NEO4J FOR ANTIBIOTIC RESISTANCE: AN ALTERNATIVE VIEW OF COMPREHENSIVE ANTIBIOTIC RESISTANCE DATABASE (CARD)

Shubham Singh
Computer Science Department
Illinois Institute of Technology
Chicago,IL
ssingh127@hawk.iit.edu

Aniruddh Suresh Pillai
Computer Science Department
Illinois Institute of Technology
Chicago,IL
apillai8@hawk.iit.edu

***Abstract*— Graph databases are a paradigm shift away from relational databases, with a significant emphasis on "relationships." Unlike relational databases, which compute associations at runtime, graph databases save relationships for fast querying and data retrieval. This paper describes antibiotic resistance- an alternative view of comprehensive antibiotics. We transformed the biological data from relational databases/files and provide new insights. Even though there are numerous graph databases available, Neo4j is without a doubt the industry leader. It's easy to use and scale. Its Cypher language is simple to grasp. Finally, the Graph Data Science Library offers a wide range of powerful graph manipulation and machine learning capabilities.**

I. INTRODUCTION

Graph databases are designed specifically to record and traverse relationships. Relationships are first-class citizens in graph databases, and they account for most of the value of graph databases. Data is saved in the same way that ideas are sketched on a whiteboard. Your data is saved without being constrained by a predefined model, allowing for a very flexible way of thinking about and using it. [1]

Graph databases store data entities in nodes and relationships between entities in edges. An edge always has a start node, an end node, a type, and a direction, and it can indicate parent-child connections, actions, ownership, and other such things. There is no restriction on the number or kind of associations that a node can have. In a graph database, you may traverse a graph along specified edge types or over the whole graph. Because the associations between nodes are not computed at query times but are stored in the database, traversing the joins or relationships in graph databases is highly fast. Graph databases are useful for use cases like social networking, recommendation engines, and fraud detection, where you need to construct linkages between data and easily query these relationships.

When compared to other database types, such as RDBMS, graph databases place the greatest emphasis on relationships. This means that, unlike RDBMS, graph databases do not require the usage of foreign keys or references to compute relationships during execution. Because the operations in relational databases are based on set operations, this causes unanticipated latency or memory utilization in query execution for the RDBMS. RDBMS processes get slower as data size increases. Graph databases, on the other hand, function on connections or pathways and are hence more efficient even for massive data sets.

Graph databases can detect and prevent complex fraud. Relationships in graph databases can be used to conduct financial and purchasing transactions in near-real time. You may use quick graph queries to discover that, for example, a potential purchaser is using the same email address and payment card as a known fraud instance. Graph databases may also assist you in detecting relationship patterns such as several persons linked with the same personal email address or many people having the same IP address but live at separate physical places.

Neo4j, Inc. created a graph database management system called Neo4j. Neo4j is a GPL3-licensed source-available "community edition," with online backup and high availability enhancements distributed under a closed-source commercial license. Its developers describe it as an ACID-compliant transactional database with native graph storage and processing. Neo also sells Neo4j extensions under a closed-source commercial license. Neo4j is written in Java and may be accessed via the Cypher query language over a transactional HTTP endpoint or the binary "Bolt" protocol from software written in other languages.

The graph query language Cypher in Neo4j allows you to retrieve data from the graph. It's like SQL for graphs, and it was inspired by SQL, so you can concentrate on getting the data you want from the graph (not how to go get it). Because of its closeness to other languages and intuitiveness, it is by far the easiest graph language to learn. Cypher is unusual in that it allows you to visually match patterns and relationships. You create a graph pattern over your data when you write a query.

Users of Neo4j use Cypher to build expressive and efficient queries for any type of create, read, update, or delete (CRUD) operation on their graph, and Cypher is Neo4j's primary interface.

Cypher uses an ASCII-art type of syntax where (nodes)-[: ARE_CONNECTED_TO]->(otherNodes) using rounded brackets for circular (nodes), and -[:ARROWS]-> for relationships.

