MEDINFO 2021: One World, One Health – Global Partnership for Digital Innovation
P. Otero et al. (Eds.)

doi:10.3233/SHTI220071

POKR: Building a Computable Heterogeneous Knowledge Resource for Precision Oncology

Yanbo Guo[a,c], Chen Wang[b], Raymond Moore[b], Hongfang Liu[c], Feichen Shen[c]

[a] *Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, USA*
[b] *Department of Quantitative Health Sciences, Mayo Clinic, Rochester, MN, USA*
[c] *Department of Artificial Intelligence & Informatics, Mayo Clinic, Rochester, MN, USA*

Abstract

Precision oncology is expected to improve selection of targeted therapies, tailored to individual patients and ultimately improve cancer patients' outcomes. Several cancer genetics knowledge databases have been successfully developed for such purposes, including CIViC and OncoKB, with active community-based curations and scoring of genetic-treatment evidences. Although many studies were conducted based on each knowledge base respectively, the integrative analysis across both knowledge bases remains largely unexplored. Thus, there exists an urgent need for a heterogeneous precision oncology knowledge resource with computational power to support drug repurposing discovery in a timely manner, especially for life-threatening cancer. In this pilot study, we built a heterogeneous precision oncology knowledge resource (POKR) by integrating CIViC and OncoKB, in order to incorporate unique information contained in each knowledge base and make associations amongst biomedical entities (e.g., gene, drug, disease) computable and measurable via training POKR graph embeddings. All the relevant codes, database dump files, and pre-trained POKR embeddings can be accessed through the following URL: https://github.com/shenfc/POKR.

Keywords:

Precision Oncology, Computable Heterogeneous Knowledge Resource, Knowledge Graph Embeddings

Introduction

Cancer is responsible for millions of deaths worldwide every year. Although significant progress has been achieved in cancer medicine, many issues remain to be addressed for improving cancer therapy. For this reason, oncological research is putting a lot of effort towards finding new and efficient therapies. Precision oncology [1] is one of the research directions that aims to improve selection of targeted therapies tailored to individual patients and ultimately improve cancer patients' outcomes. With the advancement of computational capacity and power, in silico drug discovery [2] further accelerates the process and provides timely support for cancer researchers. Several cancer genetics knowledge databases have been successfully developed for such purposes, including Clinical Interpretation of Variants in Cancer (CIViC) [3] and OncoKB [4], with active community-based curations and scoring of genetic-treatment evidences. Many researchers leveraged these two knowledge resources in their studies or applications. For example, Bertucci et al [5]. used OncoKB to determine the actionability of somatic genetic alterations in a metastatic breast cancer genomic characterization study. Janjigian et al [6]. utilized OncoKB to infer the oncogenic effect and clinical actionability of individual somatic mutations in a study to build genetic predictors of therapeutic response in esophagogastric cancer. In addition, CIViCpy [7] was designed by Wagner et al as a Python software development and analysis toolkit for the CIViC knowledgebase. Lever et al [8]. used the CIViC database to curate clinically relevant cancer biomarkers mined from 121,589 PubMed abstracts and full-text papers. According to a comparative analysis, although CIViC and OncoKB have a highly similar goal and creation process and also acquire the data from the same origin, each one holds a substantial amount of unique information [9]. Therefore, integrative analysis of multimodal features buried in the two knowledge resources holds potential power to facilitate precision oncology research (e.g., cancer drug repurposing). However, such efforts are largely unexplored.

In this pilot study, we sought to integrate the heterogeneous precision oncology knowledge resources, CIViC and OncoKB, in order to incorporate unique information contained in each knowledge base and make associations amongst biomedical entities (e.g., gene, drug, disease) computable and measurable. Specifically, we first merged entities contained in both resources and stored the enriched precision oncology knowledge resource (POKR) into a graph database named Neo4j. A graph database can best represent the connected network and is more intuitive when it comes to a sophisticated network not only for explicit relationships but implicit relationships. Therefore, we choose neo4j as our carrier for our graph database. We then trained graph embeddings for the POKR and made quantitative and qualitative analysis for the vectorized entities. On one hand, the POKR stored in Neo4j can provide intuitive visualization and graph topological analysis for non-techinical users. On the other, the POKR embeddings can be viewed as a pre-trained data resources to assist technical users to conduct further investigation for different research purposes in precision oncology.

In the following, we first introduce materials used in this study. Next, we describe the methods for constructing POKR and corresponding graph embeddings. We then present evaluation results followed by discussion.

