Enriching Clinical Sample Analysis with Biological Knowledge Graphs: A Preliminary Study

Fatemeh Shad Bakhsh[1], Juan Manuel Rodriguez[1], Alessandro Ranieri[2], Kenneth Kastaniegaard[2] and Daniele Dell'Aglio[1]

[1]*Department of Computer Science, Aalborg University, Aalborg, Denmark*
[2]*Biogenity, Aalborg, Denmark*

Abstract

Biological researchers often face challenges in analyzing clinical samples due to the limited amount of samples they can collect. This issue hinders the use of traditional statistical methods; instead, they often rely on their domain knowledge to guide the exploration of the data. To ease the task, we aim to develop a system to support the researchers by integrating data from biological knowledge graphs (KGs), such as Reactome and UniProt, which can drive data exploration through recommendation system techniques. In this article, we present the first step towards such a system by studying whether the data from biological KGs can be represented through embeddings so that proteins of interest are organized and categorized according to their shared functionalities. We propose Cluster-GAE, a graph autoencoder method inspired by Cluster-GCN. Cluster-GAE combines graph sampling techniques with Graph Neural Networks (GNNs) to learn embedding representations from large-scale biological networks. Our experiments suggest that Cluster-GAE produces embeddings with properties similar to the embeddings of traditional KG embedding methods without the need to process the whole graph at once. Moreover, the experiments show that the embeddings produced by Cluster-GAE are of a higher quality than the embeddings produced by a KG obtained through traditional sampling techniques, in particular Random Walk and Forest Fire. Finally, through t-SNE visualization and functional enrichment analysis, we showcase the ability of Cluster-GAE to identify protein clusters that are related to different biological processes, molecular functions, and cellular components.

Keywords

Knowledge graphs, Graph Neural Networks, Pathway analysis

1. Introduction

Biological researchers collect clinical samples to study living organisms, quantifying the amounts of thousands of proteins of interest and looking for relations among them. However, collecting samples is expensive and time-consuming. Therefore, trials often include only a limited number of samples [1]. Moreover, the number of features tends to be much greater than the number of samples, and considering small perturbation, finding the features related to a particular phenomenon is a complex task. Currently, state-of-the-art methods, such as Stabl [2], rely on multiple sampling and data augmentation to find relevant features.

Biological databases, such as Reactome [3], KEGG [4], UniProt [5], are invaluable tools for enhancing the analysis of biological data, as they act as encyclopedic knowledge about proteins, reactions, and biological pathways. These databases are often structured through graph-based data structures or knowledge graphs. Hence, to ease the analysis of datasets with a few data points (the samples) and many features (the proteins), we argue for leveraging this information to create an item-item recommender system for discovering relevant features leveraging the information encompassed by Reactome or UniProt. In particular, we propose to rely on graph representation learning to capture the relevant

HealthRecSys'24: The 6th Workshop on Health Recommender Systems co-located with ACM RecSys 2024

fbakhs22@student.aau.dk (F. S. Bakhsh); jmro@cs.aau.dk (J. M. Rodriguez); ara@biogenity.com (A. Ranieri); kkas@biogenity.com (K. Kastaniegaard); dade@cs.aau.dk (D. Dell'Aglio)

0000-0002-1130-8065 (J. M. Rodriguez); 0000-0003-3806-5638 (K. Kastaniegaard); 0000-0003-4904-2511 (D. Dell'Aglio)

structural and semantic information from the graph by using Graph Auto-Encoders (GAE) [6] to extract node embeddings to discover relevant features within the original dataset.

In this context, this work presents an exploratory analysis of different graph encoder capabilities to capture relevant information on the node embeddings. Specifically, we analyze the effectiveness of GAE and graph sampling techniques to leverage biological knowledge graphs, such as **Reactome** and **UniProt**, to create embeddings. Such embeddings can later be used to discover protein-protein associations in the context of analysis of limited numbers of clinical samples. We propose *Cluster-GAE*, a method that combines GAE and Cluster-GCN [7], a technique for applying Graph Convolutional Network (GCN) to large graphs. We experimental study the embeddings generated by Cluster-GAE and other GAE solutions. We observe that Cluster-GAE organizes protein embeddings so that they can be clustered according to their biological features. This is a promising result towards the creation of a recommender system to support biological researchers.