II. RELATED WORK AND BACKGROUND

A. Related Work

According to the available literature, there appears to be an increase in the demand for a flexible schema. As a result, in the context of today's big data applications, the Relational Database Management System is losing importance. Furthermore, with relational database management systems, linking many tables reduces efficiency, whereas graph databases are designed to address these issues.

Relational databases are gradually being replaced by Graph Databases, according to a recent study. Graph databases are used in a wide range of fields, including recommendation engines, the semantic web, and social networking, to name a few. RDBMSs and graph databases, such as Neo4J, distinguish themselves in terms of data model features, query structures, and so on.

In terms of information storage, graph databases provide a refined method for modeling and traversing related data. Applications can be developed in graph database systems, such as Neo4J, that are not limited by the constraints of their RDBMS counterparts.

Graph databases are an excellent source for test case recommendation to get beneficial findings. The interaction between nodes and the integration of entities results in a unique graph structure that supports design recommendations as easy graph traversals.

According to recent studies, as compared to relational databases, a set of predefined queries executes faster. Furthermore, graph databases provide greater flexibility in terms of schema reorganization. This observation is useful when putting commercial systems in place in several fields.

One thing we've observed throughout the COVID outbreak is the indiscriminate, cautious prescribing of antibiotics to patients. As we transition out of COVID, patients may face antibiotic resistance and possibly superbugs. We must assess the best treatment to prescribe for the specific infection, considering if the patient is allergic and whether the drug will be administered intravenously or orally. We can give physicians a fuller image of the antibiotics, possible diagnosis, and therapies by connecting data for patients, diagnoses, dosages, and durations into the Neo4j graph database. Our knowledge graph assists clinicians in providing the most effective care to their patients while minimizing antibiotic resistance and aggravating the medical environment issue.

B. Background

Antibiotics were a breakthrough moment in the history of public health. They are antibiotics that are used to prevent and treat illnesses caused by bacteria, such as pneumonia and tuberculosis. Millions of people have been saved because of antibiotics.

Antibiotic resistance bacteria have emerged because of their overuse and misuse. These bacteria can withstand antibiotics because they have resistomes, which are genes that confer antibiotic resistance. A few of these gene's code for proteins that can reduce or promote drug import, export, or deactivation. Others encode antibiotic-resistant drug targets that have been mutated. Antibiotic resistance genes are transferred both vertically (between mother - daughter cells) and horizontally (among different bacteria), resulting in the fast spread of antibiotic-resistant bacteria over the world.

As a result, we must respond quickly and rely on scientific evidence. The Comprehensive Antibiotic Resistance Database is a vital source of information (CARD).

Since its inception in 2013, the CARD database has accumulated over 3,300 gene sequences and antibiotics. The data was meticulously selected and structured in Antibiotic Resistance Ontology (ARO) and Antimicrobial Resistance (AMR) gene identification models by the team. Bioinformatic tools for data analysis were also given by the database. CARD has grown in importance as a research and industrial data source.

III. ANALYSIS OF GRAPH DATABASES

It is very important to select the proper technology before starting any project/task. Choosing the correct vendor and product may be a difficult process that needs extensive study and frequently comes down to more than just the solution and its technical capabilities. H here are a couple of the best Graph Databases, along with their details and USP's

We are considering Neo4j, AWS Neptune, OrientDB, Apache GraphX in this paper for comparison analysis.

A. Database details and utilization.

NEO4J

The fastest way to graph. Today's Neo4j Graph Data Platform is a package of apps and tools that help the world make sense of data, centered around the leading native graph database.

The Platform contains the Neo4j Graph Data Science Library, which is the leading enterprise-ready analytics workspace for graph data and is accessible as open source as well as through a commercial license for organizations, as well as the graph visualization and exploration tool. Bloom, the Cypher query language (simple to learn and use across Neo4j, Apache Spark, and Gremlin-based products using open-source toolkits: "Cypher on Apache Spark (CApS) and Cypher for Gremlin."), Neo4j ETL and Kettle for data integration, and numerous other tools, integrations, and connectors to help developers and data scientists easily build graph-based solutions.