Background and Materials

CIViC

CIViC is a community-driven open-source resource for Clinical Interpretation of Variants in Cancer. The goal of CIViC is to disseminate knowledge with regard to precision medicine. It can be viewed as a knowledge graph that stores large

amount of cancer-related genes, variants, drugs, disease and their associations/links. In this study, we utilized the CIViC 1/1/2021 version, which contains 2,586 nodes and 6,724 edges.

OncoKB

OncoKB is a precision oncology knowledge repository that contains information related to the effects and treatment implications of specific cancer gene alterations. All the data is organized into a knowledge graph format. In this study, we extracted 377 nodes and 1,916 edges from the OncoKB version that was released on 11/12/2020.

Neo4j

Neo4j is a graph database that is able to not only store the data itself but also store the relationships between data. In this study, all the graph data contained in CIViC and OncoKB were stored in Neo4j. Combining both database in Neo4j gives us 2,808 nodes and 8,464 edges in total.

Methods

The overall workflow is generally divided into three steps. As shown in Figure 1, we first integrated CIViC and OncoKB. We then dumped the integrated knowledge resource into a graph knowledge base, Neo4j. In addition, we trained the corresponding knowledge graph embeddings for the integrated knowledge resource.

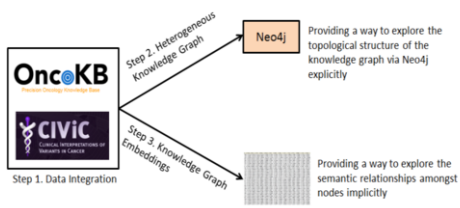

Figure 1. Overall study workflow.

Data Integration and Graph Generation

The OncoKB and CIViC data we acquired is stored in csv format with every row representing one connected relationship between two entities. Although we used csv files in our study, we could utilize all kinds of files as long as the file contains the entity and relationship between entities.

The entity is denoted as a node in our graph database and the relationship between entities is denoted as an edge between two nodes. The nodes can be labeled as "gene", "disease", "variant of a gene" or "drug for a disease". Other than "label", a node also has other customized optional attributes such as "name" or anything related to that node.

We read all the csv files into neo4j by rows using Cypher (a language built for Neo4j), and we then got a well-labeled, connected, informative graph database. By asserting uniqueness on each node in Neo4j, we removed duplicate nodes or relationships. With this clean and concise graph database, it is easy to export the final integrated dataset for generating embeddings. Since we leveraged the node2vec algorithm in this study to calculate the embeddings, we focused on the connection between the source node and target node. Therefore, we exported a file consisting of the source node and the corresponding target node, with which we will be able to train embeddings that best represent the network. Figure 2 shows an example of how the data is stored in Neo4j. Nodes in the same color represents same type of the node. The label on the link represents the type of relationship between nodes. For instance, the gene "ERG" may cause disease Ewing Sarcoma and disease Prostate Carcinoma. We can extract information such as all relationships connected with a certain gene, or all the diseases that can be cured by a certain medicine, or extract one type of relationships that we are interested in.

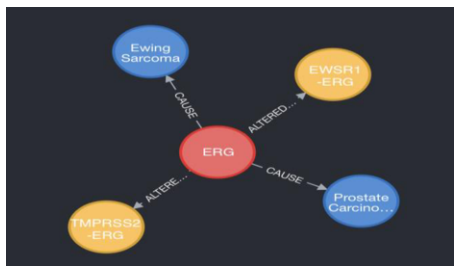

Figure 2. A peek from Neo4j with the red node as a gene named ERG and all its relationships with other nodes that connect with ERG.

Computable Graph Embeddings Generation

We applied the node2vec model to generate embeddings. Node2vec [10] is introduced to graph data for embeddings from word2vec, which is originally applied on natural language processing. Node2vec uses biased random walk to sample linear paths in a graph. A path is similar to a sentence in an article; nodes along the path resembles the words in a sentence in word2vec. By using neural network to predict the adjacent node within certain window of the target node, we can learn the embeddings for the target node from the weight matrix.

Node2vec used second-order random walk to sample a customized length path. To specify the process, we call these three nodes involved in the random walk source node E_s, intermediate node E_m, and target node E_t. Let c_i denote the ith node in the walk. Node c_i are generated by the following distribution:

$$\mathbf{P}(\mathbf{c_i} = \mathbf{E_t} | \mathbf{c_{i-1}} = \mathbf{E_m}) = \begin{cases} \frac{\boldsymbol{\pi}(\mathbf{E_m}, \mathbf{E_t})}{\mathbf{Z}} & \text{if } (E_i, E_t) \text{is an edge} \\ \mathbf{0} & \text{otherwise} \end{cases} \quad (\mathbf{1})$$