In the following, we first introduce the background and related work in Section 2. Next, we illustrate the overall solution we envision in Section 3, and we describe the part we implemented so far, which is later experimentally analysed in Section 4. We conclude with some remarks and the next steps in Section 5.

2. Background and related research

Let P be a set of proteins of interest and y a phenomenon, e.g. being sick. Let S be a clinical sample data set. Each clinical sample can be represented as a vector of size $|P| + 1$, which contains the amount of proteins of interest and the presence or absence of the phenomenon. Therefore, S can be represented as a matrix of size $|P| + 1 \times n$, where n is the number of clinical samples.

Biologist researchers are interested in discovering relations between groups of proteins (i.e. subsets of P) and the phenomenon through the analysis of S. The main challenge for the analysis is the number of proteins $|P|$, which is much larger than the number of clinical samples n. This is because the collection of clinical samples is costly and time-consuming, so trials often involve a number of subjects [8].

This combination of few data instances and numerous features presents substantial challenges for conventional statistical analysis [9, 2]. Even well-known sparsity-promoting methods, such as LASSO [10], are hindered as small errors in the samplings might result in completely different results.

A natural way to cope with the problem is to adopt feature selection techniques like Stabl [2]. Stabl adds noise by generating artificial features and repeatedly subsamples the dataset, looking for features that are frequently selected in the different runs. However, Stabl is over-conservative as it tries to minimize the false discovery ratio that depends on the quality of the generated features and subsampling of the existing data.

Alternatively, one can consider exploiting pre-existing knowledge about the proteins P, coming from literature, to drive the analysis of S. Such knowledge is, for example, stored in biological knowledge graphs (KGs) such as UniProt [5] or Reactome [3]. They store information about proteins, genes, genomes, and pathways, along with their relations and functions. Moreover, such KGs are usually well curated and connected to each other through links between the same entity in different KGs. As a result, there is a massive amount of data that can be useful to drive the analysis process of biological researchers.

While the traditional manner of exploiting these knowledge graphs is to manually query the proteins and navigate through the relations [11], representation learning emerged in the recent years as a means to feed KG data to machine learning algorithms, to solve tasks like classification and recommendation. For example, Burkhart et al. [12] propose to enrich biological prediction models using a Graph Convolutional Network (GCN) to enrich the samples in the dataset. As a result, the data is not processed as tabular data, but information about the relations of the different features is added to the predictive model. Although this model increments the information in the model, it is not a feature reduction technique, and the results are hard to interpret.

Pershad et al. [13] show that the use of node embeddings can be more effective than traditional drug recommendation methods. Unlike Burkhart et al. [12], the base graph is not Reactome but a graph computed using a probabilistic algorithm over the samples that return protein-protein-interaction networks. In particular, this work presents evidence that the Node2Vec algorithm can derive meaningful embeddings using these PIP networks.

However, biological researchers are rarely equally interested in all proteins. They have a number of proteins they are interested in, and they aim to discover what are the connected proteins, and how they vary w.r.t. each other. Therefore, we argue that this can be modeled as an item-item recommendation task, where given one or more proteins, the goal is to retrieve a list of related proteins which can be later analyzed by the researchers.

3. A recommender system for clinical sample analysis

Figure 1 shows a high-level schema of the solution we envision to support researchers in analyzing clinical samples. There are three main steps, represented by the dark boxes. The first is the sample-based *selector*, which takes the clinical samples S and a biological KG as inputs, and outputs a KG with information specific to S. The second is the embedding *extractor*, which learns embeddings from the KG produced by the selector. Finally, the protein *recommender* uses the embeddings to compute the list of recommended proteins given the input ones. In the following, we provide information on the current implementation of the components.

The sample-based selector

Instead of working on the whole graph, we filter a subgraph with the proteins of interest. To do it, we query Reactome to extract the part of the graph that contains the protein/features present in the clinical samples to analyze. Specifically, for each protein in P, we extract the nodes of the features and nodes up to two hops of distance.

