall, J. Richter, G.M. Rubin, J.A. Blake, C. Bult, M. Dolan, H. Drabkin, J.T. Eppig, D.P. Hill, L. Ni, M. Ringwald, R. Balakrishnan, J.M. Cherry, K.R. Christie, M.C. Costanzo, S.S. Dwight, S. Engel, D.G. Fisk, J.E. Hirschman, E.L. Hong, R.S. Nash, A. Sethuraman, C.L. Theesfeld, D. Botstein, K. Dolinski, B. Feierbach, T. Bernardini, S. Mundodi, S.Y. Rhee, R. Apweiler, D. Barrell, E. Camon, E. Dummer, V. Lee, R. Chisholm, P. Gaudet, W. Kibbe, R. Kishore, E.M. Schwarz, P. Sternberg, M. Gwinn, L. Hannick, J. Wortman, M. Berriman, V. Wood, N. de la Cruz, P. Tonellato, P. Jaiswal, T. Seigfried, and R. White. The gene ontology (go) database and informatics resource. *Nucleic Acids Research*, 32(Database issue):D258–D261, 2004.
- [23] I. Sillitoe, N. Bordin, N. Dawson, V.P. Waman, P. Ashford, H.M. Scholes, C.S.M. Pang, L. Woodridge, C. Rauer, N. Sen, et al. Cath: increased structural coverage of functional space. *Nucleic Acids Research*, 49:D266–D273, 2021.
- [24] J. Wang, F. Chitsaz, M.K. Derbyshire, N.R. Gonzales, M. Gwadz, S. Lu, G.H. Marchler, J.S. Song, N. Thanki, R.A. Yamashita, et al. The conserved domain database in 2023. *Nucleic Acids Research*, 51(D1):D384–D388, 2023.

- [25] I. Pedruzzi, C. Rivoire, A.H. Auchincloss, E. Coudert, G. Keller, E. de Castro, D. Baratin, B.A. CuChe, L. Bougueleret, S. Poux, et al. Hamap in 2015: updates to the protein family classification and annotation system. *Nucleic Acids Research*, 43:D1064–D1070, 2015.
- [26] H. Mi, D. Ebert, A. Muruganujan, C. Mills, L.-P. Albou, T. Mushayamaha, and P.D. Thomas. Panther version 16: a revised family classification, tree-based classification tool, enhancer regions and extensive api. *Nucleic Acids Research*, 49:D394–D403, 2021.
- [27] R.D. Finn, A. Bateman, J. Clements, P. Coghill, R.Y. Eberhardt, S.R. Eddy, A. Heger, K. Hetherington, L. Holm, J. Mistry, et al. Pfam: the protein families database. *Nucleic Acids Research*, 42:D222–D230, 2014.
- [28] A.N. Nikolskaya, C.N. Arighi, H. Huang, W.C. Barker, and C.H. Wu. Pirsf family classification system for protein functional and evolutionary analysis. *Evolutionary Bioinformatics Online*, 2:197–209, 2007.
- [29] T.K. Attwood, A. Coletta, G. Muirhead, A. Pavlopoulou, P.B. Philippou, I. Popov, C. Romá-Mateo, A. Theodosiou, and A.L. Mitchell. The prints database: a fine-grained protein sequence annotation and analysis resource—its status in 2012. *Database*, 2012:bas019, 2012.
- [30] C.J.A. Sigrist, E. de Castro, L. Cerutti, B.A. CuChe, N. Hulo, A. Bridge, L. Bougueleret, and I. Xenarios. New and continuing developments at prosite. *Nucleic Acids Research*, 41:D344–D347, 2013.
- [31] I. Letunic, S. Khedkar, and P. Bork. Smart: recent updates, new developments and status in 2020. *Nucleic Acids Research*, 49:D458–D460, 2021.
- [32] E. Akiva, S. Brown, D.E. Almonacid, A.E. Barber, A.F. Custer, M.A. Hicks, C.C. Huang, F. Lauck, S.T. Mashiyama, E.C. Meng, et al. The structure-function linkage database. *Nucleic Acids Research*, 42:D521–D530, 2014.
- [33] A.P. Pandurangan, J. Stahlhacke, M.E. Oates, B. Smithers, and J. Gough. The superfamily 2.0 database: a significant proteome update and a new webserver. *Nucleic Acids Research*, 47:D490–D494, 2019.
- [34] W. Li, K.R. O’Neill, D.H. Haft, M. DiCuccio, V. Chetvernin, A. Badretdin, G. Coulouris, F. Chitsaz, M.K. Derbyshire, A.S. Durkin, et al. Refseq: expanding the prokaryotic genome annotation pipeline reach with protein family model curation. *Nucleic Acids Research*, 49(D1):D1020–D1028, 2021.

