Digital Personalized Health and Medicine
L.B. Pape-Haugaard et al. (Eds.)

doi:10.3233/SHTI200147

Document Oriented Graphical Analysis and Prediction

Shorabuddin SYED, MS[a,1], Mahanazuddin SYED, MS[a], Hafsa Bareen SYEDA, MD[a], Fred PRIOR, PhD[a], Meredith ZOZUS, PhD[b], Melody L. PENNING, PhD[a], Mohammed ORLOFF, PhD[c,d]

[a]*Department of Biomedical Informatics, University of Arkansas for Medical Sciences, Little Rock, AR, USA* [b]*Department of Population Health Sciences, University of Texas Health Science Center at San Antonio, San Antonio, TX, USA* [c]*Department of Epidemiology, University of Arkansas for Medical Sciences, Little Rock, AR, USA* [d]*Winthrop P. Rockefeller Cancer Institute, University of Arkansas for Medical Sciences, Little Rock, AR, USA*

Abstract. In general, small-mid size research laboratories struggle with managing clinical and secondary datasets. In addition, faster dissemination, correlation and prediction of information from available datasets is always a bottleneck. To address these challenges, we have developed a novel approach, Document Oriented Graphical Analysis and Prediction (DO-GAP), a hybrid tool, merging strengths of Not only SQL (NoSQL) document oriented and graph databases. DO-GAP provides flexible and simple data integration mechanism using document database, data visualization and knowledge discovery with graph database. We demonstrate how the proposed tool (DO-GAP) can integrate data from heterogeneous sources such as Genomic lab findings, clinical data from Electronic Health Record (EHR) systems and provide simple querying mechanism. Application of DO-GAP can be extended to other diverse clinical studies such as supporting or identifying weakness of clinical diagnosis in comparison to molecular genetic analysis.

Keywords. Data integration, clinical trials, data quality, data processing

1. Introduction

Traditionally, to develop a relational database or data repository, first, the data requirement is analyzed and then data structure is designed based on the requirements [1]. The data structure remains rigid and can only hold the data elements that it was designed for [2]. Then, data integration is performed by feeding-in data from various sources into a data warehouse and processing it after wards. Fixed source to target Extract Transformation and Load (ETL) processes are used to load data into a data repository [3]. Drawback of using relational databases is that it is difficult to keep up with changing data requirements and it takes months to modify source to target mapping and implement ETLs. Not only SQL (NoSQL) databases were developed to overcome fixed data structure limitation of relational data models [4].

[1] Corresponding Author, Shorabuddin Syed, University of Arkansas for Medical Sciences, Little Rock, AR, USA; E-mail: ssyed@uams.edu.

The schema-less design of NoSQL allows flexibility and scalability to add information to any field without a need to define the structure in advance compared to SQL databases [5-7]. One of the common NoSQL database is document-oriented database that uses documents as the structure for data storage and queries, document is a block of XML or JSON [7, 8]. Each document can have the same or different structure. MongoDB is one of the document-oriented database system, it is highly flexible, partition tolerant and consistent [4, 9].

Graph database models their schema as graphs, using nodes and edges. Each node represents an entity (patient, physician etc.) and they are connected by edges, representing relationship between the nodes[10]. Graph database is queried by traversing through the connected links. The graph database are expressive and can be easily conceptualized making them an ideal choice to adopt for data analysis[10]. Moreover, “graph databases excel in traversal-type queries ”[11], they facilitate deep knowledge discoveries using simple-to-mid complex queries. Neo4j (www.neo4j.com) is a graph database designed to manage information with graph-like nature.

DO-GAP attempts to take advantages of both the models (Document and Graph DB) and solves challenges (i) flexible schema that enables faster and more iterative development, and scalable databases (ii) clean and readily available data for graph database. DO-GAP uses MongoDB as document-based data repository and Neo4j graph database for simple analysis and easy visualization.

2. Methods

To address our hypothesis, a pipeline was built for processing of the incoming data, store it in primary data repository, cleansing and transferring study specific data to graph database for analysis and knowledge discovery as shown in Figure 1.