This is implemented as Cypher queries, which are evaluated over Reactome using Neo4J. Listing 1 shows an example query, which is parametrized. At Line 1, the query reads the list of protein names (Line 1). At Lines 2 and 3, the query retrieves the Reacome nodes associated with the input proteins. The block at Lines 4–7 retrieves the neighbor nodes (at a maximum distance of two) of the input protein nodes. The block at Lines 10-11 adds constraints on the nodes to be retrieved, such as being related to mice (mus musculus), a possible type of organism studied by the clinical samples. Finally, the operation at Line 12 returns the edges.

After the construction of the out graph, we verified its connectivity, to avoid having isolated proteins which could not be processed in the next steps.

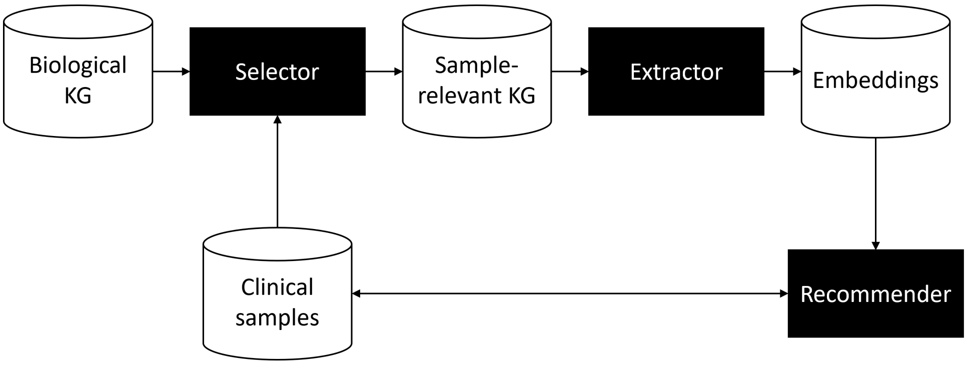


Figure 1: The recommender system architecture

Listing 1: The query executed by the sample-based selector

```
UNWIND $proteinNames AS proteinName
MATCH (p)
WHERE ("EntityWithAccessionedSequence" IN labels(p) OR
        "GenomeEncodedEntity" IN labels(p)) AND
        ANY(name IN p.name WHERE name = proteinName)
CALL apoc.path.subgraphAll(p, {
  maxLevel: 2,
  minLevel:1
})
YIELD nodes, relationships
WITH p,
      [node in nodes WHERE node.speciesName = $specie] AS filteredNodes,
      relationships
RETURN p AS protein, filteredNodes AS nodes, relationships
```

The embedding extractor

Despite the graph induced by the clinical samples is smaller than the original biological knowledge graph, its size is still large. Therefore, the learning algorithm that extracts the embeddings needs to exploit an adequate strategy to cope with the graph volume. We considered two alternatives: the GAE extractor and the Cluster-GAE extractor. The former uses graph sampling to reduce the size of the knowledge graph and generates the embeddings using a GAE. The latter does not perform the graph sampling. Instead, it uses the approach initially proposed by Cluster-GCN based on a divide-and-conquer strategy.

The *GAE extractor* exploits graph sampling to reduce the size of the graph. The goal is to improve computation performance while retaining the essential structural properties of the original graph. As sampling algorithms, we consider Random Walk (RW) with Restart [14] and Forest Fire (FF) [14]. We set as a constraint the necessity of having a connected graph as an output of the process.

After that, a GAE model learns low-dimensional, yet informative representations of the graph's nodes. The GAE model comprises an encoder and a decoder, with the encoder being a GCN that embeds nodes into a latent space. The decoder then attempts to reconstruct the graph's adjacency matrix from these embeddings. This process is facilitated by a reconstruction loss function that guides the model to capture the essential topological similarities based on features between nodes [6].

The *Cluster-GAE extractor* divides the data into partitions and then samples from these partitions during the GAE training phase. This extractor omits the sampling step described above. Instead, it adapts the Cluster-GCN algorithm [7], which enhances computational efficiency by dividing the graph into multiple clusters. We replace GCN with a GAE, as the former solves a node classification task, while we aim at learning embeddings to build a recommendation task. The GAE was built using the architecture and loss function and task as defined in [6].

