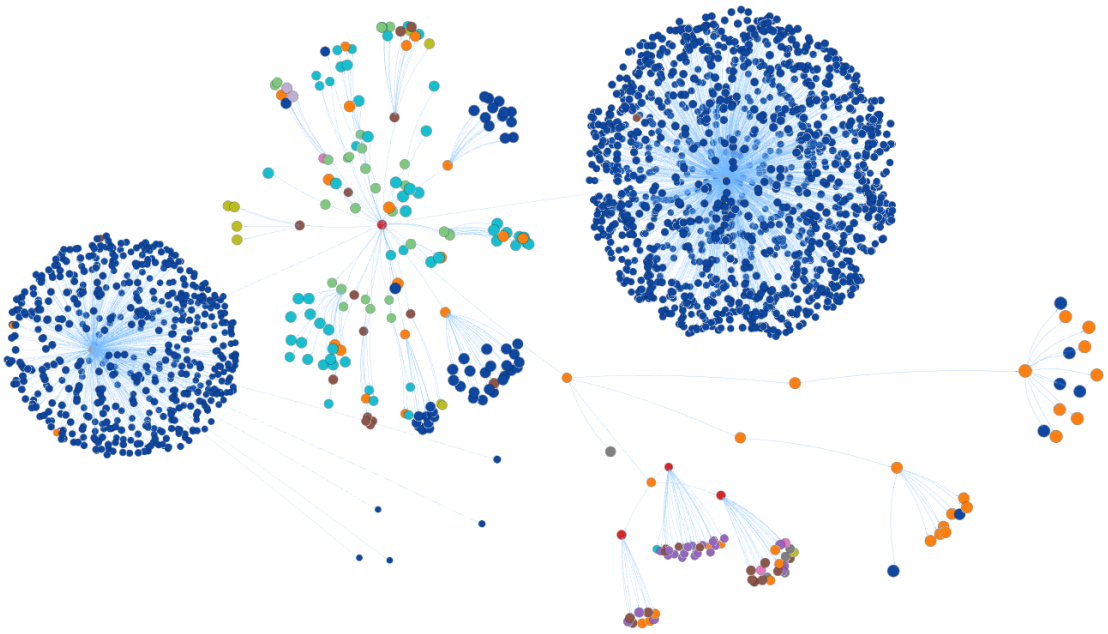

Investigation Of Phylogenetic Relations Using Graph Data Science Algorithms

Master's thesis in Data Science and AI

Ankita Rahavachari & Guru Prakash Subramanian

Department of Computer Science and Engineering
Data Science and AI Master's Program
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

Master's thesis 2021

Investigation Of Phylogenetic Relations Using Graph Data Science Algorithms

Ankita Rahavachari
Guru Prakash Subramanian

Department of Computer Science and Engineering
Data Science and AI Master's Program
Chalmers University of Technology
Gothenburg, Sweden 2021

Investigation of Phylogenetic Relations Using Graph Data Science Algorithms
Ankita Rahavachari & Guru Prakash Subramanian

Supervisor: Dr. Hao Wang, Department of Biology and Biological Engineering
Examiner: Marina Axelson-Fisk, Department of Mathematical Sciences

Master's Thesis 2021
Department of Computer Science and Engineering
Data Science and AI Master's program
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Graph Visualization of approximately 2000 nodes representing the evolutionary biological dataset from NCBI using GraphXR

Typeset in LaTeX
Gothenburg, Sweden 2021

A Chalmers University of Technology Master's thesis template for LaTeX

ANKITA RAHAVACHARI
GURU PRAKASH SUBRAMANIAN
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Driven by the vast amount of fast-growing biological databases, a total of 2.1 million diverse species have been categorized within the NCBI taxonomy database either by DNA, RNA, protein, or genome sequences. This thesis focuses on performing a comprehensive analysis of the classified taxonomic branches and nodes in the taxonomy database through utilizing various graph algorithms. By converting these taxonomy data into a Neo4j database, a super graph with 2,121,053 unique branches and 2,323,131 intermediate and end nodes was obtained in a rooted tree structure. In contrast to the classic Linnaean system with eight major ranks (from domain to species), there are 37 additional taxonomic ranks that have been used in describing the complicated phylogeny of the accumulated species. Surprisingly, nearly 10% of the taxonomic nodes are found with a rank either "norank" or "clade" that remain unclassified and await for systematic assignment. In addition, incomplete investigation of skipping cases of taxonomic ranks revealed thousands of lineages that lack one more major rank. They are deviated from the classic taxon hierarchy defined in the Linnaean system, which appears lagging behind the pace of current biological advancement and should be revisited for upgrading. Finally, a bioinformatic tool for estimating phylogenetic distance between any two given organisms was developed and provided with a graphical interface for user exploration.

Keywords: Data science, Neo4j, Graph Analytics, GraphXR, Neo4j browser, Cypher, NCBI, Phylogenetics

Acknowledgements

First of all, we would like to express our heartfelt gratitude to our supervisor Dr. Hao Wang who has been supportive and motivating throughout the thesis work. His interest in this topic, as well as his knowledge and eagerness to discuss ideas, and advising changes in writing have been invaluable to us. We highly appreciate his efforts that made us to complete this master thesis. We could not have asked for a better supervisor.

We would also like to thank Marina Axelson Fisk, our Examiner for her support and directions which helped us to structure the thesis writing.

The resources provided by the Swedish National Infrastructure for Computing at C3SE has been used for computations in this project.

Finally, we both would like to thank our parents who supported us in every way to ensure that we could achieve our dream.

Guru Prakash Subramanian & Ankita Rahavachari, Gothenburg, August 2021

Contents

List of Figures

List of Tables

Acronyms

ACID Atomicity, Consistency, Isolation, and Durability make up the properties of a Database System. 2
APOC Awesome Procedures On Cypher. 4

CSV Comma-Seperated Values. 4

DBMS Database Management System. 2

GDS Graph Data Science. 4
GUI Graphical User Interface. 14

NCBI National Center for Biotechnology Information. 1

SQL Structured Query Language. 3

WCC Weakly Connected Components. 15

Glossary

Clause
In coding terminology, 'Clause' is a sequence of keywords that performs the intended action. 3

Cypher Querying
Cypher querying pertains to manipulation and access to Graph Database in Neo4j. 2

Edge
A line that denotes a relationship or a path from one node to the other. 1, 3

Graph
A collection of edges and vertices. 1

GraphXR
A plugin tool for Visualization of Graph Database. 4

Label
Denotes the Identity of a node in the graph. 2

Neo4j
Graph Database. 1

Node
A vertex in the graph containing attributes. 1

Plugin
A tool that complements the original software for performing added functionalities. 2, 5

Property
An attribute of a node that denotes how the action should be performed . 2

Relationship
An association that defines a connection between two nodes is denoted as an edge. 2

Taxonomy
A scheme of scientific classification. 1

1
Introduction

The growth of data in the field of biology is enormous and it becomes a challenge to manipulate the huge amount of data efficiently. This drives the biological research towards contributions from data science and machine learning primarily for analysis and building complex models to manipulate the available data [22]. This project is reliant on the NCBI Taxonomy database that includes indexes of categorized organisms which have at least one nucleotide or protein sequence [18]. The taxonomic position of an organism can be viewed or retrieved using a taxonomy browser [5]. Considering the efficiency and performance of this browser new tools such as Taxonkit [27] and Taxallnomy [25] were developed recently that included the functionalities of exploring the taxonomy data using Go programming and generating hierarchically complete taxonomy tables from the NCBI database using customized algorithms.

Even though the above tools provide efficient manipulations on the NCBI database, the analysis related to the visualization of the whole taxonomy tree, and its statistics were missed. While improving the ways to visualize the dataset, there is an urgent need to analyze the existing available dataset to derive meaningful insights and understand how different organisms are related. In this project, a batch of NCBI taxonomy datasets released in April 2021 were utilized to conduct a comprehensive analysis of the taxonomy tree as well as a detailed explanation of the relationships and organisms involved. The analysis of huge NCBI taxonomic data relied on the use of the Neo4j Graph Database which exhibited results without compromising the run time efficiency and memory consumption. Towards the end, Neo4j and Python programming are integrated to create a Graphical User Interface that acts as a toolbox to trace the path/relationship of any two given organisms.

Here, the technical and biological concepts required for this project are explained in detail.

1.1 Graph Database

A Graph is generally composed of nodes and relationships between them. A Node represents a vertex and a relationship represents the Edge connecting two nodes. For example, in a traveling salesman problem [24], every node represents a city and the directed edge represents the distance between the two cities. Graph data thus is defined as complex data which can be represented in a table format with one-to-one,

one-to-many, or many-to-many relationships.

Unlike other Database Management Systems (DBMS), the Graph Database, also known as NoSQL databases, is based on Relationships rather than foreign keys. It explicitly supports querying on a data network and allows us to work with more expressive data such as phylogenetic tree structures [24]. These databases are highly adaptable to constantly changing real-world data. Another significant advantage of Graph Databases over relational databases is their ability to perform relatively consistent high scale join-query operations even as the dataset grows. This is because the operations are limited to one section of the graph, thereby increasing efficiency and performance [24].

There are several Graph Databases products available, such as TigerGraph, Amazon Neptune, Neo4j, Cassandra, and Datastax. Among these, the Neo4j Graph Database was preferred due to its enormous support of graphs scaling to unlimited nodes and relationships [2]. Other primary advantages include fast read and write performance without compromising the data integrity. It is the only Graph Database that provides ACID compliance to ensure the predictability of relationship-based queries, scalable architecture to optimize the speed, and native storage as well [14]. While working on large datasets, Neo4j is the go-to option due to its loading speed of huge size of data with low memory footprint. Apart from the above advantage, Neo4j also offers amazing plugins which indeed make the work of an Analyst easier than any other Graph Database. The in-depth explanation of Neo4j, Cypher Querying, Neo4j Plugins, and Neo4j Graph Data Science Algorithms can be found in section [1.2].

Previously the graph models were used to exploit the complex cell biology networks and help to focus on the local or global partition of the graph to derive insights into the characteristics of cellular networks for understanding of these complex structures and treatment development for cancer [21]. In this project, we aim to conduct comprehensive analysis of the existence of organisms, their hierarchy, and investigation of their taxonomic relationships with categorized ranks.

1.2 Neo4j Graph Database

In this section, we will look into the in-depth details of the Neo4j Graph Database. It is a software whose enterprise edition can directly be downloaded from the Neo4j official webpage [6]. Rather than analyzing data as a table, this software allows you to analyze the data as a network of relationships. This is of predominance when it comes to biological databases. Neo4j is a Labelled Property Graph Model. It, therefore, has the following characteristics [24]:

- Existence of Nodes(entities) and Relationships(named edges).
- Every Node contains properties (key-value pairs).
- Nodes can have one or more Labels.
- Relationships can also have a Name, Property, and are directed.