Neo4j is the most scalable, ACID-compliant graph database in the industry, featuring a high-performance distributed cluster architecture and self-hosted and cloud deployment options. Neo4j assists the globe in making sense of data by allowing enterprises to grasp the intrinsic connection of that data using graph technology. Our principles influence our technology, devotion to our customers, and business culture.[2]

AWS NEPTUNE

Amazon Neptune is a managed graph database available from Amazon.com. It is a web service that is part of Amazon Web Services (AWS). Amazon Neptune supports popular graph models Property Graph and W3C's RDF, as well as their corresponding query languages Apache TinkerPop Gremlin and SPARQL, allowing you to simply design searches that effectively explore densely linked datasets. To create social networking apps, Amazon Neptune can swiftly and easily process massive collections of user profiles and interactions. It allows you to store relationships between information such as customer interests, friends, and purchase history in a graph and quickly query it to make recommendations that are personalized and relevant With Amazon Neptune, you can use relationships to process financial and purchase transactions in near real time to easily detect fraud patterns helps you build knowledge graph applications, Life Sciences, Network / IT Operations etc.[8]

OrientDB

OrientDB is the first Multi-Model Distributed Database Management System (DBMS) with a True Graph Engine. Multi-Model NoSQL is a second-generation NoSQL that can manage complicated domains with exceptional performance. OrientDB handles relationships without the need of JOINs, instead relying on direct pointers. This enables for consistent speed when traversing relationships, regardless of database size.Fully transactional: supports ACID transactions, ensuring that all database transactions are executed reliably and that all outstanding documents are retrieved and committed in the case of a crash.

Graph structured data model: native graph management. Fully compliant with the open-source graph computing framework Apache TinkerPop Gremlin (formerly known as Blueprints), and allows SQL queries with modifications to handle relationships without SQL join, manage trees, and graphs of related documents.

It is Cloud-capable: OrientDB is cloud-deployable and works with the following providers: Amazon Web Services, Microsoft Azure, CenturyLink Cloud, Jelastic, and DigitalOcean. [7]

Apache Graphx

GraphX is Apache Spark's graph and graph parallel computing API (graph processing system). Because it is designed on top of the Apache Spark unified analytics engine for Big Data processing, it integrates ETL, exploratory analysis, and iterative graph computing inside a single system. The information can be seen as graphs or collections. It allows you to efficiently alter and combine graphs with RDDs, as well as develop a custom iterative graph algorithm using the Pregel API. GraphX, like Neo4j's Graph Data Science Library, has a growing library of community-contributed graph algorithms in addition to being a highly flexible API that supports the following use cases PageRank, connected components, label propagation, SVD++, Strongly connected components, and Triangle count can all be done with simplicity. [9]

B. Comparison between graph DB's

Table 1 Comparison between graph databases.

Databases				
Comparison factors	Neo4J	Amazon Neptune	Orient DB	Apache GraphX
Available versions	Enterprise	Enterprise	Community	Community
Query Language	Cypher	Apache TinkerPop	SQL	Java, Scala, Python and R,
Risk Tolerant	Yes	Yes	Yes	Yes
Accessibility	Yes	Yes with replicas	Yes	Yes
Software Platform	Linux, Windows, Mac	AWS Cloud	Linux, Windows, Mac	Linux, Windows, Mac, Container
Security	Data level security not	In Built uses AWS security.	In Built (Data Protection and	In Built, Admin controlled
Advantages	Easy and robust, Cypher language is closely related to	High Throughput,Low Latency Read	storing and querying graphs containing	Inherits Sparks performance, Fastest Graph operations
SAAS Support	Yes	Yes	Yes	Yes

IV. DATA COLLECTION

There are 43 medicines, 263 bacterium pathogens, and 2,640 resistance mechanisms to consider. It's worth noting that there are inconsistencies between the downloaded and web data. According to aro.obo, SHV-52 (ARO:3001109) has no confer resistance to drug class connection. However, according to its website , it can assist the host to resist penam, carbapenem, and cephalosporin (Figure 1) [5]. This report is solely based on the obtained CARD data to avoid any misunderstanding. The project's second flaw is that taxonomic resolution is limited to the species level rather than the strain level, which may have been more exact.