Where $\pi(E_m, E_t)$ is an unnormalized transition probability, Z is a constant for normalization. Given the weight over edge (E_m, E_t) as $w(E_m, E_t)$, $\pi(E_m, E_t)$ can be calculated as:

$$\boldsymbol{\pi}(\boldsymbol{E_m}, \boldsymbol{E_t}) = \boldsymbol{\alpha}(\boldsymbol{E_m}, \boldsymbol{E_t}) * \boldsymbol{w}(\boldsymbol{E_m}, \boldsymbol{E_t}) \quad (\mathbf{2})$$

where $\boldsymbol{\alpha}(\mathbf{E_m}, \mathbf{E_t})$ is a biased term introduced in node2vec involving two hyperpaprameters p and q to balance between breadth-first search and depth-first search during random walk. Suppose the shortest distance between $\mathbf{E_m}$ and $\mathbf{E_t}$ is denoted as $\mathbf{sd}(\mathbf{E_m}, \mathbf{E_t})$, $\boldsymbol{\alpha}(\mathbf{E_m}, \mathbf{E_t})$ can be calculated like this:

$$\boldsymbol{\alpha}(\boldsymbol{E_i}, \boldsymbol{E_t}) = \begin{cases} \frac{\mathbf{1}}{\boldsymbol{p}} & \boldsymbol{if\ sd}(\boldsymbol{E_m}, \boldsymbol{E_t}) = \mathbf{0} \\ \mathbf{0} & \boldsymbol{if\ sd}(\boldsymbol{E_m}, \boldsymbol{E_t}) = \mathbf{1} \\ \frac{\mathbf{1}}{\boldsymbol{q}} & \boldsymbol{if\ sd}(\boldsymbol{E_m}, \boldsymbol{E_t}) = \mathbf{2} \end{cases} \quad (\mathbf{3})$$

After sampling paths from the graph, we apply the skip-gram model to train embeddings for each node using stochastic gradient descent to optimize the loss function:

$$\max_{f} \sum_{E_s \in E} logP\big(N(E_s)|f(E_s)\big) \quad (4)$$

Where $N(E_S)$ denotes all the sampled neighbors of $E_S \in E$.

Embedding optimization

Since different combinations of hyperparameters will give us different embeddings, it is crucial to find the optimal embeddings for different purposes. We proposed two ways of finding the optimal embeddings: a). link prediction, which focuses more on connections between nodes and b). clustering, which takes account of the grouping of similar nodes.

a) Link prediction

Link prediction provides one way to determine the optimal embeddings. The idea is to predict the relationship between two nodes and use the performance of the prediction to evaluate the quality of the node embeddings. Given two nodes E_s, E_t and their embeddings $f(E_s)$, $f(E_t)$ a representation between nodes E_s, E_t can be calculated using either one of these 4 operations, namely Hadamard, average, weighted L1, weighted L2:

$$Hadamard(E_s, E_t) = f(E_s) * f(E_t) \quad (5)$$

$$Average(E_s, E_t) = \frac{f(E_s) + f(E_t)}{2} \quad (6)$$

$$L1(E_s, E_t) = |f(E_s) - f(E_t)| \quad (7)$$

$$L2(E_s, E_t) == |f(E_s) - f(E_t)|^2 \quad (8)$$

These representations were used to train a supervised binary classification task, in order to determine if there is a connection between nodes E_s and E_t.

Embeddings can provide good link prediction results indicates that this set of node embeddings are encoded with most of the node connection information; it also implies these embeddings can help us find implicit connections between nodes.

b) Clustering

Clustering is another interesting way of selecting embeddings since it focuses more on the similarity information over the nodes. In this study, we used the k-means model [11] to cluster vectorized nodes. K-means model is an unsupervised learning model where we need to decide a k value before the training. The process can be summarized into three steps: select a k-value; initialize the k centroid randomly; use the centroids to cluster all the nodes and find the average in each cluster to reestablish the new centroids.

We used the silhouette score and purity score to evaluate the clusters. silhouette coefficient for each cluster is calculated using the mean intro-cluster distance a and the mean nearest-cluster distance b as $\frac{b-a}{\max(a,b)}$. The silhouette score is the mean of the silhouette coefficient for all the clusters. The silhouette score tells us if clusters have clear boundaries and enough distance between themselves. The silhouette score focuses on the relationships between each cluster while the purity score focuses on the inside of a cluster to see the ratio of the nodes within one cluster that are coming from same category.

To render the high-dimensional embeddings into a lower-dimensional space for us to better visualize the network clustering result, we selected the t-distributed stochastic neighbor embedding (t-SNE) [12]. t-SNE projects high-dimensional embeddings into 2-dimentional space after clustering. Figure 3 shows an example of such clustering after dimension reduction using our dataset.