- [35] K. Mendler, H. Chen, D.H. Parks, B. Lobb, L.A. Hug, and A.C. Doxey. Annotree: visualization and exploration of a functionally annotated microbial tree of life. *Nucleic Acids Research*, 47(9):4442–4448, 2019.
- [36] X. Wei, H. Tan, B. Lobb, W. Zhen, Z. Wu, D.H. Parks, J.D. Neufeld, G. Moreno-Hagelsieb, and A.C. Doxey. Annoview enables large-scale analysis, comparison, and visualization of microbial gene neighborhoods. *Briefings in Bioinformatics*, 25(3):bbae229, 2024.
- [37] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint*, arXiv:1301.3781, 2013.
- [38] D. W. Buchan and D. T. Jones. Learning a functional grammar of protein domains using natural language word embedding techniques. *Proteins: Structure, Function, and Bioinformatics*, 88(4):616–624, 2020.
- [39] D. Miller, A. Stern, and D. Burstein. Deciphering microbial gene function using natural language processing. *Nature Communications*, 13:5731, 2022.
- [40] A. Bateman, P. Coghill, and R.D. Finn. Dufs: families in search of function. *Acta Crystallographica Section F: Structural Biology and Crystallization Communications*, 66(Pt 10):1148–1152, 2010.
- [41] S. El-Gebali, J. Mistry, A. Bateman, S.R. Eddy, A. Luciani, et al. The pfam protein families database in 2019. *Nucleic Acids Research*, 47(D1):D427–D432, 2019.
- [42] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.
- [43] M. Van Kempen, S. S. Kim, C. Tumescheit, M. Mirdita, J. Lee, C. L. Gilchrist, et al. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, 42(2):243–246, 2024.
- [44] A.B. Yoo, M.A. Jette, and M. Grondona. Slurm: Simple linux utility for resource management. In *Proceedings of the 9th International Workshop on Job Scheduling Strategies for Parallel Processing*, pages 44–60, 2003.
- [45] Oracle Corporation. Mysql: The world’s most popular open source database.
- [46] A.R. Quinlan and I.M. Hall. Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842, 2010.

- [47] M.I. Shamos and D. Hoey. Geometric intersection problems. In *17th Annual Symposium on Foundations of Computer Science (SFCS 1976)*, pages 208–215, Houston, TX, USA, 1976.
- [48] H. Tan, G. Moreno-Hagelsieb, and A. C. Doxey. Gene function prediction using an annotree-based genomic language model. In *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 5120–5126, 2024.
- [49] F. Almeida and G. Xexéo. Word embeddings: A survey. *arXiv preprint*, arXiv:1901.09069, 2019.
- [50] Y. Hwang, A.L. Cornman, E.H. Kellogg, et al. Genomic language model predicts protein co-regulation and function. *Nature Communications*, 15:2880, 2024.
- [51] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.
- [52] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, page 500902, 2022.
- [53] E. Haggård-Liungquist, E. Jacobsen, S. Rishovd, et al. Bacteriophage p2: genes involved in baseplate assembly. *Virology*, 213(1):109–121, 1995.
- [54] C. Browning, M. M. Shneider, V. D. Bowman, D. Schwarzer, and P. G. Leiman. Phage pierces the host cell membrane with the iron-loaded spike. *Structure*, 20(2):326–339, 2012.
- [55] R. Rehurek and P. Sojka. Gensim–python framework for vector space modelling, 2011.
- [56] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.
- [57] T. G. Dong, B. T. Ho, D. R. Yoder-Himes, and J. J. Mekalanos. Identification of t6ss-dependent effector and immunity proteins by tn-seq in *Vibrio cholerae*. *Proceedings of the National Academy of Sciences*, 110(7):2623–2628, 2013.
- [58] T. Costa, C. Felisberto-Rodrigues, A. Meir, et al. Secretion systems in gram-negative bacteria: structural and mechanistic insights. *Nature Reviews Microbiology*, 13:343–359, 2015.