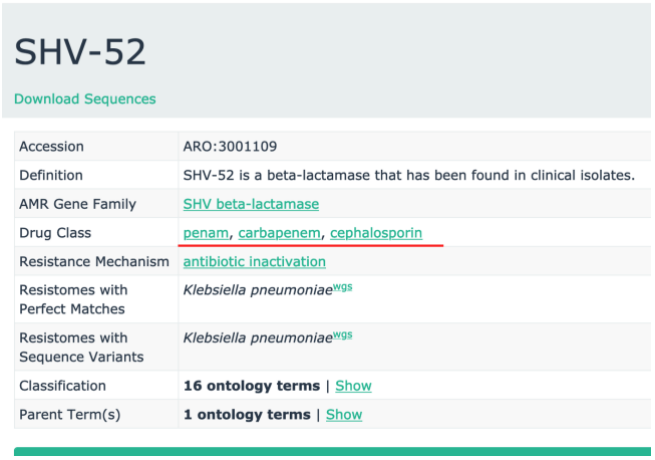

Figure 1

V. DATA ANALYSIS

A python script was created to preprocess the data [5]. Since the dataset was downloaded as 2 files named aro.obo and card_prevalence.txt we first preprocess the data and create csv files which are imported into Neo4j.

VI. DATA INSERTION

The data was inserted into Neo4j using cypher queries. Three different nodes were created, and were labeled as "drug", "pathogen", and 'resistance". Two different kinds of relationships were established, which were "confers_resistance_to_drug_class" and "has_resistance".

A total of 43 drugs, 129 pathogens, and 2338 resistance nodes were created. A total of 595 confers_resistance_to_drug_class and 14781 has_resistance relationships were identified.

VII. RESULTS

A. Overview

With Neo4j Bloom, we can quickly construct a topological perspective of the CARD data (Figure 2).

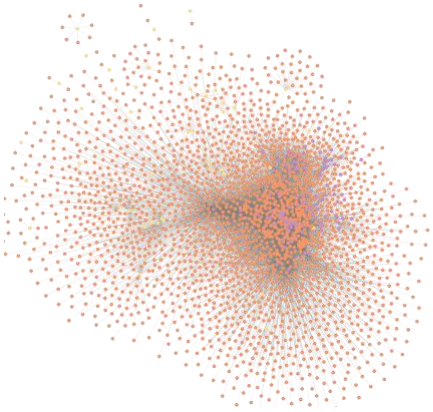

Figure 2

In figure, the data is mostly grouped around the purple pathogen nodes, as can be seen. Some of those enormous "hubs" can be found in Neo4j Bloom. Escherichia coli, Klebsiella pneumoniae, Pseudomonas aeruginosa, and Acinetobacter baumannii are the bacteria. These are also the bacteria that have been investigated the most in scientific studies. [5]

B. Examining Superbugs

Multidrug-resistant bacteria, or superbugs, are microorganisms that are resistant to more than one antibiotic. They can have many genes, each of which confers antibiotic resistance. Alternatively, a single gene can work against a wide range of antibiotics. Carbapenem-resistant Enterobacteriaceae (CRE) have developed resistance to "all or almost all" medicines, including carbapenems, a "last resort" treatment. [5]

We use the following query to get the result of the top 10 superbugs (Figure 3).

MATCH (p:pathogen)-->(r:resistance)
OPTIONAL MATCH (r)-->(d:drug)
RETURN p.name, COUNT(DISTINCT(r)) as resistance_count, COUNT(DISTINCT(d)) as resistance_drug_count ORDER BY resistance_count DESC LIMIT 10;

