Application of AI in Bioinfomatics, 2025, pp. 1–5

doi:

FunKG: A knowledge graph framework for metagenomic annotation and protein function prediction

Alejandra Lopez-Velazquez[1]

[1]Biological and Environmental Science and Engineering (BESE), King Abdullah University of Science and Technology (KAUST), SA

Abstract

Functional annotation in metagenomics often relies on isolated sequence similarity searches, which struggle to assign accurate functions to novel or fragmented proteins. In this project, I present a knowledge graph-based framework that leverages contextual information—such as taxonomic origin, genomic neighborhood, and hierarchical ontology structure—to predict protein molecular functions. By integrating metagenomic assemblies into a structured graph and evaluating five knowledge graph embedding models on link prediction tasks, I demonstrate the potential utility of context-aware inference. This framework establishes a foundation for biologically meaningful annotations and future improvements.

Key Messages

- Constructed a metagenomic knowledge graph integrating taxonomic, genomic, and functional data
- Benchmarked five KG embedding models for protein function prediction on real metagenomic data
- Positioned the framework for extension with curated references, GO taxon constraints, and ontology-aware evaluation

Introduction

The functional characterization of proteins in metagenomic datasets remains a central challenge in microbiome research. As sequencing technologies reveal a vast diversity of environmental sequences, most newly predicted proteins lack experimental validation or close homologs, limiting the ability to assign molecular functions with confidence.

Recent advances in computational annotation pipelines, such as DeepGOmeta [9] and InterProScan [4], provide useful predictions based on sequence, structure, or domain signatures, yet their accuracy and contextual interpretation in metagenomic settings remain limited by the sparsity and ambiguity of available data.

Knowledge graphs (KGs) offer a principled way to integrate heterogeneous biological data, such as taxonomic, genomic, and functional annotations, into a unified, structured representation that can be queried and modeled using graph machine learning techniques.

KGs represent biological knowledge as triples (subject, predicate, object), allowing the modeling of relationships such as 'contig encodes protein' or 'protein has_function GO:0004672'. Embedding such graphs into continuous vector spaces enables downstream tasks such as link prediction, where missing edges, for example, unobserved protein-function associations, can be inferred.

In metagenomic contexts, KG construction has recently been proposed as a tool to support downstream applications such as microbial host [7] and feeding efficiency modeling in animal microbiomes [11]. Notably, Zhang et al. (2024) [12] proposed AEKE, an error-aware KG embedding model that adjusts the weight of triples during training by considering structural consistency, attribute-based similarity, and anomaly signals.

This is particularly relevant for prediction of metagenomic functions, where spurious associations, such as a bacterial contig annotated with a eukaryote-specific function, are common due to overgeneralized prediction methods. However, current approaches often overlook critical biological constraints, such as those defined by the Gene Ontology (GO) taxon constraints ('*in_taxon*' and '*never_in_taxon*'), which could be leveraged to improve prediction validity.

In this project, I aim to establish a baseline framework for integrating metagenomic functional annotation into a knowledge graph, and to evaluate its performance on protein function prediction tasks. I construct a KG from metagenomic assemblies by integrating taxonomic assignments, protein predictions, and molecular function annotations. To reduce graph complexity while maintaining biological meaning, I use representative proteins for clustering and focus exclusively on GO molecular function terms. Five canonical KG embedding models—TransE, TransF, TransR, TransH, and TransD—are trained and compared using link prediction metrics.

This leads to the central research question: *What is an effective baseline framework for integrating metagenomic*

functional annotation into a knowledge graph, and how can it inform strategies to improve protein function prediction?

This work establishes a minimal yet extensible foundation for evaluating function prediction in metagenomic data using KGs. It also opens the path for future improvements, including higher-confidence functional annotation via InterProScan, integration with curated reference datasets (e.g., E. coli proteins), incorporation of GO taxon constraints to filter implausible triples, and ontology-aware evaluation strategies that account for hierarchical relationships in GO rather than treating ancestor terms as false positives. As such, this framework can serve as a robust starting point for error-aware and biologically grounded function prediction in large-scale environmental sequence data.

Methods

Data collection

The sequencing data was selected from one of the soil samples collected across the Empty Quarter in February 2024: 'F57PRr2', which corresponds to plant rhizosphere in site 57. In general, DNA of samples was extracted using QIAGEN kits: DNeasy PowerLyzer PowerSoil and DNeasy PowerSoil Pro. Samples were further sequenced using Illumina NovaSeq 6000 in a paired-end format, with each read having a length of 150 base pairs.

Assembly

First, I processed the raw reads using fastp (v0.22.0) [1], applying overlap-based error correction, quality filtering (minimum Phred score 20, maximum 40% low-quality bases), and complexity filtering. Then I assembled the reads into contigs using MEGAHIT (v1.2.9) [6] with *–presets meta-sensitive*. I evaluated the quality and contiguity of these assemblies using QUAST (v5.2.0) [2] and remapping original reads to the final assembly using Bowtie (v2.5.1) [5].