This modification allows us to generate meaningful embeddings and understand the underlying structure of the graph. Basically, this method divides the data into partitions and then samples from these partitions during the training phase.

The protein recommender

The last step consists of using the embeddings to group features using clustering or detecting similar proteins/features using embedding distance, recommending how to group the features to analyze to the researcher.

Table 1
Comparison of Clustering Metrics for Different Embedding Dimensions and Sampling Methods for K-means with $K = 2$ and $K = 3$. Metrics include Silhouette Score (Sil), Calinski-Harabasz Index (CH), and Davies-Bouldin Index (DB), evaluated for embedding dimensions of 64 and 128.

Sampling Method	**K-Means $K = 2$**						**K-Means $K = 3$**					
	d = 64			**d = 128**			**d = 64**			**d = 128**		
	Sil	**CH**	**DB**	**Sil**	**CH**	**DB**	**Sil**	**CH**	**DB**	**Sil**	**CH**	**DB**
RW	**0.672**	791.043	0.866	0.417	1130.384	1.083	0.502	1141.947	0.777	0.420	904.776	0.943
FF	0.432	1068.283	0.972	0.460	1261.628	0.969	0.513	1433.257	**0.707**	0.497	1129.500	0.805
Cluster-GAE	0.551	**2138.868**	**0.736**	**0.534**	**2026.599**	**0.576**	**0.775**	**1794.486**	0.770	**0.556**	**1615.266**	**0.761**
No Sampling	0.369	784.169	1.281	0.226	394.942	1.922	0.438	864.572	0.918	0.270	454.419	1.493

4. Preliminary Analysis of the Solution

This section presents our preliminary analysis of the system, specifically on the embedding extractor component. We analyze the behavior of the GAE and Cluster-GAE extractors by analyzing the quality of the generated embeddings through cluster analysis. The code for the experiments is available at https://github.com/dkw-aau/clinical-sample-enrichment.

4.1. Experimental setting

Dataset

We use a dataset consisting of clinical samples from mice characterized by $3\,825$ proteins, one boolean variable associated with the presence of the phenomenon of study, and 48 data instances.

Using the features of the dataset, we query Ractome with a query similar to the one in Listing 1. The query builds a graph with $50,164$ nodes and $1,667,138$ edges. Compared to Reactome, which includes $2,427,555$ nodes and $10,102,445$ edges, we observe that the generated graph is denser, as it preserves 2.06% of the nodes and 16.5% of the edges.

Evaluation Metrics

To evaluate the extractors, we consider two metrics.

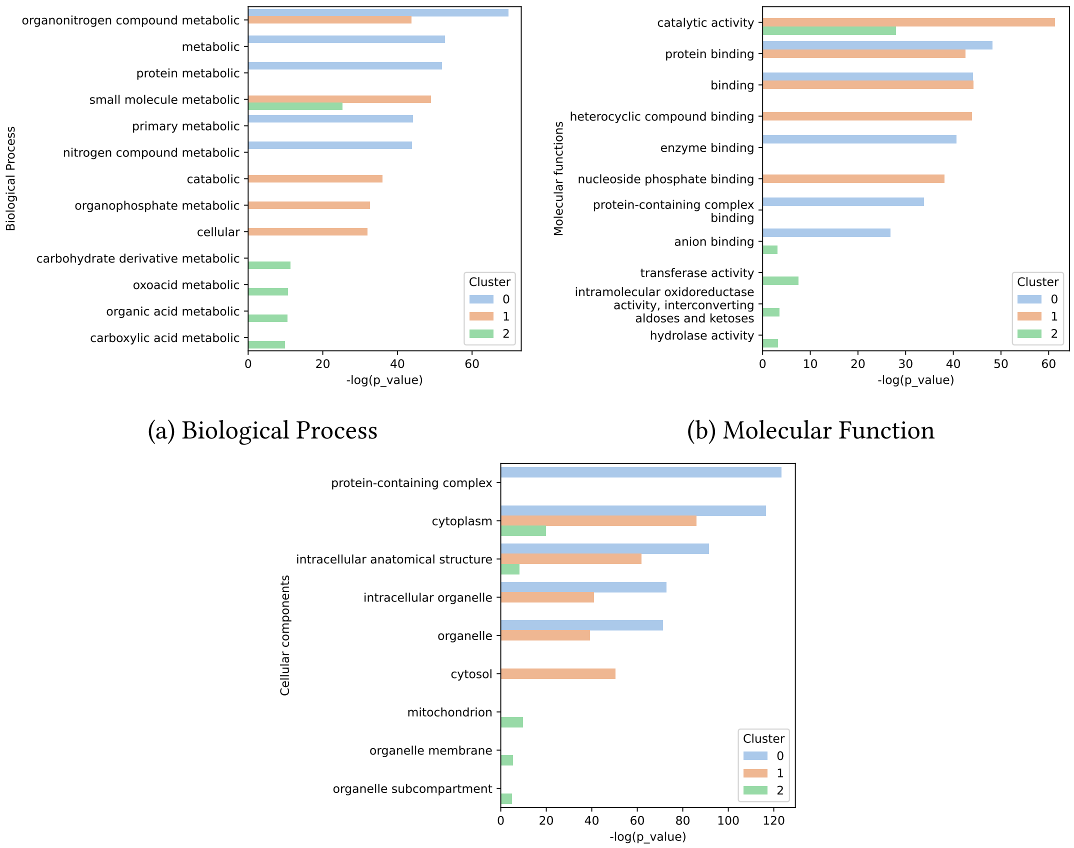
Firstly, we use the *Earth Mover's Distance (EMD)*. EMD evaluates the dissimilarity between two probability distributions, offering a quantitative assessment of the differences in protein embedding matrices across different sampling methods [15].