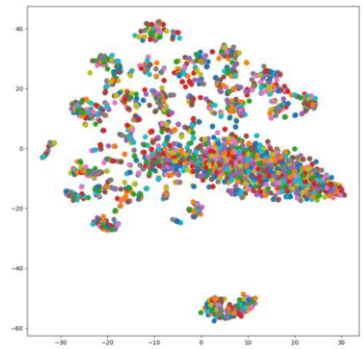

Figure 3. Optimal clustering embeddings renders into 2-dimensional space. Different color presents different cluster the nodes belong to.

Results

For link prediction task, the hyperparameters used in the experiments are window size of 10, number of walks of 5, length of walk of 5, dimension of embedding of 128, return hyperparameter p of 5 and in-out hyperparameter q of 0.65. For clustering task, the hyperparameters used in the experiments are window size of 5, number of walks of 10, length of walk of 10, dimension of embedding of 50, return hyperparameter p of 0.2, in-out hyperparameter q of 0.2, number of clusters k of 8.

Based on our experience, random forest is able to achieve an optimal performance among common classification algorithms. Therefore, we chose random forest as our classification model to train on edge embeddings. For each representation between 2 nodes, label = 1 indicates there is an edge between these two nodes; label = 0 indicates there is no edge between these two nodes. We split the dataset into 60%, 10% and 30% for training, validation and testing for all the positive examples, while randomly sampling the same amount of negative examples and splitting it in the same ratio for training, validation and testing to balance the dataset.

Table 1. Evaluation results for the 4 edge embeddings operation

	Precision	Recall	F1 score
Hadamard	0.88	0.88	0.88
Average	0.88	0.88	0.88
L1	0.86	0.85	0.85
L2	0.86	0.86	0.86

To evaluate the performance, we plotted the receiver-operating characteristic (ROC) curve and computed the area

under the ROC curve. Moreover, we used precision, recall, F1 score to further quantify the link prediction performance among 4 different edge embedding operations. As shown in Table 1, both the Hadamard and Average operations achieved the optimal performance of 0.88 F1 score while L2 yielded a suboptimal result of 0.86 F1 score. As shown in Figure 4, both Hadamard and Average achieved the optimal ROAUC score of 0.93, and L1 and L2 produced the same ROAUC score of 0.91.

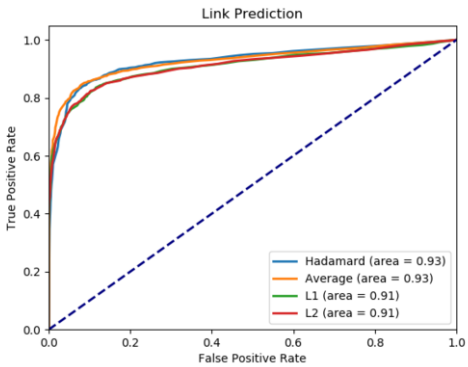

Figure 4. ROC score for 4 different operations.

For clustering task, the optimal silhouette score is 0.4 and the optimal purity score is 0.56. As shown in Table 2, we also picked some clusters and listed the contained biomedical entities accordingly. A full list of clusters could be downloaded from: https://github.com/shenfc/POKR.

Table 2. Selected clusters with biomedical entities

Cluster #	Biomedical Entities	Entity Type
2	Endometrial Serous Adenocarcinoma	Disease
	KRAS:A146	Variant
	Docetaxel	Drug
5	Thyroid Gland Follicular Carcinoma	Disease
	Fluorouracil	Drug
	CDK4	Gene

Discussion and Future Work

In this work, Neo4j provides a way to explore the topology of the precision oncology graph and visualize both the nodes and relationships, which is friendly to non-technical users for knowledge discovery. In addition, Neo4j provides various built-in functions to generate graph embeddings. However, in this study, we sought to construct graph embeddings from scratch by ourselves. The rationale is that we aimed to generate embeddings with different purposes using different tasks (link prediction and clustering), therefore, training embeddings from scratch provided us more flexibilities compared to using Neo4j built-in functions.

Link prediction is able to measure the embeddings using a binary classification task. However, the connections between nodes are solely based on the topological structure of the integrated POKR but without incorporating any prior knowledge as weights from existing ontologies. In the future, it would be very interesting to assign different weightage over the edge based on prior knowledge and build enhanced graph embeddings.