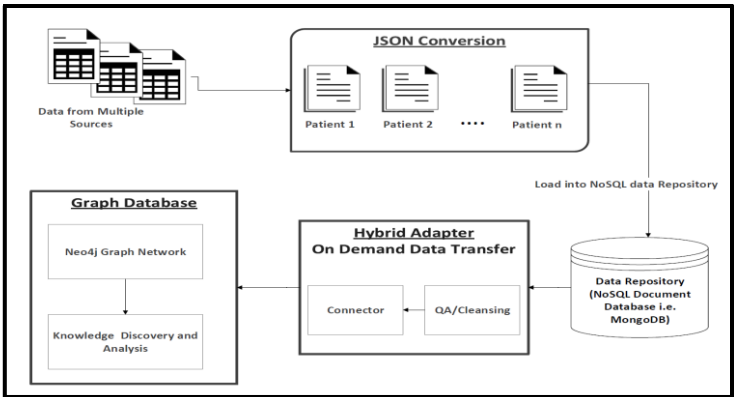

Data from Multiple Sources
JSON Conversion
Patient 1
Patient 2
....
Patient n
Load into NoSQL data Repository
Graph Database
Neo4j Graph Network
Knowledge Discovery and Analysis
Hybrid Adapter
On Demand Data Transfer
Connector
QA/Cleansing
Data Repository (NoSQL Document Database i.e. MongoDB)

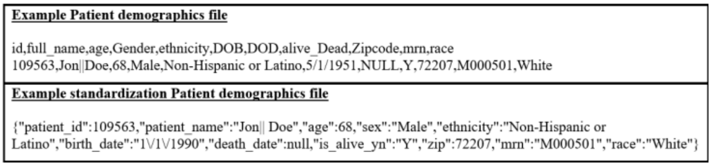
Figure 1. DO-GAP Architecture.

2.1. Data Architecture

The flow starts by collecting and assembling data from multiple sources, extract patient’s longitudinal data from the source files and create a JSON document for each patient. The

JSON files created will be merged into primary data repository, MongoDB. On-demand, study-specific data from MongoDB is cleansed and transferred to graph database, Neo4j.

Laboratories may receive clinical and biological data from multiple sources in different format, frequency and file types. Metadata, a mapping document, for each source file is created to standardize the source data elements. JSON conversion tool creates a JSON document based on source and metadata files. The conversion is independent of source file format, thereby minimizing data pre-processing time. Figure 2, shows an example of patient demographic Comma Separated Values (CSV) file and its equivalent JSON file created using the JSON conversion tool. A master metadata file is maintained during the process.



Example Patient demographics file

```
id,full_name,age,Gender,ethnicity,DOB,DOD,alive_Dead,Zipcode,mrn,race
109563,Jon||Doe,68,Male,Non-Hispanic or Latino,5/1/1951,NULL,Y,72207,M000501,White
```

Example standardization Patient demographics file

```
{"patient_id":109563,"patient_name":"Jon|| Doe","age":68,"sex":"Male","ethnicity":"Non-Hispanic or
Latino","birth_date":"1\/1\/1990","death_date":null,"is_alive_yn":"Y","zip":72207,"mrn":"M000501","race":"White"}
```

Figure 2. A sample demographic file and its equivalent standardized JSON file.

The JSON documents created are loaded into NoSQL document database i.e. MongoDB, using its native data import tool. To inform researchers on the available data fields in DO-GAP, a list of all field names (i.e. Keys) present in the MongoDB and their description is available. This list is updated when new data elements are imported into the database. Researchers can select from the available fields for data extraction and analysis.