Secondly, we use *clustering metrics*. These metrics evaluate the quality of clustering results by assessing how well the identified clusters adhere to desirable properties like compactness, separation, and connectedness. The clustering metrics we consider are:

- **Silhouette Score**: Measures how similar an object is to its own cluster compared to other clusters. A higher silhouette score indicates better-defined clusters [16].
- **Calinski-Harabasz Index**: Measures the ratio of the between-cluster dispersion to the within-cluster dispersion. A higher Calinski-Harabasz index indicates denser and more well-separated clusters [17].
- **Davies-Bouldin Index**: Measures the average similarity between each cluster and its most similar cluster. A lower Davies-Bouldin index indicates better cluster separation [18].

4.2. Effect of sampling

We study how the sampling method affects the generated embeddings. We create graph embeddings using the sampling pipeline but omit the sampling step. As a direct comparison of the embeddings is not feasible due to the randomness in the learning process, we compare the distances between embeddings of various proteins in the dataset. We calculated the cosine distance for all protein embeddings derived from



(a) Biological Process

(b) Molecular Function

(c) Cellular Components

Figure 2: Top 5 over-represented pathways in the clusters after doing the functional enrichment analysis.

different extractions: the GAE extractor with RW and, FF and the Cluster-GAE extractor. Subsequently, we compute the EMD between the extractors and a baseline, named *No sampling*. This baseline consists of a GAE extractor without any sampling procedure.

Table 2
The Earth Mover's Distances between the extractors and the baseline

Sampling Method	$d = 64$	$d = 128$
RW	0.481	0.495
FF	0.277	0.321
Cluster-GAE	**0.081**	**0.308**

Table 2 presents the EMD values for embeddings of dimensions 64 and 128. We observe that the behaviour of the Cluster-GAE embeddings is the most similar to the one of the baseline. This is evidence that preserving the input graph allows the exploitation of more information in the learning process.

Looking into the performance of the two GAE extractors, we observe that the one using FF as a sampling method produces closer embeddings to the baseline. When the embedding dimension is 128, the EMD value of the GAE extractor with FF is closer to the one of the Cluster-GAE extractor than the GAE extractor with RW. The results suggest that FF is effective in preserving the input graph structure.