- [59] T. Garin, C. Brin, A. Préveaux, A. Brault, M. Briand, M. Simonin, and A. Sarniguet. The type vi secretion system of *Stenotrophomonas rhizophila* cfbp13503 limits the transmission of *Xanthomonas campestris* pv. *campestris* 8004 from radish seeds to seedlings. *Molecular Plant Pathology*, 25(1):e13412, 2024.
- [60] T. E. Wood, S. A. Howard, A. Förster, L. M. Nolan, E. Manoli, et al. The *Pseudomonas aeruginosa* t6ss delivers a periplasmic toxin that disrupts bacterial cell morphology. *Cell Reports*, 29(1):187–201.e7, 2019.
- [61] C. V. Crisan, A. T. Chande, K. Williams, V. Raghuram, L. Rishishwar, et al. Analysis of *Vibrio cholerae* genomes identifies new type vi secretion system gene clusters. *Genome Biology*, 20:1–14, 2019.
- [62] X. Liang, R. Moore, M. Wilton, M. J. Wong, L. Lam, and T. G. Dong. Identification of divergent type vi secretion effectors using a conserved chaperone domain. *Proceedings of the National Academy of Sciences*, 112(29):9106–9111, 2015.
- [63] S. Ma, Y. Dong, N. Wang, J. Liu, C. Lu, and Y. Liu. Identification of a new effector-immunity pair of *Aeromonas hydrophila* type vi secretion system. *Veterinary Research*, 51:1–14, 2020.
- [64] B. Jana, C. M. Fridman, E. Bosis, and D. Salomon. A modular effector with a dnase domain and a marker for t6ss substrates. *Nature Communications*, 10(1):3595, 2019.
- [65] J. T. Hespanhol, D. E. Sanchez-Limache, G. G. Nicastro, et al. Antibacterial t6ss effectors with a vrr-nuc domain are structure-specific nucleases. *Elife*, 11:e82437, 2022.
- [66] Y. Liu, Z. Zhang, F. Wang, D. D. Li, and Y. Z. Li. Identification of type vi secretion system toxic effectors using adaptors as markers. *Computational and Structural Biotechnology Journal*, 18:3723–3733, 2020.
- [67] H. Hu, H. Zhang, Z. Gao, et al. Structure of the type vi secretion phospholipase effector tle1 provides insight into its hydrolysis and membrane targeting. *Acta Crystallographica Section D: Biological Crystallography*, 70(Pt 8):2175–2185, 2014.
- [68] E. van Genderen, Y. W. Li, I. Nederlof, and J. P. Abrahams. Lattice filter for processing image data of three-dimensional protein nanocrystals. *Acta Crystallographica Section D: Structural Biology*, 72(Pt 1):34–39, 2016.
- [69] A. González-Magaña, I. Tascón, J. Altuna-Alvarez, M. Queralt-Martín, J. Colautti, et al. Structural and functional insights into the delivery of a bacterial rhs pore-forming toxin to the membrane. *Nature Communications*, 14(1):7808, 2023.

- [70] G. Shalom, J. G. Shaw, and M. S. Thomas. In vivo expression technology identifies a type vi secretion system locus in *Burkholderia pseudomallei* that is induced upon invasion of macrophages. *Microbiology (Reading, England)*, 153(Pt 8):2689–2699, 2007.
- [71] J. Colautti, H. Tan, N. P. Bullen, and et al. A widespread accessory protein family diversifies the effector repertoire of the type vi secretion system spike. *Nature Communications*, 15:10108, 2024.
- [72] B. J. Tremblay, B. Lobb, and A. C. Doxey. Phylocorrelate: inferring bacterial gene-gene functional associations through large-scale phylogenetic profiling. *Bioinformatics (Oxford, England)*, 37(1):17–22, 2021.
- [73] Z. A. Hamburger, M. S. Brown, R. R. Isberg, and P. J. Bjorkman. Crystal structure of invasin: a bacterial integrin-binding protein. *Science*, 286(5438):291–295, 1999.
- [74] M. M. Shneider, S. A. Buth, B. T. Ho, M. Basler, J. J. Mekalanos, and P. G. Leiman. Paar-repeat proteins sharpen and diversify the type vi secretion system spike. *Nature*, 500(7462):350–353, 2013.
- [75] T. E. Wood, S. A. Howard, S. Wettstadt, and A. Filloux. Paar proteins act as the ‘sorting hat’ of the type vi secretion system. *Microbiology (Reading)*, 165(11):1203–1218, 2019.
- [76] X. Y. Yang, Z. Q. Li, Z. She, Z. Geng, J. H. Xu, Z. Q. Gao, and Y. H. Dong. Structural analysis of *Pseudomonas aeruginosa* h3-t6ss immunity proteins. *FEBS Letters*, 590(16):2787–2796, 2016.
- [77] T. Paysan-Lafosse, M. Blum, S. Chuguransky, T. Grego, B. Lázaro Pinto, G. A. Salazar, et al. Interpro in 2022. *Nucleic Acids Research*, 51(D1):D418–D427, 2023.