Based on user selection, Hybrid adapter cleanses the selected data elements. The cleansing process includes applying standard rules such as removing potential duplicates and null values etc. This cleaned data set will be fed to Connector, it unwinds JSON documents and stores it as deconstructed array document using $unwind, $project, $out MongoDB operators[4]. The deconstructed array document is then written to CSV file using mongoexport utility. Connector uses the generated CSV files to create nodes in Graph database i.e. Neo4j, and builds relationship between the nodes. By this approach only required data elements for a study is selected on-demand, cleansed and made available as a new study in graph database. Thereby overcoming Graph database's data quality and volume limitations[12]. In Graph database component, the data extracted from MongoDB is loaded into Neo4j and is analyzed using native tools such as Neo4j browser and Bloom [13].

3. Application

About 2,300 patients with any type of lung cancer disease[14] were chosen for the study. Selected patient's clinical facts, genomic profiles, comorbidities, social and family history were loaded into DO-GAP for knowledge discovery, resulting in 2,760 nodes and 9,683 relationships. We ran a Cypher query to find the common gene alterations in our sample. As shown in Figure 3, most patients have gene TP53 and LRP1B, TP53 and SPTA1 alterations. Cypher queries are simple to learn as non-technical users can query the database by conceptualizing the graphs, which is difficult with relational databases.

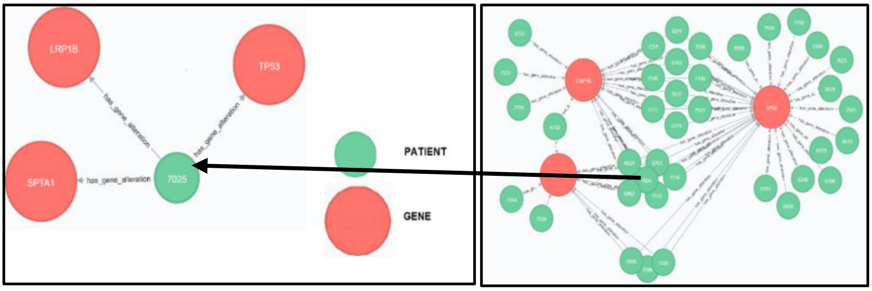


Figure 3. Partial graph showing links between Patient and their Gene alteration.

The output of graph analysis heavily depends upon connectivity between nodes [15], in DO-GAP we can click on any two gene nodes and can ask for common patterns (clinical facts) between them. Graph traverse through all links and displays every node (clinical facts) that is connected to these two gene nodes, hence the data or knowledge discovery.

Graph database can be queried using native visualization tools of Neo4j. In our case, we have used Bloom [16] to demonstrate the ease of querying. As shown in Figure 4, a simple English-like querying language syntax is used to find all patients with any gene alterations and are smokers. In addition, it displays the profile of query results, for example in Figure 4; we have 43 patients that are smokers with at most 10 gene alterations. Graph database query can be expanded to include other clinical facts along with gene information.

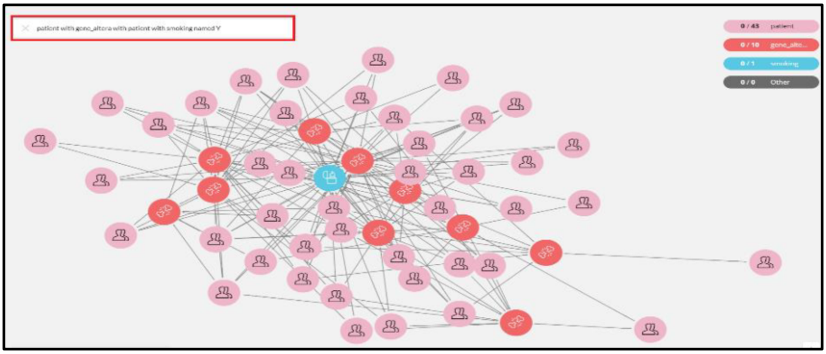

patient with gene_altera with patient with smoking named Y

Figure 4. Very simple query to find patients who are smokers and has gene alteration.