Annotation

From the resulting metagenomic assemblies, I first performed taxonomic assignment of contigs using Kraken2 (v2.1.3) [10]. Coding sequences (CDSs) were predicted using Prodigal (v2.6.3) [3] in metagenomic mode.

Predicted proteins were then clustered using MMseqs2 (v14.7) [8] with an initial sequence identity threshold of 0.5 and coverage of 0.9 (`--min-seq-id 0.5 -c 0.9`). To increase clustering sensitivity, I later removed the coverage constraint (`-c 0.9`).

Functional annotation of proteins was performed using DeepGOmeta [9], which predicts Gene Ontology (GO) terms across three categories: molecular functions, biological processes, and cellular components. Only the most specific annotations were retained, as the more general can be inferred through the Gene Ontology.

Knowledge Graph Construction

To represent the relationships among contigs, predicted proteins, and functional annotations, I constructed a knowledge graph (KG) using biological triples derived from the processed metagenomic assemblies. The KG consisted of five core types of relations:

- `(contig, has_taxa, taxon)`
- `(contig, encodes, protein)`
- `(protein, member_of, cluster)`
- `(protein, has_function, GO_term)`
- `(GO_term, is_a, GO_term)`

To reduce the size and complexity of the graph while preserving biological meaning, I applied two optimizations: (1) proteins were clustered using MMseqs2, and only the representative sequence of each cluster was retained for annotation; and (2) only Gene Ontology terms from the Molecular Function (MF) category were used, excluding Biological Process (BP) and Cellular Component (CC) terms.

Model Training

To evaluate the ability of knowledge graph embeddings to recover missing functional annotations, I trained five models—TransE, TransF, TransR, TransH, and TransD—using the PyKEEN library. These models learn vector representations of entities and relations by optimizing a margin-based ranking loss to distinguish true triples from corrupted ones.

The knowledge graph was split such that 80% of the `has_function` triples were used for training and 20% for testing, while all other triples were included in the training set to preserve contextual information.

Each model was trained under multiple settings, varying batch sizes (1000 or 2000) and embedding dimensions (128, 150, or 256) to assess performance under different learning capacities.

Model Evaluation

To assess the performance of the embedding models in predicting missing `(protein, has_function, GO_term)` triples, I conducted a link prediction task using standard evaluation metrics. During evaluation, I defined a whitelist of protein entities corresponding to the test set. This ensured that predictions were only evaluated on proteins for which ground truth annotations were available, avoiding spurious scoring on unannotated entities. Evaluation was restricted to the `has_function` relation, and negative samples were generated by replacing the GO term with random terms not associated with the protein.

Model predictions were ranked by plausibility scores, and the top candidates were evaluated using the following metrics:

- **Hits@10**: The proportion of test triples for which the correct GO term was ranked among the top 10 predictions.
- **Mean Rank (MR)**: The average rank position of the correct GO term among all predictions.
- **Mean Reciprocal Rank (MRR)**: The average inverse of the rank assigned to the correct GO term, favoring correct predictions ranked near the top.

These metrics provide a comprehensive view of both accuracy and ranking quality, allowing comparison across different embedding models and KG configurations.

Results

Assembly general statistics

The assembly used for the analysis consists of 335k contigs which encode a total of 292k proteins. When clustering

the proteins with the parameters `--min-seq-id 0.5 -c 0.9` I obtained 257k clusters. I decided to remove the coverage parameter to make the clustering step less strict, which resulted in 226k clusters.

Knowledge graph overview

The original design of the assembly consisted 8.8M edges, despite changing the clustering parameters, the size did not significantly reduce in size. Therefore, further adjustments were done, such as functionally annotating directly only the representatives of the clusters, and focusing only in the molecular functions category of the gene ontology.

This resulted in a sparse but semantically coherent KG suitable for downstream embedding and prediction tasks. The final graph consisted of 2.4M edges.

Model training

Model names follow the format `Model- epochs- batch size- embedding size`, which indicates the training configuration used for each run. The following figures display training loss curves for the models, which were grouped based on the range of loss values.

Most models, including TransD, TransE, and TransH, show rapid loss decay within the first 20 epochs, followed by a plateau. TransR models, however, exhibit minimal improvement across epochs, suggesting poor training performance. In contrast, TransF models (Figure 2) start with higher losses but converge rapidly, likely due to their greater modeling capacity. These trends highlight TransD and TransF as more effective at capturing relational structure, while TransR consistently underperforms.