- Every Relationship will have a source and destination node.

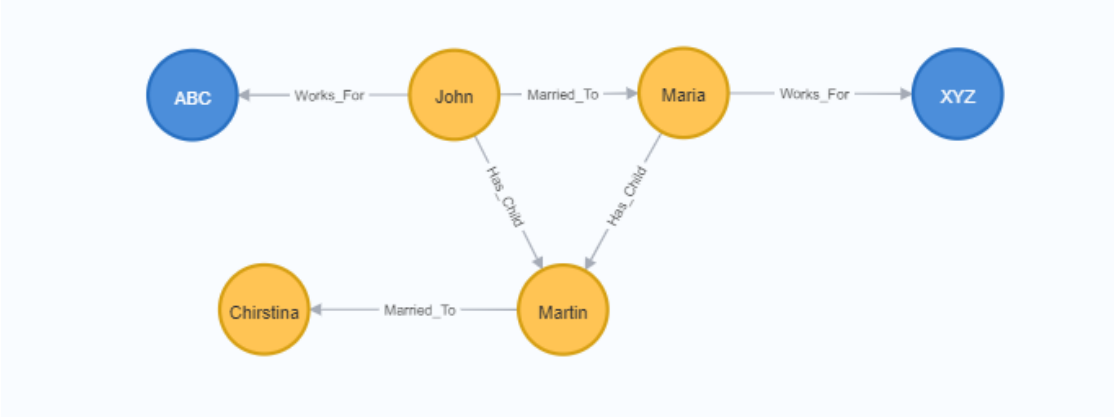

Figure 1.1: A simple family hierarchy of Neo4j graph model with nodes, labels, and relationships

The nodes are assigned a similar color based on the label they belong to. There are two labels namely Organization denoted by blue color and Person denoted by orange color associated with Figure 1.1. ABC, XYZ are organizations in which John, and Maria work. Likewise, Christina is married to Martin, who is the child of John and Maria. The name of the relationship is mentioned on the Edge joining the two nodes.

To query the graphs in Neo4j, Cypher scripting language can be used. It is the standard Neo4j querying language. According to Ian Robinson, this graph query language is easy to learn and helps us to infer knowledge about graphs [24]. It follows a similar structure to SQL and has the Clauses listed in 1.1.

Cypher Clause	**Definition**
CREATE	Used for creating nodes and relationships
MATCH	Used for searching specific patterns in the whole database or a portion of the database
MERGE	To check if a pattern already exists in the graph. If it doesn't, then that pattern will be created
WITH	Enables the user to add the second part of the query to an existing cypher query to make a pipeline
WHERE	It is used along with a MATCH clause to add a constraint or used to filter the results of WITH clause
RETURN	This clause helps the user to display the parts of the results they are interested in. It can be nodes, relationships, or a path between two nodes

Table 1.1: Cypher Clauses with their definition

Cypher queries are case insensitive. Every time a clause is used it is highlighted automatically by the Neo4j Browser Code-check functionality. In Code 1, the query gets all the relations that are directly connected to the node John. Since we store the results in a variable called 'p' and return it, the output of this query will return a graph containing all the paths directly connected to node 'John'. The result of this query is represented in 1.2.

```
1 MATCH (:Person { name:"John" } )-[ r ]-( )
2 RETURN r
```

Listing 1: Cypher Query to return all the relations that are directly connected to the node John

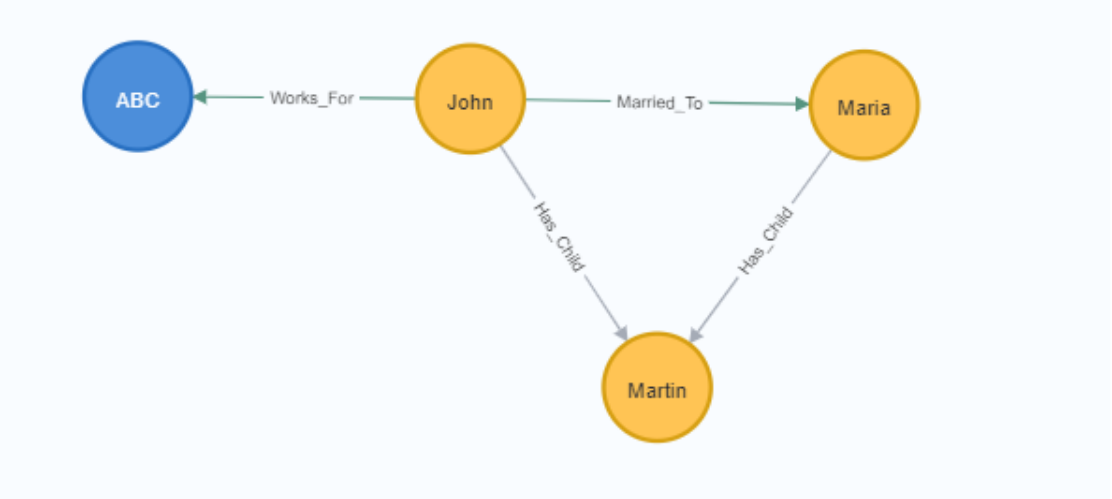

Figure 1.2: Nodes with Relationship representation

From Figure 1.2 it can be observed that the query has also returned the connection between Maria and Martin. This is due to the fact that the node Martin is directly connected to both John and Maria.

The above-listed clauses in Table 1.1 are the basic clauses available in Cypher Query Language . Apart from these, there exist a plethora of clauses that can be used based on the user's requirement. Neo4j Platform has built-in APOC procedures that can be used as part of Graph Data Science (GDS) analysis. These procedures also enable the user to stream the cypher results as a tree structure, download the results in CSV format and help to bridge the connection between applications,like Neo4j and GraphXR , the latter allows convenient generation of visualizations.

1.3 Graph Analytics and Neo4j GDS Algorithms

Next is to introduce how to apply Graph Analytics or the GDS Algorithms to the data to get meaningful insights. Graph Analytics is closely related to the statistics

of the graph. It allows inferring relationships from the data. This includes basic information of the graph such as the mean, median, and standard distributions of relationships and nodes [19]. The Neo4j Graph Database enables the user to perform more complex querying by deploying Data Science Algorithms on the graph. The development of GDS serves Data Scientists to utilize this functionality to visualize the structural information of a graph. GDS therefore has huge potential in the fields of healthcare, marketing, and fraud detection [19].

GDS Algorithm comprises three major types including Path Finding, Community Detection, and Centrality Algorithms. In this project, we focus on using the first two types as it is more relevant to the taxonomy database. The explanation and usage of the algorithms are described in section 2.3. Neo4j on the other hand also has a Plugin called 'Graph Data Science Playground' which provides automated 'Recipes' of the GDS algorithms. This functionality is provided for people who are unfamiliar with Cypher Scripting but still want to perform analysis with the results derived from this plugin.

1.4 Graph Visualization Plugins

Graph rendering is a memory-intensive process for just displaying a few nodes and relationships. The Neo4j platform offers multiple plugins for the users to explore the graph for visualization such as Neo4j browser, Neo4j Bloom, GraphXR, and Gephi API.

1.4.1 Neo4j Browser

The Neo4j Browser [8] is a Neo4j Plugin that facilitates Visualization and Cypher querying functionality. The browser can be invoked when an active-graph database connection is present. This browser offers Result Frames that display the output for an executed Cypher query. It can also present the output in varied formats such as a graph or exportable JSON/CSV format depending on the Cypher query. However, it has an upper limit on the number of nodes and relationships it can display at a time which is set to 300 nodes by default.

1.4.2 Neo4j Bloom

Neo4j Bloom [7] is a graph exploration tool that allows the exploration of nodes and relationships in an interactive way which can also be installed via the Neo4j Graph App tool. Bloom can display up to 10,000 nodes but is limited by the layout customization.

1.4.3 Gephi API

Gephi API [3] is a separate plugin tool that can be connected with Neo4j. Cypher querying is used to stream nodes and relationships from Graph database to this software. Though it supports the display of more nodes when compared to Neo4j

Bloom, it lacks a Hierarchical layout tool that helps in understanding the topology of tree structure.

1.4.4 GraphXR

GraphXR [4] is a 3D plugin that connects to the Graph database and displays the nodes and relationships through the cypher query execution. It has a wide variety of customization tools in its arsenal that help in visualizing graphs effectively. For instance, the cypher query to return the path from the root node to subsequent depths can be displayed as a hierarchical tree with the ranks coded in varying colors that enable easy identification. Similar GraphXR visualization of a simple family database can be seen in Figure 1.3. Also, GraphXR is a dynamic visualization tool that allows for the exploration of an individual node to its subsequent connected relationships. The primary advantage that is available only in GraphXR is the functionality of snapshots that enables the user to save the current customized graph layouts and quickly retrieve them when required.

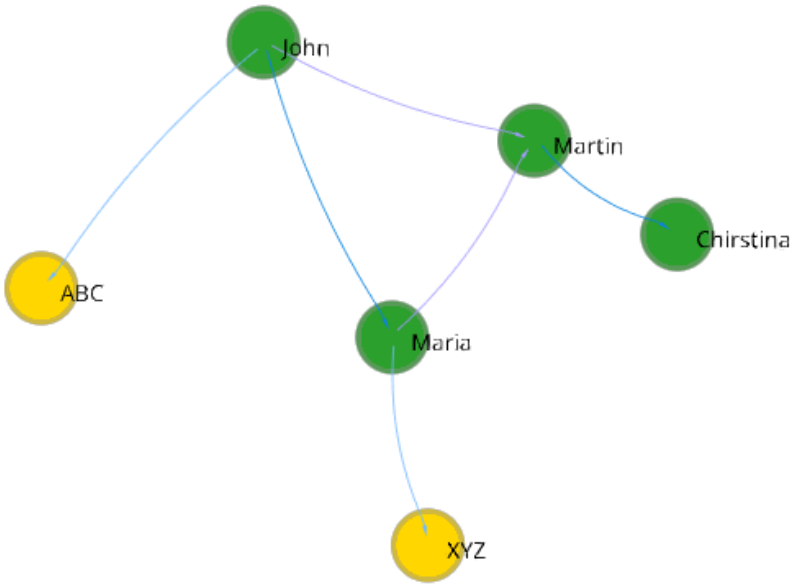

Figure 1.3: GraphXR Visualization

1.5 Phylogenetic Study

Biodiversity is considered to be one of the most interesting phenomenons on earth. According to research conducted and published by the Census of Marine Life [10], it was estimated that our species (homo sapiens) share the planet with 8.7 million distinct types of plants, microbes, and animals. Species and genes that have evolved over numerous generations each have distinct features that allow us to categorize

them as unique organisms. These organisms can replicate themselves by reproducing nearly identical copies, which however may experience significant adaptive changes during the long evolutionary history.