4. Discussion and Conclusion

DO-GAP offers hybrid and flexible data integration, expansion, productive data analysis and prediction with minimal coding. Advantage of using DO-GAP is twofold (1) flexible schema that enable faster and more iterative development, and scalable NoSQL databases. (2) As strength of graph data analysis heavily depends on quality of data, we have achieved it by using hybrid adapter that automatically cleanse user selected data and import to graph database. Graph databases reduces query complexity compared to both relational and MongoDB databases. Use of graph database reduces data analysis and data inference time; we have demonstrated this by two basic use cases (i) common gene alteration, and (ii) simplicity of querying heterogeneous data. As this was a pilot study, we selected the use cases that profile the data, which is the first step in research data analysis.

There are several limitations in this study that we wish to acknowledge. First, to resolve identity of individuals, i.e. entity resolution, which is a complex process in itself and is beyond the scope of this paper. Currently, DO-GAP process these records as separate subjects, in future, an entity resolution systems such as OYSTER [17], can be employed to resolve this limitation. Second, the cleansing component of hybrid adapter only checks for null values and duplicate clinical facts, but can be expanded to incorporate in-depth data quality checks. Finally, results of selected use cases can be easily reproduced using relational databases by technical users. However, DO-GAP was designed to be a data management and analysis tool for non-programmers such as lab technicians, for whom conceptualizing relational database schema and building appropriate queries is challenging task. This pilot project, paves way for future work that will seek to address in-depth graph queries for better findings and correlations. We envision to address aforementioned limitations in our future work that will empower users to explore critical insights which is currently not available.

Funding Acknowledgment and Consent: Patients data used were obtained under IRB approval (IRB#202880) at the University of Arkansas for Medical Sciences (UAMS). The study was supported partly by the Seeds of Science Grant by the Winthrop P. Rockefeller Cancer Institute.

References

[1] Sahatqija K, Ajdari J, Zenuni X, Raufi B, Ismaili F. Comparison between relational and NOSQL databases2018. 0216-21 p.

[2] Kent WJCA. A simple guide to five normal forms in relational database theory. 1983;26(2):120-5.

[3] H Inmon W. Building the Data Warehouse1996.

[4] David Hows Peter M, Eelco Plugge,Tim Hawkins. The Definitive Guide to MongoDB Second ed: Apress; 2013.

[5] Schulz WL, Nelson BG, Felker DK, Durant TJS, Torres R. Evaluation of relational and NoSQL database architectures to manage genomic annotations. Journal of biomedical informatics. 2016;64:288-95.

[6] Leavitt N. Will NoSQL Databases Live Up to Their Promise? Computer. 2010;43(2):12-4.

[7] Wu T-LS. An Overview of Present NoSQL Solutions and Features. .

[8] Imam AA, Basri S, Ahmad R, Watada J, González-Aparicio MT. Automatic schema suggestion model for NoSQL document-stores databases. Journal of Big Data. 2018;5(1):46.

[9] Moniruzzaman ABM, Hossain S. NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison2013.

[10] Renzo Angles CG. Survey of graph database models. ACM Comput Surv. 2008.

[11] Lysenko A, Roznovăţ IA, Saqi M, Mazein A, Rawlings CJ, Auffray C. Representing and querying disease networks using graph databases. BioData Min. 2016;9:23-.

[12] Pokorný J, editor Graph Databases: Their Power and Limitations. Computer Information Systems and Industrial Management; 2015 2015//; Cham: Springer International Publishing.

[13] NeoTechnology. https://neo4j.com/bloom/.

[14] ICD10Data.com. Malignant neoplasm of bronchus and lung C34 [Internet]. [cited on 2018 Dec 9]. https://www.icd10data.com/ICD10CM/Codes/C00-D49/C30-C39/C34-.

[15] Rahman MR. Knowledge Base Population based on Entity Graph Analysis: Université Paris-Saclay; 2018.

[16] NeoTechnology. Neo4j Bloom Quick Start. [Internet]. [cited on 2018 Dec 9]. Available from: https://neo4j.com/docs/bloom-user-guide/current/about-bloom/. [

[17] Talburt JR, Zhou Y. A practical guide to entity resolution with OYSTER. 2013. p. 235-70.