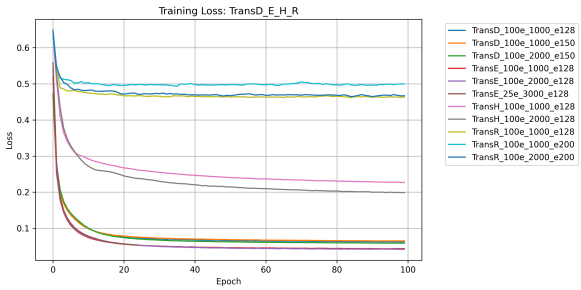

Fig. 1. Training loss curves for models: TransD, TransE, TransH, and TransR

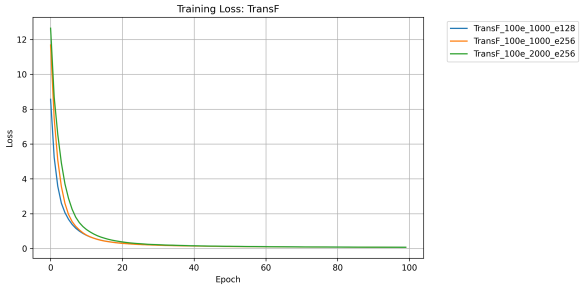

Fig. 2. Training loss curves for the TransF models

Model evaluation

Model performance was evaluated using standard link prediction metrics: Hits@10, Median Rank, Harmonic Mean Rank, and Mean Reciprocal Rank (approximated as the inverse of Harmonic Mean Rank). Results are summarized in Table 1. The best-performing model overall was `TransH_100e_2000_e128`, which achieved the highest Hits@10 (39.44%) and lowest median rank (15), followed closely by `TransD_100e_1000_e150` (Hits@10 of 39.10%) and several TransF configurations (up to 37.80% Hits@10).

These models also showed the lowest harmonic mean ranks and highest inverse harmonic mean ranks (MRR), indicating better ranking of correct tail predictions. In contrast, TransR consistently underperformed across all configurations, with lower Hits@10 scores (below 17%) and substantially worse rank-based metrics, confirming the trend observed during training loss analysis. These findings highlight TransH, TransD, and TransF as the most effective architectures for function prediction in this knowledge graph context.

Discussion

Based on both loss convergence and evaluation metrics, TransH demonstrated the best overall performance among the five evaluated knowledge graph embedding models. It achieved the highest Hits@10 (39.44%), the lowest median rank (15), and competitive harmonic mean rank and MRR values, confirming its ability to generate biologically relevant and well-ranked predictions. Given these results, TransH was selected for future work and further optimization in this knowledge graph-based function prediction framework.

To improve the biological relevance and reliability of predictions, several enhancements to the current framework are proposed. First, instead of relying solely on DeepGOmeta for functional annotation, future versions will use **InterProScan**, a more conservative tool for predicting protein functions based on domain architecture. This is expected to reduce overannotation and increase confidence in GO term assignments.

Second, the integration of proteins from **well-annotated reference** organisms such as E. coli can help anchor novel or metagenomic sequences to known biology. By merging curated annotations with environmental data in the knowledge graph, it becomes possible to improve function transfer across species and enhance embedding quality.

Third, the use of Gene Ontology (GO) **taxon constraints** will be implemented to biologically validate predicted functions. GO includes relations such as *only_in_taxon* and *never_in_taxon* that restrict certain functional terms to specific clades. Incorporating these constraints before or during training—by assigning lower confidence to biologically implausible triples or filtering them altogether—will help correct errors such as assigning a eukaryote-specific function to a bacterial protein. This approach aligns with error-aware methods like AEKE, which adjust the influence of noisy or inconsistent triples based on structural and attribute-based coherence.

Fourth, the **evaluation methodology** will be revised to respect the hierarchical nature of the GO. Standard link prediction metrics (e.g., Hits@10, MRR) assume flat label structures and penalize the prediction of more general but correct ancestor terms. This is inappropriate for ontology-based annotations. Future evaluation will therefore consider hierarchy-aware metrics, where the prediction of an ancestor

Table 1. Tail Prediction Metrics for Knowledge Graph Models

Model	Hits@10 (%)	Median Rank	Harmonic Mean Rank	Inverse Harmonic Mean Rank
TransE_25e_3000_e128	34.10	43	5.48	0.1830
TransE_100e_2000_e128	28.30	80	8.27	0.1210
TransE_100e_1000_e128	31.80	46	7.27	0.1370
TransF_100e_2000_e256	37.80	27	**4.68**	**0.2130**
TransF_100e_1000_e256	33.70	27	4.69	**0.2130**
TransF_100e_1000_e128	34.10	43	5.29	0.1890
TransD_100e_2000_e150	32.40	62	5.20	0.1920
TransD_100e_1000_e150	39.10	27	4.69	**0.2130**
TransD_100e_1000_e128	34.40	43	5.29	0.1890
TransR_100e_2000_e200	16.26	40	15.62	0.0640
TransR_100e_1000_e200	15.84	61	16.29	0.0614
TransR_100e_1000_e128	16.49	34	14.85	0.0673
TransH_100e_2000_e128	**39.44**	**15**	6.24	0.1601
TransH_100e_1000_e128	36.13	16	6.87	0.1455

term is not penalized if the ground truth is its descendant. Additionally, metrics specific to protein function prediction will be incorporated, including:

$F_{\max}$: the maximum F1 score computed over varying thresholds, commonly used in CAFA challenges.