The phylogenetic study can be defined as the study of the development of relationships between different organisms [30]. It answers interesting research questions such as what is the historical relationship between two species or genes, and how a particular sequence evolved. Apart from its crucial role in research, this study is also considered by forensics to provide crucial evidence based on DNA samples during the court sessions [17]. Recently, biologists have used the phylogenetic study to theorize that whales are closely associated with hoofed animals like Hippopotamus and ruminants such as cows [12]. The phylogeny is often represented in the form of a tree structure 1.4. It is also worth noting that only 13.7 percent of the 8.7 million species [28] have a name and their place (a.k.a Rank) in the phylogeny so far. In this thesis, the distance estimation method is used to study organisms that are currently identified. Findings related to this study are described in 3.3 and 3.7.

It is impossible to study millions of organisms and their history without a system to define the classifications. In the 1700s, a Swedish botanist, zoologist, and taxonomist **Carl Linnaeus** devised a classification system [15]to group the organisms that share the common traits into different hierarchical levels. The largest group of similar taxa are called Kingdom and are considered to be the most diverse category. Likewise, the smallest group is known as Species. The other major lineages in between include phylum, class, order, family, and genus. They are known as hierarchical ranks to which an organism belongs.

Due to the exponential discovery of new species on earth, a fundamental revision to the Linnaean Classification of taxons was made and a new taxon named Domain was introduced. This is even larger and more inclusive than the Kingdom. To date, Bacteria, Archaea, and Eukarya are the only three domains of life on earth [29]. Eukarya includes humans and other animals. Figure 1.4 represents the phylogenetic tree of relations present in three domains.

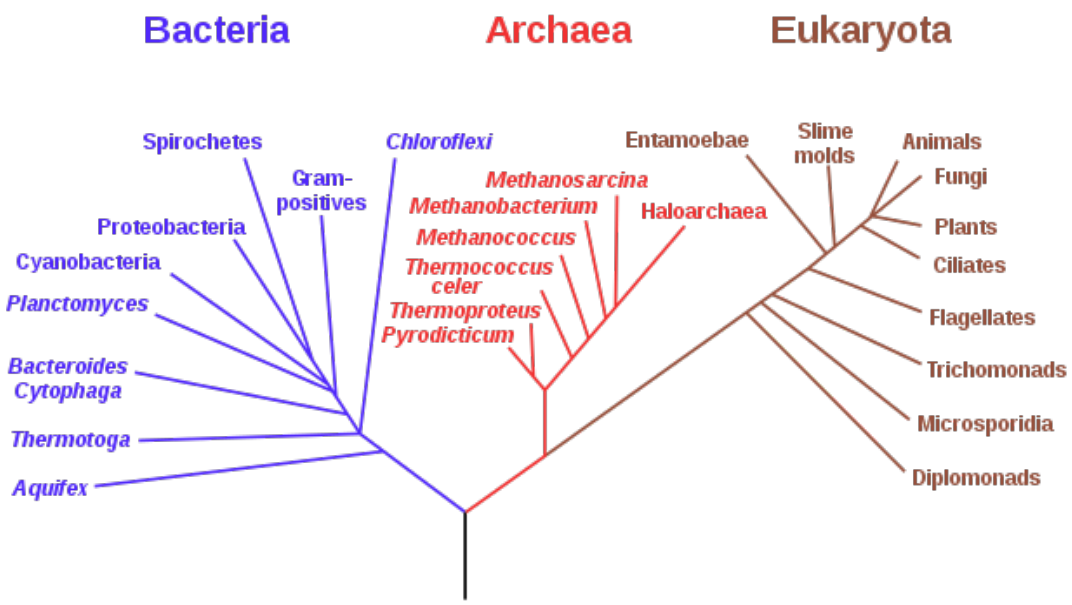

Figure 1.4: Phylogenetic Tree of Life, Credits: Wikimedia Commons

The recent update made in 2020 by the NCBI curators [26] added a new domain-level branch of**"Viruses"**and the term "domain" was replaced as "superkingdom".

1.5.1 The Taxonomy database from National Center for Biotechnology Information

This project is primarily reliant on the National Center for Biotechnology Information (NCBI) Taxonomy database. The NCBI is responsible for the nomenclature of the organism names. NCBI curators update the repository by adding names after a verified publication or approval [26]. This repository consists of the names of organisms and their classifications for every sequence in the nucleotide and protein sequence database of the International Nucleotide Sequence Database Collaboration. In the NCBI taxonomy database, the root node is parted into four superkingdoms: Eukaryota, Bacteria, Archaea and Viruses. Each "superkingdom" has child nodes from succeeding ranks onward.

2
Approach

This chapter explains the implementation procedures used to obtain the results. The first section below describes the data extraction process which is followed by the Neo4j node creation and explanation of Cypher queries that were used to analyze the data. Then the Graph data science algorithms used in this project are illustrated in detail followed by the graph visualization procedures and Neo4j-Python integration to create a graphical user interface. The procedures to conduct such analysis on the taxonomic dataset and the source code to reproduce these analyses can be found in the Github Repository [23].

2.1 Data extraction from NCBI database

Initially, the dump files "names.dmp" and "nodes.dmp" were downloaded from the NCBI FTP webpage [20]. The columns tax-id, parent_tax_id, rank, scientific_name, and common_name were extracted using Python programming. The extracted columns are then converted to CSV format that facilitates data loading in Neo4j Database. The extracted dataset consists of approximately 2.3 million rows of which the first 20 rows are shown in Figure 2.1. The 'tax_id' column consists of the unique ID given to each organism by NCBI [26], 'parent_tax_id' represents the id of the organism's parent and there is repetition of the same id in this column because one parent can have more than one child. The 'rank' name indicates the rank to which the organisms belong, 'scientific_name' and 'common_name' are the names assigned to the organism by which they are known. The nodes that do not have a common name are replaced by Not Available (NA). Due to the constraint employed in section 2.2, it is necessary to add a parent '0' to the Root node in the original dataset depicted in Figure 2.1 before loading the CSV file into the Neo4j database.

The data can be interpreted as, for example, tax_id 1 has parent 0 and belongs to norank category with scientific name root and common name NA. Every row in this dataset forms a single parent-child relationship. This CSV file is imported to the active database created in Neo4j Desktop. One should install the required plugin and configuration files before proceeding to the Neo4j browser window.

2.2 Neo4j Implementation

A database server is configured in the Neo4j Desktop which is capable of storing the CSV file data. Plugins such as Graph Data Science, APOC, and GraphQL are

tax_id	parent tax_id	rank	scientific_name	common_name
1	0	norank	root	NA
2	131567	superkingdom	Bacteria	eubacteria
6	335928	genus	Azorhizobium	NA
7	6	species	Azorhizobiumcaulinodans	NA
9	32199	species	Buchneraaphidicola	NA
10	1706371	genus	Cellvibrio	NA
11	1707	species	Cellulomonasgilvus	NA
13	203488	genus	Dictyoglomus	NA
14	13	species	Dictyoglomusthermophilum	NA
16	32011	genus	Methylophilus	NA
17	16	species	Methylophilusmethylotrophus	NA
18	213421	genus	Pelobacter	NA
19	2812025	species	Syntrophotaleacarbinolica	NA
20	76892	genus	Phenylobacterium	NA

Figure 2.1: View of first 20 nodes of 2.3 million rows in the NCBI dataset

installed using the options available on the desktop before starting the database. Once everything is set up, the user can start the database and open the Neo4j Browser window to start Cypher Querying.
A constraint on tax_id was first introduced to enforce that every tax_id is created only once. Cypher constraints come in handy and ensure that there is no redundancy while creating nodes in Neo4j. Then the dataset is loaded into Neo4j Browser in the same order as mentioned in 2.1. Once loaded, nodes will be created with the properties described in Figure 2.1 as shown in Figure 2.2 , which also represents the visualization of nodes before the parent-child hierarchical relationship gets created.

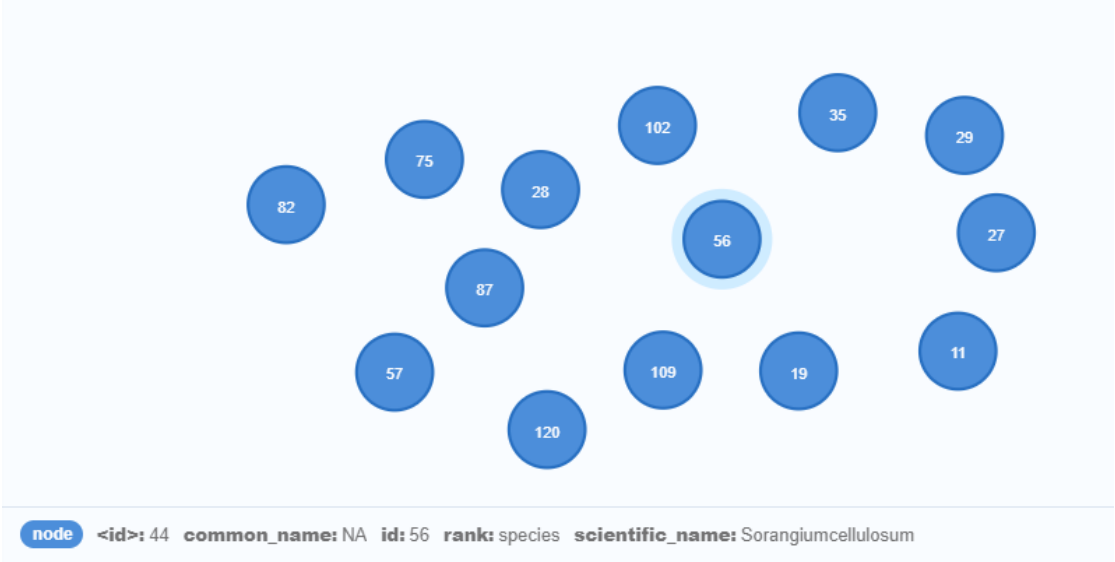

Figure 2.2: Representation of nodes along with the properties of the highlighted node