The silhouette and purity scores provide ways to quantify the clustering performance. However, in order to evaluate cluster results qualitatively, it is necessary to deep dive into each cluster for use case studies. As shown in Table 2, we detected Endometrial Serous Adenocarcinoma (ESA), KRAS:A146, and Docetaxel in cluster #2. After conducting literature reviews, we found that KRAS:A146 has a strong association with ESA [13], and Docetaxel is one of the common drugs to treat ESA in practice [14]. Similarly, in cluster #5, we observed that Thyroid Gland Follicular Carcinoma (TGFC) is associated with gene CDK4, which could be approved in [15]. We also found that TGFC could be treated by Fluorouracil from the cluster, which is recorded in one issue of the PDQ cancer information summaries [16]. In this pilot study, we only conducted a preliminary qualitative evaluation by manual literature search. It would be better to invite domain experts to evaluate the clustering results further. In future work, we plan to follow this direction and involve experts in the loop.

Conclusions

In this pilot study, we built a heterogeneous precision oncology knowledge resource, POKR, by integrating CIViC and OncoKB, in order to incorporate unique information contained in each knowledge base and make associations amongst biomedical entities (e.g., gene, drug, disease) computable and measurable via training POKR graph embeddings. In general, we stored the POKR in two formats: 1) a Neo4j visualizable graph database, and 2) POKR embeddings for computational needs. Specifically, we trained two embeddings based on a link prediction task and a clustering task. Results indicated that the integration of heterogeneous knowledge resources hold potential to facilitate knowledge discovery in precision oncology research.

Acknowledgements

This work has been supported by the Gerstner Family Foundation, Center for Individualized Medicine (CIM) of Mayo Clinic, and Genentech Research Fund in Individualized Medicine.

References

[1] Garraway, L.A., J. Verweij, and K.V. Ballman, Precision Oncology: An Overview. Journal of Clinical Oncology, 2013. 31(15): p. 1803-1805.

[2] Terstappen, G.C. and A. Reggiani, In silico research in drug discovery. Trends in Pharmacological Sciences, 2001. 22(1): p. 23-26.

[3] Griffith, M., et al., CIViC: A knowledgebase for expert-crowdsourcing the clinical interpretation of variants in cancer. bioRxiv, 2016: p. 072892.

[4] Chakravarty, D., et al., OncoKB: A Precision Oncology Knowledge Base. JCO Precision Oncology, 2017(1): p. 1-16.

[5] Bertucci, F., et al., Genomic characterization of metastatic breast cancers. Nature, 2019. 569(7757): p. 560-564.

[6] Janjigian, Y.Y., et al., Genetic Predictors of Response to Systemic Therapy in Esophagogastric Cancer. Cancer Discovery, 2018. 8(1): p. 49-58.

[7] Wagner, A.H., et al., CIViCpy: A Python Software Development and Analysis Toolkit for the CIViC Knowledgebase. JCO Clinical Cancer Informatics, 2020(4): p. 245-253.

[8] Lever, J., et al., Text-mining clinically relevant cancer biomarkers for curation into the CIViC database. Genome Medicine, 2019. 11(1): p. 78.

[9] Pallarz, S., et al., Comparative Analysis of Public Knowledge Bases for Precision Oncology. JCO Precision Oncology, 2019(3): p. 1-8.

a. Grover, A. and J. Leskovec, *node2vec: Scalable Feature Learning for Networks*, in *Proceedings of the* 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016, Association for Computing Machinery: San Francisco, California, USA. p. 855–864.

b. Krishna, K. and M.N. Murty, Genetic K-means algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1999. 29(3): p. 433-439.

c. Van der Maaten, L. and G. Hinton, Visualizing data using t-SNE. Journal of machine learning research, 2008. 9(11).

d. O'Hara, A.J. and D.W. Bell, The genomics and genetics of endometrial cancer. Advances in genomics and genetics, 2012. 2012(2): p. 33.

e. Kashima, K., et al., Complete response to docetaxel and carboplatin combination chemotherapy for a stage IV uterine papillary serous carcinoma: a case report. Int J Gynecol Cancer, 2005. 15(6): p. 1199-202.

f. Seyed Abutorabi, E., et al., Abemaciclib (CDK4/6 Inhibitor) Blockade Induces Cytotoxicity in Human Anaplastic Thyroid Carcinoma Cells. Reports of biochemistry & molecular biology, 2020. 8(4): p. 438-445.

g. PDQ Adult Treatment Editorial Board. Thyroid Cancer Treatment (Adult) (PDQ®): Health Professional Version. 2021 Feb 22. In: PDQ Cancer Information Summaries. Bethesda (MD): National *Cancer Institute (US); 2002-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK65719/.*

Address for correspondence

Feichen Shen
shenfeichen1102@gmail.com