4.3. Cluster analysis

While the previous analysis offers some insights into the effect of sampling, it does not provide useful information about the quality of the embeddings. Therefore, we perform a cluster analysis to obtain

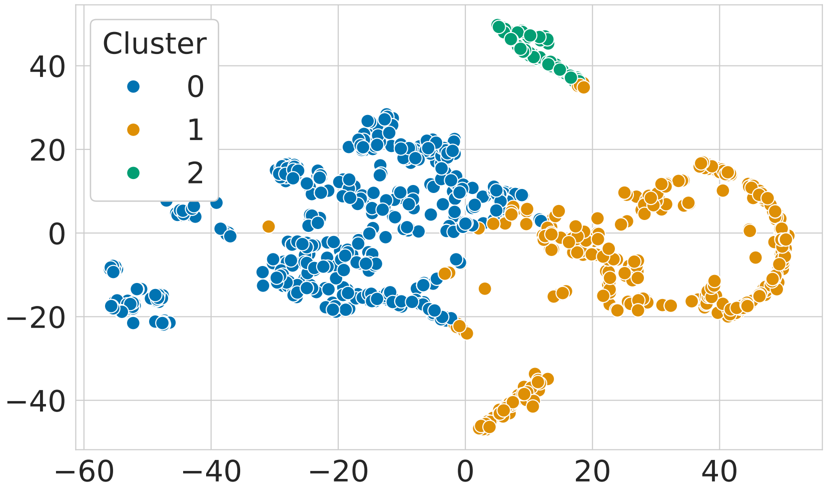


Figure 3: t-SNE visualization of protein embeddings clustered using K-means ($K = 3$) with Cluster-GAE.

qualitative and quantitative insights on the learned space.

As the clustering algorithm, we use K-means, as this algorithm is well-recognized for its effectiveness with embeddings across various domains [19, 20]. Specifically, we consider K-means with $K = 2$ and $K = 3$.

Table 1 displays the cluster metrics for the embeddings produced by the various extractors. In general, Cluster-GAE produces embeddings that are distinctly clustered and well-separated. This observation holds true regardless of the vector dimensions.

To visually investigate the space created by the Cluster-GAE extractor, we used t-SNE. Figure 3 shows the embeddings with dimensions created by Cluster-GAE and clustered through K-means when $K = 3$. The figure shows that there are two prominent clusters and a third, smaller yet more distinct cluster.

4.4. Functional enrichment analysis

We further investigate the quality of the embedding generated by Cluster-GAE through a functional enrichment analysis [21].

We use g:Profiler[1] to run such an analysis. This tool takes a set of proteins and a target organism as inputs and identifies overrepresented biological processes, molecular functions, and cellular components to which those proteins might be related. To analyze the properties of the clusters, we run the g:profile analysis using the proteins of clusters. We expect each cluster to be related to different results in a large way. This means that the quality of the clustering is reflected in the degree to which these functions are distinct between clusters.

We divided the analysis by the three top-level categories of the Gene Ontology, namely “Biological Process” (BP), “Molecular Function” (MF), and “Cellular Components” (CC). We consider $p_{value} < 0.05$ when performing the functional enrichment analysis. Figure 2 presents the top-5 results for each category divided into clusters. This Top-5 is based on $-log_{10}(p_{value})$, i.e., these are the results with the lowest p_{value}. We expect that an effective clustering algorithm would identify proteins that are related to specific BPs, MFs, and CCs in the same group. In other words, if different clusters share few functional results, the clustering is effective in capturing the distinct BPs, MFs, and CCs; therefore providing evidence of the biological relevance of the clusters.

The results show that for BPs, there is a clear separation of the top-5 retrieved processes for each

[1] https://biit.cs.ut.ee/gprofiler/gost

cluster. However, there is an overlap between Clusters 0 and 1 for “organonitrogen compound metabolic” and between proteins in Clusters 1 and 2 for the “small molecule metabolic.” All in all, there is a clear difference. If we consider all the results, Cluster 0 might be involved in 682, and 67.01% of them are unique to this cluster, Cluster 1 might contribute to 462 processes, where 47.15% of them are only related to this Cluster. Cluster 2 contributes to 72 processes and uniquely contributes to 25.0% of them. Moreover, 43.06% of Cluster 2 processes are shared with both Cluster 0 and 1. In this sense, Cluster 2 mainly comprises proteins that can participate in the BPs of both Clusters.