The next step is to create relationships between the existing nodes based on the parent-child association. Every row in the CSV file represents a direct one-to-one re-

lationship. Therefore we matched the tax_id's with the parent tax_id's and created a relationship called 'HAS_CHILD'. This denotes that a (parent)-[:HAS_CHILD]->(child). This relationship eventually forms the required phylogenetic tree structure. Code 2 shows the Cypher query used to create the relationship 'HAS_CHILD'.

```
:auto USING PERIODIC COMMIT 5000 LOAD CSV WITH HEADERS
FROM 'file:///NCBI_DATA.csv' AS row
MATCH (p:node {id: toInteger(row."parent tax_id"),
(c:node {id: toInteger(row."tax_id")})
MERGE (p)-[ :HAS_CHILD ]->(c)
```

Listing 2: Cypher Query to Create Relationships

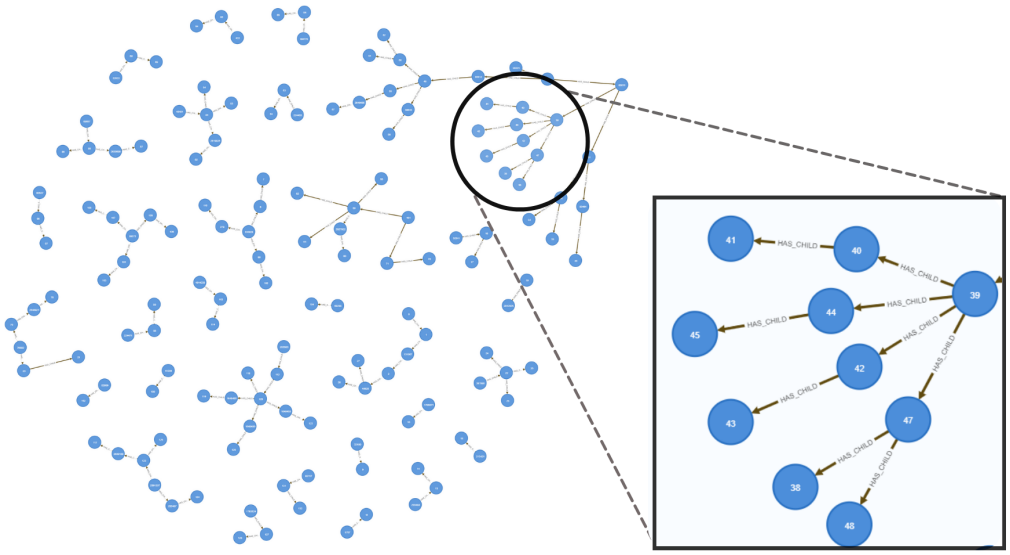

Figure 2.3: Taxonomic nodes with relationships in Neo4j Browser

Note: Figure 2.3 shows an example of how the nodes (n=100) along with relationships look like in the Neo4j Browser.

2.3 Neo4j Analysis - Graph Data Science

The Neo4j GDS includes various algorithms which when applied to large amounts of data can yield interesting insights. As the thesis relies on the NCBI taxonomy dataset containing hierarchical information of the organisms, the suitable selection of an algorithm that makes biological sense for this analysis is crucial. Based on our project, we choose to use two GDS algorithms that are efficient and effective with taxonomic data.

2.3.1 Weakly Connected Components (WCC) - Community Detection Algorithm

When working with a dataset of 2.3 million rows, preparing the data for analysis can be time-consuming. In Neo4j, however, the fact that the relationship between two nodes is prevalent can be used. This will eventually assist in answering fundamental concerns about the graph connectivity and isolated clusters in the original dataset. The built-in WCC algorithm focuses on the graph's connectedness. For this algorithm to work, there must be a unidirectional path between each node [13], with which the taxonomy data comply Figure 2.3. This algorithm operates on the principle of determining whether each node is connected and whether a node can be reached from another node in the same set.

The Cypher Query 3 to run the WCC algorithm on our dataset is given below

```
1 CALL gds.wcc.stats("WCCgraph")
2 YIELD componentCount
```

Listing 3: Cypher to run WCC algorithm

In order to preserve the original graph, a copy named 'WCCgraph' is used for the WCC analysis.The above Cypher returns the parameter given along with YIELD. The 'componentCount' is the measure of total graphs in our dataset.

2.3.2 The Shortest Path - Path Finding Algorithm

The path-finding algorithms were used to explore the connectivity of nodes by providing a particular source and target node information. This in turn allows the user to check if there exists a path between these two nodes. In this project, the Shortest path algorithm is used to find the relatedness between two organisms. This algorithm finds the shortest path from the given source and destination node. If there is more than one path from the source to destination the algorithm prioritizes the path which has the fewest edges. The Graphical User Interface developed in section 2.6 uses the shortest path algorithm.

2.4 Neo4j - Graph Visualization

The Graph XR tool constitutes a cypher query functionality to retrieve nodes and relationships from the active database connection. For instance, the cypher query to return the path from the root node to subsequent depths can be displayed as a hierarchical tree with the ranks coded in colors that enable easy identification.
Figure 2.4 gives an example view of GraphXR interface. The nodes are color-coded according to their 'rank' property. Layout functionality is used to obtain a hierarchical tree structure that enables better inference of the graph. We can observe the presence of other ranks such as clade, and no-rank which will be discussed in section 3.3.

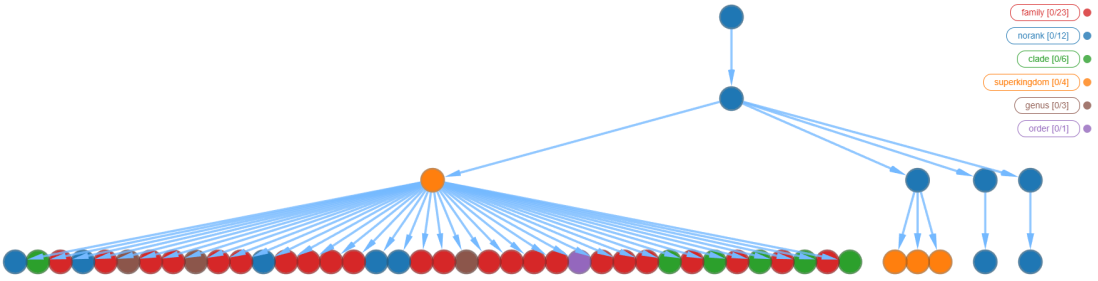

Figure 2.4: Visualization of the phylogeny using GraphXR

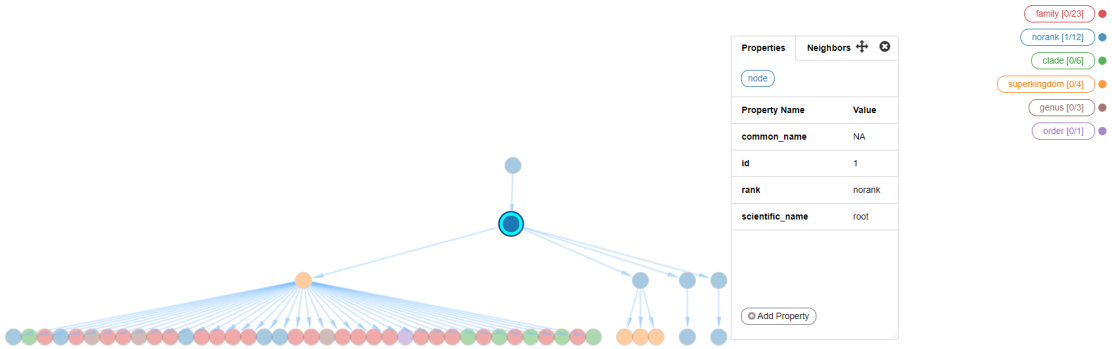

Figure 2.5: Displaying the node properties in GraphXR

Figure 2.5 displays the details of a particular node when it is selected.The Node-search functionality in GraphXR helps in identifying a node of interest by its 'id' or 'rank' property. Nodes can be analyzed on the tree structure by expanding the individual nodes to inspect their child nodes, and also learn the properties of each node by clicking and opening a pop-up.

2.5 Comprehensive analysis on NCBI dataset

The NCBI taxonomy database consists of such huge taxonomic classifications that are difficult to grasp. Thus a comprehensive analysis was conducted and statistics related to taxonomic ranks were generated by utilizing the cypher querying functionalities in Neo4j Browser. The Taxonomic graph is further analyzed by finding its most populated node that has the mostly dense child nodes. Then the graph is further explored by traversing to the lower ends for studying their composition among the most populated ranks. The identification of these nodes that act only as a child was investigated using the Cypher querying functionality in Neo4j. During this traversal, some exceptional cases occurred in which major taxonomic ranks were skipped. These cases were explored to gain insight into current taxonomic classfications.

For each of the above analyses, the CSV files were generated using cypher querying in the Neo4j Browser and their respective charts were created using Tableau Public

Software [16].

2.6 Neo4j-Python Integration:GUI

This section explains in detail the integration of Neo4j and Python. The integration acts as a platform on which the GUI 3.7 will depend. The Neo4j Graph Database can be accessed using the **Python Graphdatabase Driver** package [9]. This package facilitates access to the Neo4j server from Python-based queries submitted with the URL and password credentials, which are verified for establishing a connection. Once the connection is generated, the Cypher query to find the shortest path between two nodes with the intermittent node information is embedded into python. The result of the cypher query is initially stored as a dictionary which helps for quick key, value pair access.

The GUI is implemented using the available **PySimpleGUI** [11] python package. This package helps in creating a customizable layout containing several widgets that display node information based on the user input. The goal of this GUI is to find the phylogenetic distance between any two achieved nodes.

3

Results

3.1 Overview of the NCBI taxonomy database as a tree structure

Figure 3.1 is the overview of the phylogenetic tree that contains approximately 600 taxonomic nodes. Each node was assigned a color based on its rank. This tree was created as described in section [2.2]. To verify the connectivity of the whole taxonomic graph, the WCC algorithm was used. The number of clusters after the WCC algorithm was run obtained "1" as a result. This proved that a single complex graph encompassing a total of 2,323,131 nodes was observed without any isolated clusters.