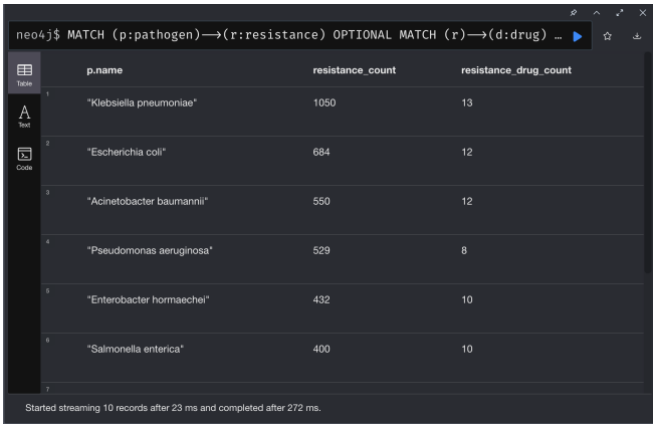

Figure 3

Klebsiella pneumoniae lives on the mucosal surfaces of the gastrointestinal tract (GI). However, if it reaches the human body, it can be extremely contagious and resistant to antibiotics. Klebsiella pneumoniae, as its name implies, causes pneumonia and is the most prevalent cause of hospital-acquired pneumonia in the United States. Other illnesses caused by this bacterium include bloodstream infections, wound and surgical site infections, and meningitis. According to our findings, K. pneumoniae is linked to over a thousand resistance nodes. Resistances to 13 antibiotics are provided by these resistance mechanisms, which is the most in our dataset. [5]

With the following graph (Figure 4) we can see that using the below query.

MATCH path = (p:pathogen)-->(r:resistance) -->(d:drug)
WHERE p.name = "Klebsiella pneumoniae"
RETURN path;

Figure 4

C. Most Resisted Antibiotics

We can also use Neo4j to figure out which antibiotics are most likely to be the targets of antibiotic resistance by using the following query (Figure 5):

MATCH (p:pathogen)-->(r:resistance) -->(d:drug)
RETURN d.name, COUNT(DISTINCT(r)) as resistance_count, COUNT(DISTINCT(p)) as pathogen_count
ORDER BY resistance_count DESC LIMIT 10;

```
1 MATCH (p:pathogen)-->(r:resistance) -->(d:drug)
2 RETURN d.name, COUNT(DISTINCT(r)) as resistance_count,
  COUNT(DISTINCT(p)) as pathogen_count
3 ORDER BY resistance_count DESC LIMIT 10;
```

	d.name	resistance_count	pathogen_count
1	"fluoroquinolone antibiotic"	12	41
2	"cephalosporin"	4	52
3	"disinfecting agents and intercalating dyes"	3	42
4	"aminoglycoside antibiotic"	2	27
5	"lincosamide antibiotic"	2	3
6	"macrolide antibiotic"	2	3
7			

Started streaming 10 records after 9 ms and completed after 41 ms.

Figure 5

Fluoroquinolones cause cell death in Gram-negative and Gram-positive bacteria by interfering with DNA replication. However, according to the CARD data, both types of bacteria have developed a variety of ways to counteract its effects (Figure 6) [5]:

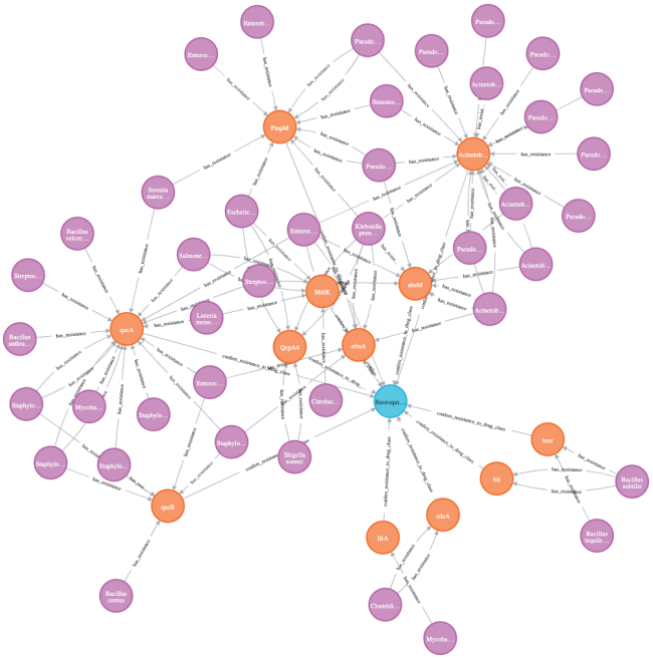

Figure 6

The second item on the list, cephalosporin, has a very different topology (Figure 7).