When considering MFs, there is a more extensive overlap between Clusters 0 and 1 on the potential MFs. However, there are some unique identifiable MFs associated with each cluster. Regarding all MFs retrieved for each Cluster, Clusters 0, 1, and 2 might be associated with 122, 93, and 23 MFs, respectively. 59.02% of Cluster 1 MFs are unique to that cluster, 44.09% of the MFs of Cluster 2 are not shared, and 43.48% of the MFs of Cluster 2 did not appear in other Clusters. Therefore, for molecular MFs, we can see that clustering separates the molecules according to their functions.

Something similar happens with MFs when we consider CCs. In the top 5, we observe mostly shared results for Clusters 0 and 1, with some unique components. When analyzing the full results of the functional enrichment, we observed that Clusters 0, 1, and 2 are related to 220, 151, and 20 CCs, respectively, and 64.09% are unique to Cluster 0, 45.03% are unique to Cluster 1, and 15.0% are unique to Cluster 2. Like in the case of molecular function, Cluster 2 encompasses proteins that are related to common CCs, as 65.0% of the 20 CCs are shared by both Clusters 0 and 1.

In summary, the results suggest that the embeddings convey information about the protein’s role in BPs, MFs, and CCs. Therefore, applying k-means using these embeddings results in clusters with a large number of unique roles in the type of results of functional enrichment analysis.

5. Conclusion and Future Works

This article presented our ongoing research in building support tools for biological researchers. The solution we envision exploits biological KGs to suggest new proteins for researchers to study in the context of clinical sample analysis.

We presented some possible strategies to learn embeddings from a biological KG. The experimental analysis suggested that Cluster-GAE is effective in processing biological KGs and creating embedding spaces, when compared to embedding techniques that exploit sampling to reduce the size of the graph.

Moreover, the embeddings of the proteins learned through Cluster-GAE can effectively be clustered according to the biological processes they contribute to. The functional enrichment analysis using g:Profiler highlights that proteins grouped within the same cluster tend to be associated with similar biological pathways and processes.

One of the current limitations we face is that we used one dataset. It is necessary to repeat the analysis on different datasets, ideally associated to various organisms and with diverse biological conditions, to ensure that our findings generalize.

The next natural step is to build the recommendations on top of the learned embeddings. The first approach is to rely exclusively on the embeddings and compute recommendations from them (e.g., through nearest neighbor techniques). However, the recommendation process should also take advantage of the information coming from clinical samples. Despite limited in the size of the samples, the contained values are a valuable source of information that should contribute to the recommendation process.

The natural focus to build a recommender system is on non-personalized recommender systems. However, the presence of various researchers analyzing the same samples opens opportunities for collaborative filtering approaches. Furthermore, incorporating user-interaction data, such as previous protein selections or research interests, could further enhance the personalization and relevance of recommendations.

Currently, the major issue for creating a recommender system for protein selection in biological research is the lack of datasets to evaluate its performance. State-of-the-art methods in protein selection in a dataset, such as Stabl [2], follow a stochastic approach. Since these methods aim to optimize the performance of predicting models by selecting a subset of proteins, and the number of proteins is much greater than the number of samples, there is a great risk of overfitting. However, the results of these methods can be used for a first validation.

With time, our goal is to create a real-life dataset by providing researchers with recommendations and storing their feedback. After collecting a dataset, we plan to extend the recommender system to consider not only the biological knowledge graph as input but also details of the research being carried out to provide personalized recommendations.

Acknowledgments

This research has been partially supported by AI Denmark.

References

[1] P. Feist, A. B. Hummon, Proteomic challenges: Sample preparation techniques for microgram-quantity protein analysis from biological samples, International Journal of Molecular Sciences 16 (2015) 3537–3563. URL: https://www.mdpi.com/1422-0067/16/2/3537. doi:10.3390/ijms16023537.

[2] J. Hédou, I. Marić, G. Bellan, J. Einhaus, D. K. Gaudillière, F.-X. Ladant, F. Verdonk, I. A. Stelzer, D. Feyaerts, A. S. Tsai, E. A. Ganio, M. Sabayev, J. Gillard, J. Amar, A. Cambriel, T. T. Oskotsky, A. Roldan, J. L. Golob, M. Sirota, T. A. Bonham, M. Sato, M. Diop, X. Durand, M. S. Angst, D. K. Stevenson, N. Aghaeepour, A. Montanari, B. Gaudillière, Discovery of sparse, reliable omic biomarkers with stabl, Nature Biotechnology (2024). doi:10.1038/s41587-023-02033-x.