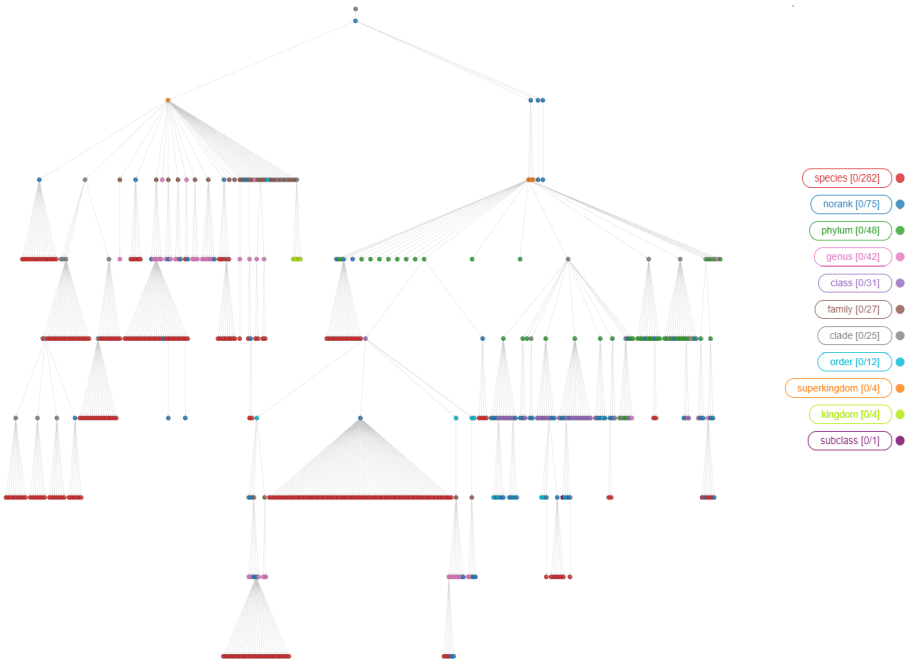

Figure 3.1: Overview of the NCBI Taxonomy in tree structure representing approximately 600 nodes

3.2 Evolutionary Ranks

The phylogenetic tree structure in 3.1 was further explored by analyzing the composition of the evolutionary ranks that comprise the taxonomic tree structure. Based on the NCBI taxonomy database, the following result was derived that comprises a total of 45 ranks 1.1. It is observed that among the 2.3 million taxonomic nodes, the category “species” dominates the NCBI dataset with 1,896,462 entries followed by the “norank” category with 223,497 entries. Figure 3.2 represents the chart showcasing the count of each major rank based on the generally used Linnaean system. The information of complete rank names, as well as their counts in the dataset, is given in Appendix A.

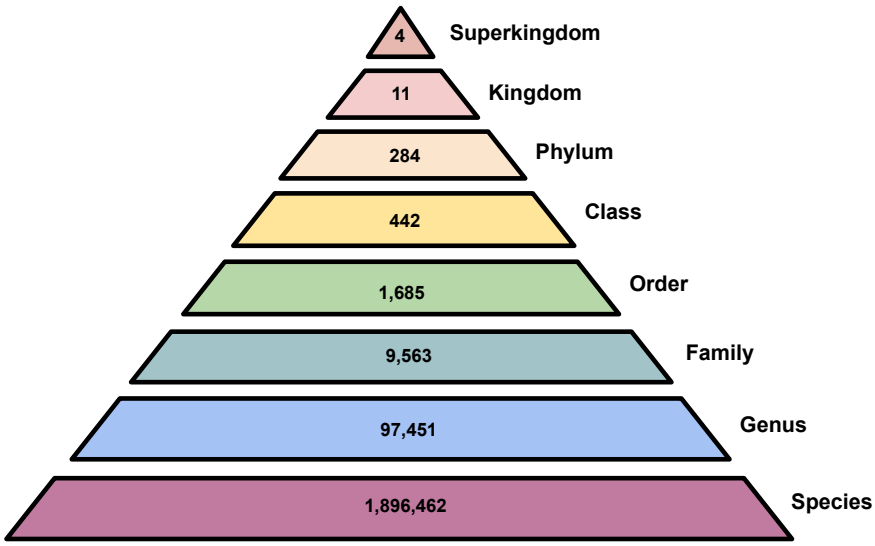

Figure 3.2: Pyramid graph statistics of the successor rank lineages

3.3 The “norank” and “clade” rank types

In the NCBI taxonomy database, many nodes are not assigned with ranks properly. Therefore those nodes were referred to as “norank” and “clade” rank types [21]. The description of these two rank types are discussed in detail in section [4.3]. The “norank” nodes constitute 8.85% (223,497) of the total nodes while the “clade” rank nodes account for just 0.03% (894). On further investigation, the distribution of “norank” nodes among the lineages was derived in Figure 3.3. The majority of the "norank" nodes accounting for 42,441 entries is found as a child node of “genus” rank and the second-largest habitat of norank nodes is under the “species" rank.

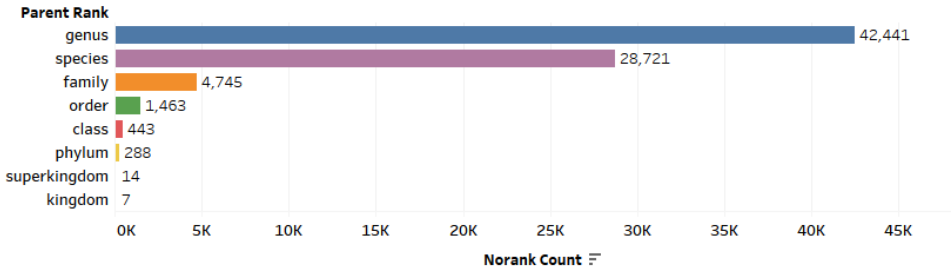

Figure 3.3: Norank distribution among lineages

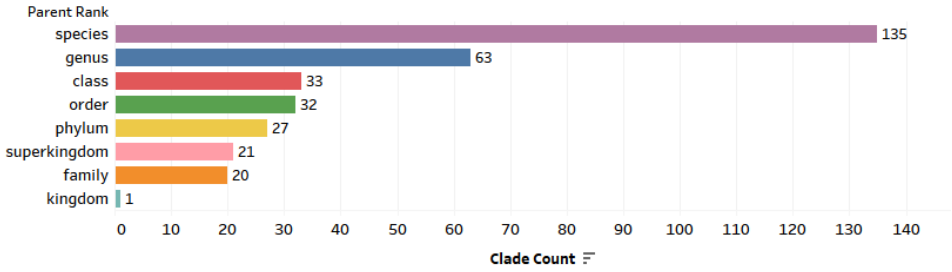

Figure 3.4: Clade distribution among lineages

A similar analysis of the distribution of "clade" nodes among the lineages resulted in 3.4. When compared to the "norank" classification the "clade" rank nodes are significantly less in number. They are prominently seen as a child node of the "species" rank accounting for 135 nodes. Also, they are evenly dispersed under the "class", "order", "phylum", "superkingdom" and "family" ranks.

3.4 Identification of dense clusters in the taxonomic tree

The presence of dense clusters in the taxonomic tree was detected during the visualization, indicating the existence of parent nodes with a rich number of child nodes. Figure 3.5 represents the packed bubble chart representation of the dense cluster statistics for the entire database. The size of the bubble varies based on the count of child nodes in each organism. One such organism with the scientific name **H1N1 subtype** and rank **"serotype"** has 36,125 child nodes. They have the highest number of outgoing edges. The node with the scientific name **H3N2 subtype** and rank **"serotype"** has the second-highest number 35,272 child nodes connected to it. The results also showed that organisms under "norank" classification dominated the statistics. Since they remain unclassified and are not assigned a specific name, they are thus excluded from consideration in Figure 3.4.

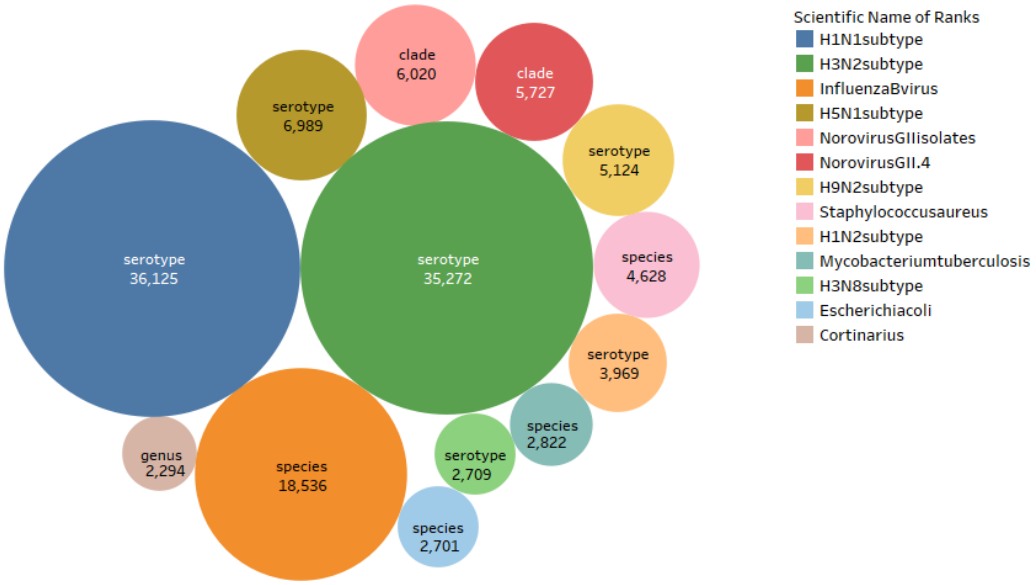

Figure 3.5: Packed bubble chart representation of the statistics of dense clusters in the taxonomic tree

3.5 End nodes in the taxonomy tree

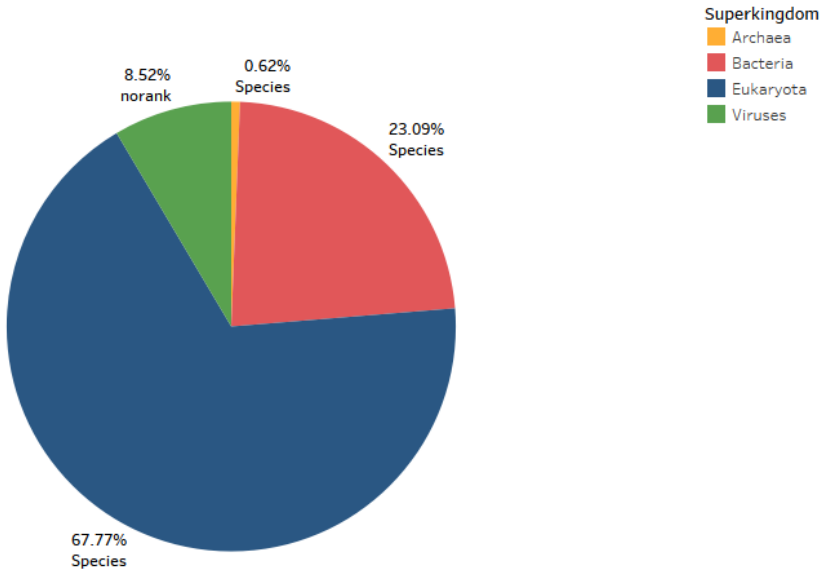

Figure 3.6: Distribution of end nodes and their rank under superkingdoms namely Archaea, Bacteria, Eukaryota and Viruses

End nodes are those that reside at branch ends of the taxonomic tree. They were found to appear throughout various levels in the tree. A total of 2,121,053 nodes were detected. Figure 3.6 was generated to locate regions where these end nodes

were prominent. It shows a pie chart that represents the distribution of end nodes under each of the "superkingdom" nodes that are the highest taxonomic rank in the current classification.