MATCH path = (p:pathogen)-->(r:resistance) -->(d:drug)
WHERE d.name="cephalosporin"
RETURN path

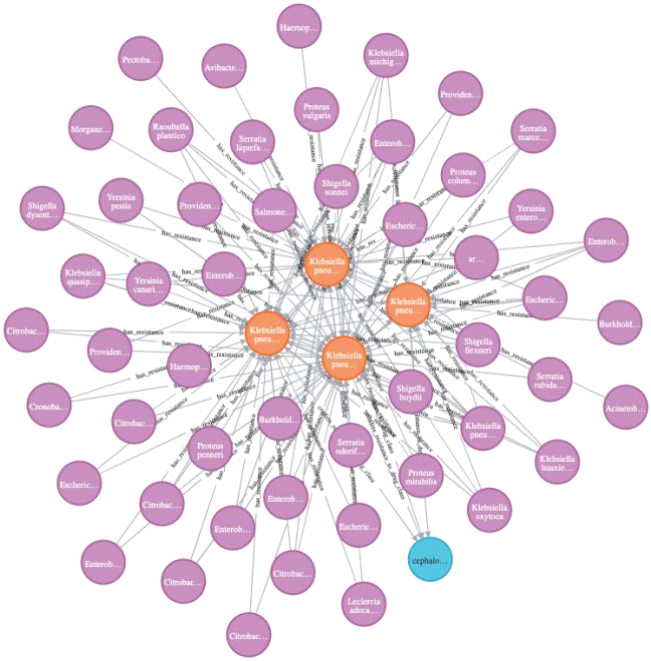

Figure 7

Unlike fluoroquinolone antibiotics, which have numerous resistance pathways, cephalosporin antibiotics have only four. A total of 52 diseases make use of them. Klebsiella oxytoca and Raoultella planticola, for example, have all four resistance pathways[5] .

CONCLUSION AND FUTURE WORK

This project demonstrates how Neo4j may provide rapid CARD insights. CARD, in my perspective, has so many relationships that a graph database, rather than a relational database, is a logical fit for it. Because CARD relates bacteria and antibiotics through resistance mechanisms, the answer to the inquiry "which antibiotics is this bacterium resistant to" is not immediately clear in its relational form. For Neo4j, this isn't a problem. We can ask the query in Cypher and receive an immediate response.

However, the downloaded CARD data does not always match the online data. Some of the relationships are also speculative. According to the description, cdeA provides fluoroquinolone resistance in E. coli and acriflavine resistance in Clostridioides difficile. However, our findings show a link between C. difficile and fluoroquinolones via the cdeA pathway.

We've only looked at the bacterial species in this project based on the card-prevalence.txt file. However, using the NCBI accession numbers provided by CARD, we can obtain the strain resolution. The outcomes would have been more accurate and specific. Of course, it would have been simpler if CARD had compiled the pathogens on a strain-by-strain basis from the start.

This research has also provided some light on the issue of antibiotic resistance. Resistance mechanisms outnumber pathogens by a factor of ten in the graphs above. It means that germs can have up to ten methods to counteract antibiotic

effects. It was also obvious that resistance mechanisms are far more numerous than antibiotics.

References

[1] https://aws.amazon.com/nosql/graph/

[2] https://neo4j.com/developer/graph-database/

[3] https://www.g2.com/products/neo4j/reviews

[4]https://neo4j.com/blog/combating-antibiotic-resistance-the-medical-climate-crisis/

[5] https://card.mcmaster.ca/

[6] https://en.wikipedia.org/wiki/Graph_database

[7] https://orientdb.org/docs/3.0.x/misc/Overview.html

[8] https://aws.amazon.com/neptune/

[9] https://spark.apache.org/graphx/