[3] B. Jassal, L. Matthews, G. Viteri, C. Gong, P. Lorente, A. Fabregat, K. Sidiropoulos, J. Cook, M. Gillespie, R. Haw, et al., The reactome pathway knowledgebase, Nucleic acids research 48 (2020) D498–D503.

[4] M. Kanehisa, M. Furumichi, Y. Sato, M. Ishiguro-Watanabe, M. Tanabe, Kegg: integrating viruses and cellular organisms, Nucleic acids research 49 (2021) D545–D551.

[5] T. U. Consortium, Uniprot: the universal protein knowledgebase in 2023, Nucleic acids research 51 (2023) D523–D531.

[6] T. N. Kipf, M. Welling, Variational graph auto-encoders, arXiv preprint arXiv:1611.07308 (2016).

[7] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, C.-J. Hsieh, Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, Association for Computing Machinery, New York, NY, USA, 2019, pp. 257–266.

[8] I. Subramanian, S. Verma, S. Kumar, A. Jere, K. Anamika, Multi-omics data integration, interpretation, and its application, Bioinformatics and Biology Insights 14 (2020) 1177932219899051. doi:10.1177/1177932219899051.

[9] E. Candès, Y. Fan, L. Janson, J. Lv, Panning for Gold: 'Model-X' Knockoffs for High Dimensional Controlled Variable Selection, Journal of the Royal Statistical Society Series B: Statistical Methodology 80 (2018) 551–577. doi:10.1111/rssb.12265.

[10] T. H. Noah Simon, Jerome Friedman, R. Tibshirani, A sparse-group lasso, Journal of Computational and Graphical Statistics 22 (2013) 231–245. doi:10.1080/10618600.2012.681250.

[11] R. Haw, H. Hermjakob, P. D'Eustachio, L. Stein, Reactome pathway analysis to enrich biological discovery in proteomics data sets, Proteomics 11 (2011) 3598–3613. doi:10.1002/pmic.201100066.

[12] J. G. Burkhart, G. Wu, X. Song, F. Raimondi, S. McWeeney, M. H. Wong, Y. Deng, Biology-inspired graph neural network encodes reactome and reveals biochemical reactions of disease, Patterns 4 (2023) 100758. doi:10.1016/j.patter.2023.100758.

[13] Y. Pershad, M. Guo, R. B. Altman, Pathway and network embedding methods for prioritizing psychiatric drugs, Pac Symp Biocomput 25 (2020) 671–682.

[14] J. Leskovec, C. Faloutsos, Sampling from large graphs, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '06, Association for Computing Machinery, New York, NY, USA, 2006, p. 631–636. doi:10.1145/1150402.1150479.

[15] Y. Rubner, C. Tomasi, L. J. Guibas, The earth mover's distance as a metric for image retrieval, International journal of computer vision 40 (2000) 99–121.

[16] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of computational and applied mathematics 20 (1987) 53–65.

[17] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, Communications in Statistics-theory and Methods 3 (1974) 1–27.

[18] D. L. Davies, D. W. Bouldin, A cluster separation measure, IEEE transactions on pattern analysis and machine intelligence PAMI-1 (1979) 224–227.

[19] R. Mussabayev, N. Mladenovic, B. Jarboui, R. Mussabayev, How to use k-means for big data clustering?, Pattern Recognition 137 (2023) 109269. doi:10.1016/j.patcog.2022.109269.

[20] Y. Yu, Q. Liu, L. Wu, R. Yu, S. L. Yu, Z. Zhang, Untargeted attack against federated recommendation systems via poisonous item embeddings and the defense, Proceedings of the AAAI Conference on Artificial Intelligence (2023) 4854–4863. doi:10.1609/aaai.v37i4.25611.

[21] U. Raudvere, L. Kolberg, I. Kuzmin, T. Arak, P. Adler, H. Peterson, J. Vilo, g:Profiler: a web server for functional enrichment analysis and conversions of gene lists (2019 update), Nucleic Acids Research 47 (2019) W191–W198. doi:10.1093/nar/gkz369.