From the Figure 3.6, it was observed that the superkingdoms termed Eukaryota, Bacteria, and Archaea all have the majority of end nodes from the rank type "species". But Viruses have 8.52% of their end nodes from the unclassified "no-rank" category.

There are 11 "kingdom" nodes (Appendix A) in the NCBI dataset. These nodes have originated from 10 "clade" and 1 "superkingdom" rank types. The statistics of end nodes under the "kingdom" rank is given below in Table 3.1. Once again the "species" dominate the end node data.

Kingdom	**Rank name**	**End nodes**
Fungi	Species	156,876
Viridiplante	Species	184,490
Metazoa	Species	956,446
Bamfordvirae	Species	1460
Helvetiavirae	Species	8
Loebvirae	Species	60
Sangevirae	Species	476
Shotokuvirae	Species	2587
Trapavirae	Species	14
Orthonavirae	norank	160,203
Pararnavirae	norank	3,305

Table 3.1: Parent and end count information of kingdom nodes

3.6 Rank Skipping among lineages

During the analysis of ranks and its relationships, it was observed that the taxonomic hierarchy 3.2 of the taxonomic tree can be deviated by skipping the major ranks. Here, we report the occurrence of higher ranks that jump to a lower one by skipping intermediate ranks in the taxonomic tree Figure 3.7.
Figure 3.7 follows the Linnean taxonomic rank hierarchy given in Figure 3.2 for the "Parent Rank" column. It is seen that from "superkingdom" rank, 565 organisms have skipped 5 hierarchical level (kingdom, phylum, class, order, family) and directly have child nodes at "genus" rank, and similarly another 225 organisms had child nodes at the rank "phylum" but this time these organisms have skipped only one intermediate level (kingdom). Also, it is noted that the highest count of rank skipping has happened from the rank type "order" to "genus" by skipping the intermittent "family" rank. A total of 1,141 organisms have followed this format of skipping the family rank.

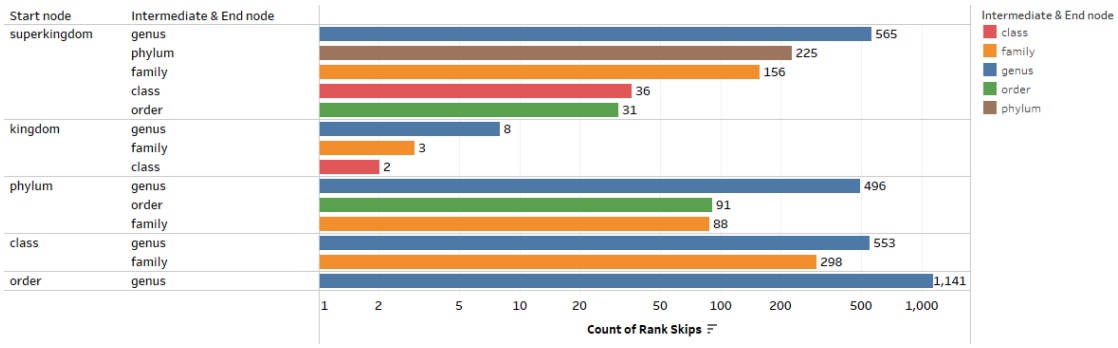

Figure 3.7: Ranks Skipping in the taxonomic tree

Figure 3.7 is considered to be a result of incomplete investigation as it does not include the rank skipping cases from higher ranks to "species". The "norank" and "clade" rank types were excluded from this analysis.

3.7 The Graphical User Interface for Phylogenetic Distance Estimation

This GUI was named **Taxonomy information toolbox**, facilitates in attaining the phylogenetic distance between two given organisms with tax id. The user can enter the source and destination tax ids in the respective text boxes given under the label **Distance Estimation** to calculate the closeness and the relatedness of any two organisms. The **Find Distance** button in Figure 3.8 activates a popup window that displays the distance between the given tax ids if they are present in the taxonomy database. Also, one can find the information of the intermediate nodes by examining the populated table from the source to the destination. The node information comprises the tax id, rank, and scientific name.

In Figure 3.9, the final output of the GUI displaying the information of phylogenetic relations between the two ranks with tax id 2 and 41 can be seen. This result also portrays as an add-on to the results presented in section [3.6]. From "superkingdom", the organism Bacteria has skipped a rank and has a child Proteobacteria form "phylum" rank type.