Semantic distance metrics: accounting for the information content or graph distance between predicted and true terms.

Precision, recall, and coverage: to measure not only accuracy but also the functional breadth of predictions.

Together, these improvements will move the system beyond technical accuracy toward biologically valid, reliable, and interpretable function prediction, making it more suitable for real-world metagenomic analysis and annotation.

Conclusion

This study presents a scalable and interpretable framework for protein function prediction from metagenomic data using knowledge graphs. By integrating taxonomic, genomic, and molecular function annotations into a structured graph and evaluating five different embedding models, I established a baseline for context-aware function inference. Among the evaluated models, TransH demonstrated the best overall performance, making it the most promising candidate for future development.

The results highlight the importance of leveraging biological context—such as clustering, GO hierarchy, and taxonomic assignments—when addressing the inherent ambiguity of metagenomic annotations. Additionally, the work lays the foundation for multiple improvements, including more reliable functional annotation using InterProScan, the integration of well-curated reference data, the enforcement of GO taxon constraints, and the adoption of ontology-aware evaluation strategies that go beyond standard link prediction metrics.

Overall, this framework moves toward a more biologically grounded and semantically informed approach to protein function prediction, addressing the limitations of traditional sequence-similarity methods and opening new avenues for high-quality metagenomic annotation.

Data availability

Github repository: https://github.com/alelvz/FunKG.git

References

1. Shifu Chen, Yanqing Zhou, Yaru Chen, and Jia Gu. fastp: an ultra-fast all-in-one fastq preprocessor. *Bioinformatics*, 34(17):i884–i890, 2018.
2. Alexey Gurevich, Vladislav Saveliev, Nikolay Vyahhi, and Glenn Tesler. Quast: quality assessment tool for genome assemblies. *Bioinformatics*, 29(8):1072–1075, 2013.
3. Doug Hyatt, Gwo-Liang Chen, Philip F LoCascio, Miriam L Land, Frank W Larimer, and Loren J Hauser. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC bioinformatics*, 11:1–11, 2010.
4. Philip Jones, David Binns, Hsin-Yu Chang, Matthew Fraser, Weizhong Li, Craig McAnulla, Hamish McWilliam, John Maslen, Alex Mitchell, Gift Nuka, et al. Interproscan 5: genome-scale protein function classification. *Bioinformatics*, 30(9):1236–1240, 2014.
5. William B Langdon. Performance of genetic programming optimised bowtie2 on genome comparison and analytic testing (gcat) benchmarks. *BioData mining*, 8:1–7, 2015.
6. Dinghua Li, Chi-Man Liu, Ruibang Luo, Kunihiko Sadakane, and Tak-Wah Lam. Megahit: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de bruijn graph. *Bioinformatics*, 31(10):1674–1676, 2015.
7. Jie Pan, Zhen Zhang, Ying Li, Jiaoyang Yu, Zhuhong You, Chenyu Li, Shixu Wang, Minghui Zhu, Fengzhi Ren, Xuexia Zhang, et al. A microbial knowledge graph-based deep learning model for predicting candidate microbes for target hosts. *Briefings in Bioinformatics*, 25(3):bbae119, 2024.
8. Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.
9. Rund Tawfiq, Kexin Niu, Robert Hoehndorf, and Maxat Kulmanov. Deepgometa for functional insights into microbial communities using deep learning-based protein function prediction. *Scientific Reports*, 14(1):31813, 2024.
10. Derrick E Wood, Jennifer Lu, and Ben Langmead. Improved metagenomic analysis with kraken 2. *Genome biology*, 20:1–13, 2019.
11. Junmei Zhang, Qin Jiang, Zhihong Du, Yilin Geng, Yuren Hu, Qichang Tong, Yunfeng Song, Hong-Yu Zhang, Xianghua Yan, and Zaiwen Feng. Knowledge graph-derived

feed efficiency analysis via pig gut microbiota. *Scientific Reports*, 14(1):13939, 2024.

12. Qinggang Zhang, Junnan Dong, Qiaoyu Tan, and Xiao Huang. Integrating entity attributes for error-aware knowledge graph embedding. *IEEE Transactions on Knowledge and Data Engineering*, 36(4):1667–1682, 2023.