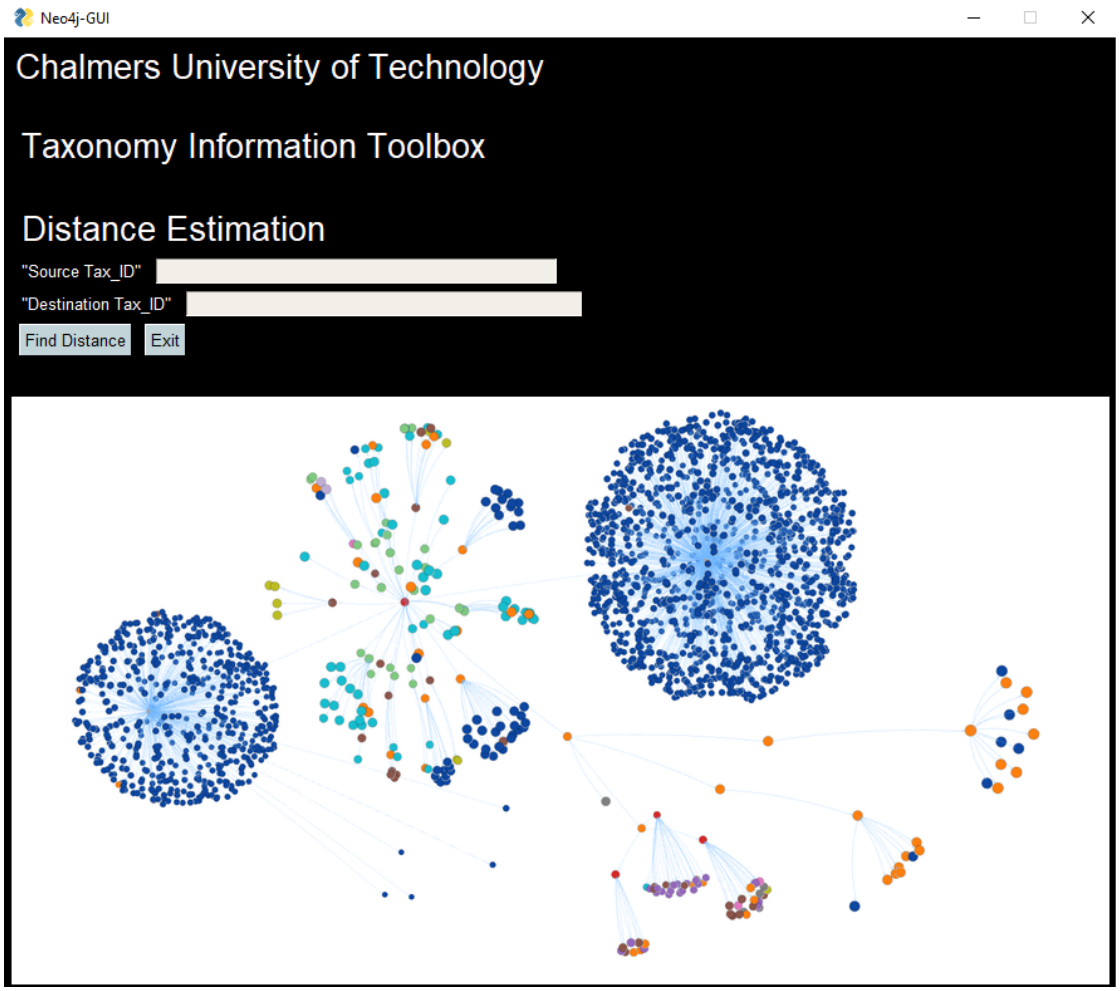

Figure 3.8: A General Layout of GUI

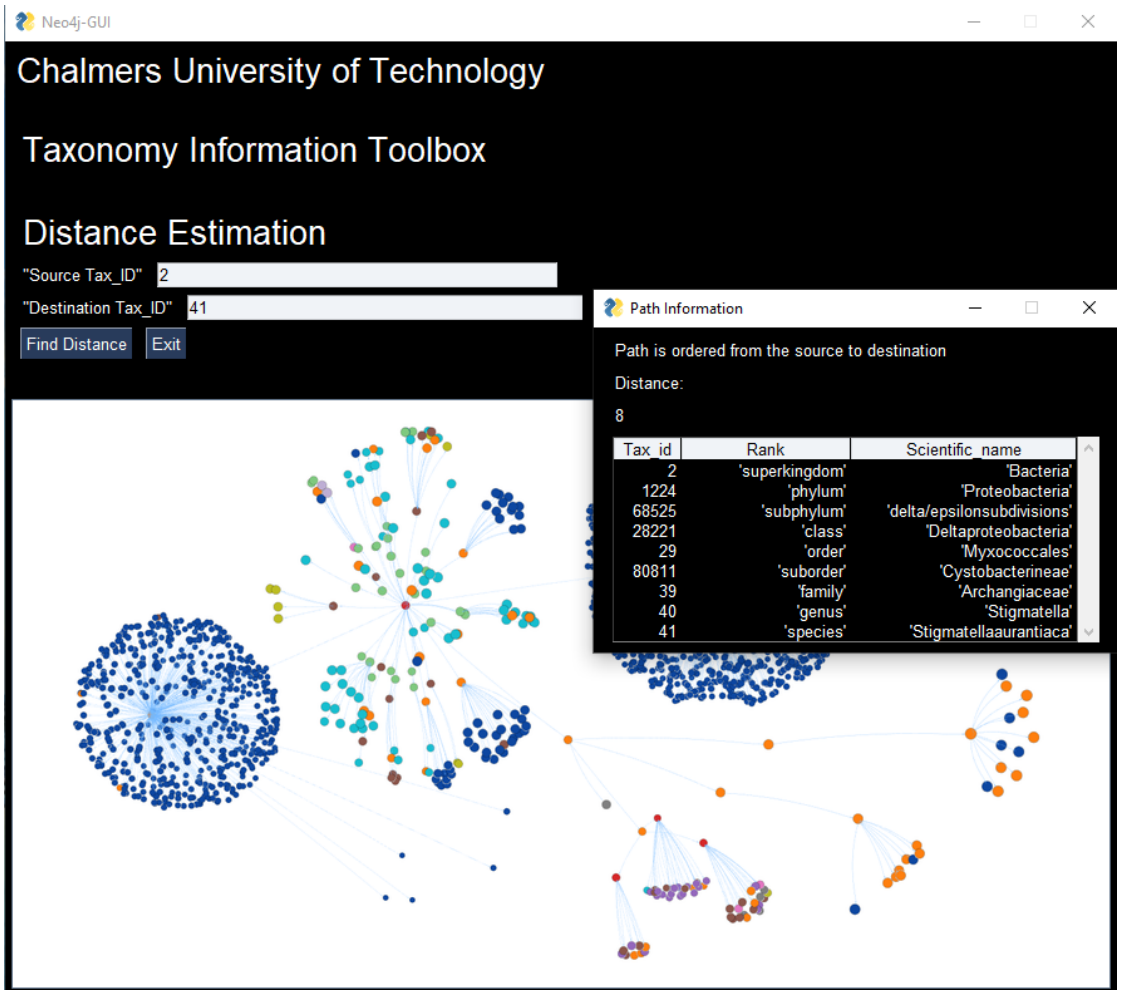

Figure 3.9: The resultant GUI model: Taxonomy Information Toolbox

4

Discussion

In this project, the focus is preferably to conduct in-depth analysis on the taxonomy dataset using Neo4j Graph Database and the Graph Data Science Algorithms, as well as building a simple graphical user interface to facilitate phylogenetic distance inference. In the sections, the results derived in chapter 3 and their applications are discussed in detail.

4.1 Inference of the overview of the taxonomy tree

Although it seems obvious that biological entities should be linked, relying solely on this hypothesis will not yield the desired outcome. As a result, it's fundamental to verify the graph's connectedness. The WCC algorithm discussed in section 3.1 also also confirmed the following statements:

- **Every parent_tax_id can be found under the column tax_id but vice versa is not true:** From section 2.1 one can argue that all the unique IDs present in the 'parent tax id' column can be found in the 'tax id'. This is just another way of interpreting the data extracted from the NCBI repository and considered to be known. The known fact along with the results of the WCC algorithm implies if there are no isolated clusters (subgraphs) in the dataset then that means every parent node is acting as a child node but some of the child nodes may or may not be a parent.

- **There can be end nodes in the graph:** This is a consequence of the above hypothesis. Since every child node doesn't act as a parent, it proves the presence of end nodes in the graph.

The taxonomy tree visualization in Figure 3.1 displays the basic structure of the NCBI taxonomy database. There were hurdles in displaying the whole database with 2.3 million nodes due to the density of the graph and memory restrictions. Therefore the number of nodes was restricted while clear inferences can still be made from the graph. The GraphXR plugin [4] came in handy to organize the NCBI taxonomy database into a tree structure as every time a node got expanded. It also provides the functionality to utilize other graph formats such as 3D graph visualization, Cubic visualizations, and many more. One can expand any node and

learn the properties of the node in the graph if they wish to traverse it further. The major advantage of using this visualization plugin was that any time a graph got created there was an option to save the present state of the graph as a snapshot. The changes and manipulations made on the overview ranging with approximately 46,000 nodes as represented in Figure A.1 in Appendix A that was retrieved from a saved snapshot. This way of saving the current progress also serves as an add-on to save different layouts and perspectives of the graph.

Initially, the overview tree graph was generated using the Neo4j Bloom Plugin. It involves the same procedures as GraphXR. The user can enter the cypher query in a separate panel in Neo4j Bloom to visualize the graph. There is also a possibility to limit the number of nodes and hierarchical levels to display. But one disadvantage in Neo4j Bloom was that the visualizations were not as appealing as GraphXR. Neo4j Bloom was restricted to the 2D representation and it was difficult to capture the overview of the taxonomy tree when the depth of the tree was increased.

4.2 Importance of Evolutionary Ranks

The ranks of a taxonomic branch are important because they provide information such as the placement of the organism under the hierarchical taxonomic classifications. An analysis in the biological field that involves the use of molecular sequence is highly dependent on the taxonomy data which is classified based on the ranks [18]. Therefore, the locating of rank names of a particular taxon in the taxonomic tree is essential to understand its hierarchy . Around 300 years ago, the Swedish botanist, zoologist, and taxonomist Carl Linnaeus devised a classification system to classify the organisms sharing similar traits into groups [15]. The canonical top-down order of the taxonomic ranks present in the NCBI dataset is shown in Figure 3.2.

There have been many updates to this classification system since the 1700s. For instance, the prevailing largest community of similar organisms is no longer named as the “domain” rank type. It has been replaced with a new rank called “superkingdom” with an addition of a new branch called Viruses. The NCBI curators have found growth of new organisms that do not fall under the category of ranks Linnaeus formulated [15]. Back then there were only 8 predominant ranks but the dataset that we possess reveals that the organisms have evolved into new types and there are a total of 45 ranks in which the organisms are classified to date. Many of these organisms are found to be “species” followed by surprisingly unclassified organisms under “norank” and “clade” rank types (Appendix A). To understand the results in a precise manner, the 8 predominant rank types :“superkingdom”, “kingdom”, “phylum”,” family”, “class”,” order”, “genus” and “species” are focused during the analysis in this project.

From the results generated in the section [3.2] and [3.3], it can be inferred that the Linnaean Classification system awaits revision.

4.3 Inferences of "norank", "clade" rank types and results of Rank Skipping

During the analysis, the occurrence of "norank" and "clade" rank types were abundantly observed. These organisms or nodes have not been assigned a specific rank by the phylogenetic systematics, then they are labeled as a "norank". The NCBI database has made a comprehensive update on the curation of norank and clade recently and this was performed to preserve the phylogenetic connections between the organisms [26]. On the other hand, the nodes with the rank name "clade" are considered to be a monophyletic group of organisms [1]. They exhibit similar characteristics to "norank" where both the parent node and the child node may belong to the same rank type. Since "norank" and "clade" are unclassified, they can have child nodes from any of the ranks listed in Appendix A.

On the other hand, the rank skipping phenomenon is considered to be an exceptional case as they deviate from the canonical classifications. Generally, the taxonomic rank classification follows the canonical top-down order with major ranks assigned in Figure3.2. These cases probably meant that these organisms are not properly categorized yet. They might also be caused due to the inclusion of "norank" and "clade" rank types in between. Once an organism with these ranks is included, the general taxonomic ranks may be resumed.

Figure 3.7 in section 3.6, is incomplete as it does not include the rank skip counts of "species", "norank", and "clade" rank types. This is due to the population count of "species" already dominates the NCBI dataset as shown in Appendix A. Generating the rank skip count for all higher ranks to species was a hurdle because it involved generating approximately 1.3 million paths and deriving the count of rank skippings that had happened. We thus speculate that the actual cases of rank skipping should be very high in number.

4.4 Applications of dense cluster analysis

The identification of dense clusters is to locate the parent node that have a high number of child nodes by using the graph database. The cypher querying functionality in the GraphXR platform enables the user to visualize the particular branch of the graph that the user is interested in. This functionality to visualize dense clusters and their subsequent child nodes would serve as a valuable tool to explore the traits in an organism and analyze its hierarchy in the taxonomic tree.

The NCBI taxonomy dataset used in this project was extracted during April 2021. It showed that the H1N1 subtype organisms have the maximum number of child nodes as shown in Figure 3.5. The more recent batch might have included more organisms. This analysis can also be applied to the updated taxonomy dataset to see if the nodes of coronavirus are among these dense clusters.

4.5 End node Analysis

Figure [10] shows that the majority of the nodes were not expandable during the visualization of the taxonomic tree. This is because these nodes do not have a child node and act as an end node. Figure 3.6 represents the end node statistics of four “superkingdom” present in the database. As always “species” dominates and make up to 67.7% of the total composition which has had its origin from the “superkingdom” Eukaryotes. Also, an interesting observation from Figure 3.6 for Viruses, indicated that the end nodes were composed of only "norank", indicating that they are poorly classified than others. This also suggests that taxonomic classification of Viruses is a field to be refined.

The Table 3.1 presented in section 3.5 has a detailed overview of end nodes originating from “kingdom” branches. The column Kingdom represents the scientific name of 11 kingdoms in the NCBI dataset. Metazoa “kingdom” yields the highest end nodes among all the kingdoms with 956,446 nodes, while Helvetiavirae “kingdom” contributes the least with 8 end nodes.

Figure 3.1 shows that the majority of the nodes were not expandable during the visualization of the taxonomic tree. This is because these nodes do not have a child node and act as an end node. The “kingdom” is the second topmost hierarchical level in phylogeny. But the results in section 3.5 Figure appear to be as they are because of the inclusion of “norank” and “clade” nodes in the taxonomic tree. Therefore the in-between “norank” and “clade” nodes connected to the root accounts to bigger branches that ended up having end nodes throughout the tree. The end nodes are also formed due to the natural discontinuity in the taxonomic branches. In the NCBI dataset, there are a total of 11 kingdom nodes. But the results from section [3.5] shows that there are about 602,667 end nodes without including the branches that originated from the 11 kingdoms. The remaining end nodes i.e., approximately 1.5 million are present below the “kingdom” rank type.

An interesting observation from Figure[15] for Viruses, indicated that the end nodes were composed of only "norank" but their counterparts were all associated with "species". This suggests that taxonomic classification of virus is a field to be refined.

4.6 Applications of the GUI for estimating phylogenetic distance

The GUI facilitates the end-user in finding the phylogenetic distance and nodes information. The GUI focuses on displaying the scientific name as a part of the node information because not all nodes have a common name.

The results of this analysis will help the researchers understand the phylogenetic relations the organisms would have likely evolved from. The distance also defines the relatedness of organisms to one another. Therefore, the GUI provides a robust

way of inferring the phylogenetic distance between any two given organisms in the context of all the known species, this would allow efficient and accurate estimation for the relative evolutionary relationships among organisms discovered in complex biological communities, such as from environmental samples or human gut microbiota. Moreover, it aids in performing an in-depth analysis of the characteristics of an organism and understanding evolutionary history.

4.7 Use of Tableau Desktop Software

The CSV files required to produce the charts presented in chapter 3 were derived using cypher querying and python programming. This project is also partly dependent on the use of Tableau Desktop for creating statistics of data due to its user-friendly functionalities. It requires no coding or scripting knowledge to generate the charts. It came in handy to conceive all the data in the CSV files in the format that we were interested in. Most importantly the packed bubble illustration shown in Figure 3.5 was created using the Tableau Desktop. It was noted that these kinds of representations were missing in other visualization plugins such as GraphXR or Neo4j Charts.

5
Conclusion

In the process of using a vast ever-growing biological database, this project has focused on conducting comprehensive analysis on the taxonomic data extracted from the NCBI FTP website [20]. To leverage the phylogenetic relationships, this pilot effort utilized the Neo4j Graph database by which statistical analysis of the taxonomic tree was conducted.

The overview of the taxonomy tree structure of the graph aided as a supplement to the additional analysis that was performed on the dataset. It helps the researchers in finding patterns and learning more complex structures that are present in the NCBI taxonomy database. This in turn can also be used to find the shared ancestors of any two organisms. From the results generated in section [3.3] to [3.6], it was observed that many taxons uploaded to the NCBI database fail to follow the Linnaean Classification. The naming nomenclature of these taxons become intriguing due to the constant deviations.

The taxonomy browser is available on the NCBI website[5]. This webpage displays the lineages of ranks as sequential lists but lacks in featuring analytics for particular organisms or rank types. In such cases, Graph databases armed with an arsenal of data science tools help achieve insights by pattern identification and machine learning models. The Taxonomy Toolbox created in section [3.7], uses a graph database as its backend which is capable of using the power of data science algorithms to traverse the graphs and provide useful insights. By utilizing cypher queries users can obtain customized lineages on any interested organisms. This toolbox was built as a proof of concept that Graph Databases are better suited for complex biological databases and more functionalities can be explored in the path of using such Graph Analytics tools.

This master thesis project has made use of the Neo4j Graph Database extensively to explore the NCBI dataset and it paves the way for future research projects in conducting in-depth analysis together with other biological datasets. It is concluded that the results of this thesis can be used as a basis for further applications of Graph Databases in the fields of biological research.

Bibliography

[1] Clade. *Merriam-Webster Dictionary*. URL: https://www.merriam-webster.com/dictionary/clade.

[2] Essentials of neo4j 3. 0, from scale to productivity & deployment. URL: https://neo4j.com/blog/neo4j-3-0-massive-scale-developer-productivity/#capabilities-data-size.

[3] Gephi - the open graph viz platform. URL: https://gephi.org/.

[4] Graphxr visualization. Publication Title: Kineviz. URL: https://www.kineviz.com/visualization.

[5] Ncbi taxonomy browser. publisher: NCBI. URL: https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi.

[6] Neo4j. Publication Title: Neo4j Graph Database. URL: https://neo4j.com/download/.

[7] Neo4j bloom - graph visualization and collaboration. Publication Title: Neo4j Graph Database Platform. URL: https://neo4j.com/product/bloom/.

[8] Neo4j browser user interface guide - developer guides. Publication Title: Neo4j Graph Database Platform. URL: https://neo4j.com/developer/neo4j-browser/.

[9] Neo4j python driver 4. 3. URL: https://neo4j.com/docs/api/python-driver/current/.

[10] Press releases 2010 \ census of marine life. URL: http://www.coml.org/press-releases-2010/.

[11] Pysimplegui. URL: https://pysimplegui.readthedocs.io/en/latest/.

[12] Tracking SARS back to its source. URL: https://evolution.berkeley.edu/evolibrary/news/060101_batsars.

[13] Weakly connected components - neo4j graph data science. Publication Title: Neo4j Graph Database Platform. URL: https://neo4j.com/docs/graph-data-science/1.6/algorithms/wcc/.

[14] Why neo4j? Top ten reasons. URL: https://neo4j.com/top-ten-reasons/.

[15] Linnaean classification, September 2016. Publication Title: Biology LibreTexts. URL: https://bio.libretexts.org/Bookshelves/Introductory_and_General_Biology/Book%3A_Introductory_Biology_(CK-12)/05%3A_Evolution/5.01%3A_Linnaean_Classification.

[16] Christian Chabot, Chris Stolte, and Pat Hanrahan. Tableau software. *Tableau Software*, 6, 2003.

[17] EMBL-EBI. Why is phylogenetics important? \textbar Phylogenetics. URL: https://www.ebi.ac.uk/training/online/courses/introduction-to-phylogenetics/why-is-phylogenetics-important/.

[18] Scott Federhen. The NCBI Taxonomy database. *Nucleic Acids Research*, 40(Database issue):D136–143, January 2012. doi:10.1093/nar/gkr1178.

[19] Amy Hodler and Mark Needham. *Graph Data Science for Dummies*. Wiley.

[20] National Center for Biotechnology Information, U. S. National Library of Medicine 8600 Rockville Pike, Bethesda MD, and 20894 Usa. National center for biotechnology information. URL: https://www.ncbi.nlm.nih.gov/,https://ftp.ncbi.nlm.nih.gov/pub/taxonomy/.

[21] Kankesu Jayanthakumaran, editor. *Advanced technologies*. In-Teh, 2009.

[22] Ezequiel López-Rubio and Emanuele Ratti. Data science and molecular biology: prediction and mechanistic explanation. *Synthese*, 198(4):3131–3156, April 2021. URL: https://link.springer.com/10.1007/s11229-019-02271-0, doi:10.1007/s11229-019-02271-0.

[23] Ankita Rahavachari, Guru Prakash Subramanian, and Hao Wang. URL: https://github.com/Hao-Chalmers/TaxonomyInGraph.

[24] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph databases [new opportunities for connected data*. O'Reilly, Sebastopol, CA, 2015. OCLC: 1028626678. URL: https://neo4j.com/graph-databases-book/?ref=home.

[25] Tetsu Sakamoto and J. Miguel Ortega. Taxallnomy: an extension of NCBI Taxonomy that produces a hierarchically complete taxonomic tree. *BMC Bioinformatics*, 22(1):388, December 2021. URL: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-021-04304-3, doi:10.1186/s12859-021-04304-3.

[26] Conrad L Schoch, Stacy Ciufo, Mikhail Domrachev, Carol L Hotton, Sivakumar Kannan, Rogneda Khovanskaya, Detlef Leipe, Richard Mcveigh, Kathleen O'Neill, Barbara Robbertse, Shobha Sharma, Vladimir Soussov, John P Sullivan, Lu Sun, Seán Turner, and Ilene Karsch-Mizrachi. NCBI Taxonomy: a comprehensive update on curation, resources and tools. *Database*, 2020:baaa062, January 2020. URL: https://academic.oup.com/database/article/doi/10.1093/database/baaa062/5881509, doi:10.1093/database/baaa062.

[27] Wei Shen and Hong Ren. TaxonKit: A practical and efficient NCBI taxonomy toolkit. *Journal of Genetics and Genomics*, page S1673852721000837, April 2021. URL: https://linkinghub.elsevier.com/retrieve/pii/S1673852721000837, doi:10.1016/j.jgg.2021.03.006.

[28] National Geographic Society. Biodiversity, August 2019. Publication Title: National Geographic Society. URL: http://www.nationalgeographic.org/encyclopedia/biodiversity/.

[29] Wikipedia contributors. Phylogenetic tree — Wikipedia, The Free Encyclopedia, 2021. URL: https://en.wikipedia.org/w/index.php?title=Phylogenetic_tree&oldid=1023200352.

[30] Edward Orlando Wiley and Bruce S Lieberman. *Phylogenetics: theory and practice of phylogenetic systematics*. John Wiley & Sons, 2011.

A

Appendix

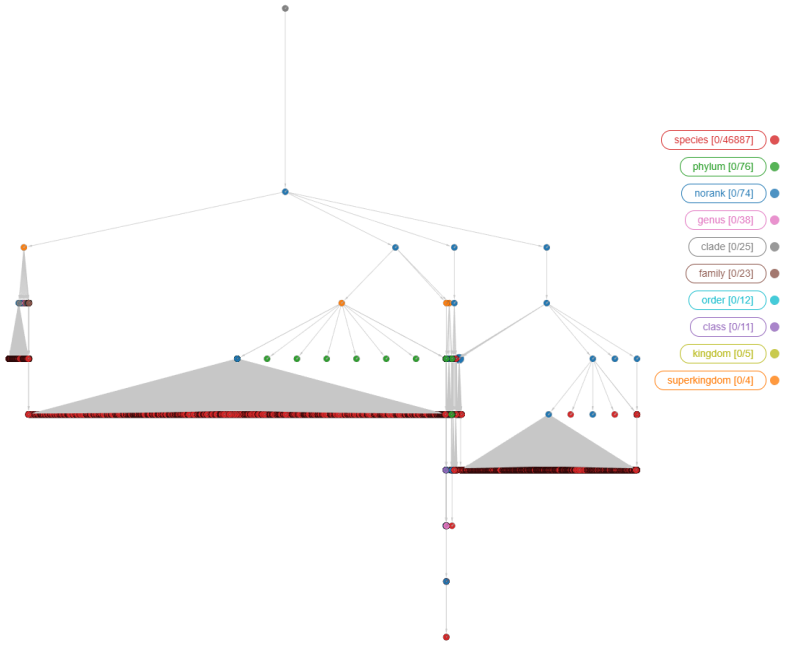

Figure A.1: Overview of NCBI taxonomy database as a tree structure with approximately 46,000 nodes represented in different colours based on their rank

Rank Name	**Count of Nodes**
species	1,896,462
norank	223,497
genus	97,451
strain	44,647
subspecies	25,390
family	9,563
varietas	8,595
subfamily	3,062
tribe	2,219
subgenus	1,690
order	1,685
isolate	1,320
serotype	1,243
clade	894
superfamily	865
formaspecialis	741
forma	572
subtribe	526
section	473
class	442
suborder	374
speciesgroup	331
phylum	284
subclass	159
serogroup	139
infraorder	130
speciessubgroup	124
superorder	55
subphylum	33
parvorder	26
subsection	21
genotype	20
infraclass	18
biotype	17
morph	12
kingdom	11
series	9
superclass	6
cohort	5
subvariety	5
pathogroup	5
superkingdom	4
subcohort	3
subkingdom	1
superphylum	1

Table A.1: The rank names available in the NCBI dataset along with their count