Graph-based representation, integration, and analysis of neuroscience data

The case of the murine basal ganglia

Maren Parnas Gulnes

Thesis submitted for the degree of
Master in Informatics: Programming and System Architecture
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2020

Graph-based representation, integration, and analysis of neuroscience data

The case of the murine basal ganglia

Maren Parnas Gulnes

Graph-based representation, integration, and analysis of neuroscience data

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

The amount of publicly available brain-related data has significantly increased over the past decade. Neuroscience data is spread across a variety of sources, typically provisioned in ad-hoc manners and non-standard formats, and often with no connections between the various sources. This makes it difficult for researchers to understand, integrate, and reuse brain-related data. There is a clear need to find effective mechanisms to manage data in this field, especially since brain-related data is highly interconnected, evolving over time, and often needed in combination. At the same time, the field of data management has recently seen a shift from representing data in the relational model towards alternative data models. Especially graph databases have seen an increase in use due to their ability to manage highly-interconnected, continuously evolving data.

This thesis presents an approach for organizing brain-related data in a graph model, investigates how the graph representation affects the understanding of the data, how it facilitates the integration of data from various sources, and how it enhances the usability of the data. The thesis exemplifies the approach in the context of a unique data set of quantitative neuroanatomical data about the murine basal ganglia — a group of nuclei in the brain essential for processing information related to movement. Specifically, the murine basal ganglia data set is modeled as a graph, integrated with relevant data from third-party repositories (Brain Architecture Management System, InterLex, and NeuroMorpho.Org), and analyzed this data using popular graph algorithms to extract new insights. Access to the data is provisioned via a web-based user interface and API. A thorough evaluation of the graph model and the results of the graph data analysis and usability study of the user interface indicate the potential of graph-based data management in the neuroscience domain. The thesis contributes with a practical and generic approach for representing, integrating, analyzing, and provisioning brain-related data, and a set of software tools to support the proposed approach.

Acknowledgments

I would like to thank everyone who supported, guided, and contributed to this thesis. First and foremost, I want to thank my main supervisor Dumitru Roman for formidable guidance, assistance, and motivation throughout this work and writing of this thesis. I am grateful for his enlightening and invaluable support. Next, I want to thank to my co-supervisor, Ingvild Elise Bjerke, for her expertise in neuroscience and invaluable feedback throughout the thesis research.

Further, I extend my gratitude to the people at SINTEF Digital and the Analytical Solutions and Reasoning research group at the University of Oslo for their support and feedback. I also want to thank the people at the Department of Medicine at the University of Oslo for insight and inspiration to the thesis research, and especially the researchers who participated in the thesis studies for providing valuable feedback.

Finally, I want to thank my friends and family for their motivation and support. Particularly my partner Roald Lyngvær for his patience and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

The brain is the organ humans rely on the most but understand the least; however, not for the lack of trying. Since ancient times, humans have wondered about the mind, hoping to comprehend its function fully. Neuroscience research has generated large amounts of data about the brain. The amount has increased significantly over the past decade due to the advances in technology, causing heaps of brain-related data [1]. Data, however, is only a small part of understanding the brain. To convert this data into knowledge and understanding, researchers need to observe the data combined. Therefore, there is a need to examine how neuroscience data can be modeled and stored to facilitate combination and reuse.

The data that exists about the brain is in large quanta, complex, spread across repositories in multiple formats. As an example of this complexity, brain-related data can represent a part of the human brain's 86 billion neurons, and for each neuron, any of the approximately 7000 connections (synapses) [2, 3]. The amount of data available raises some concerns. First, as the data volume increases, it becomes increasingly difficult for researchers to find relevant data. Second, as researchers often collect and create data with a specific purpose, the naming and quality of the data vary, causing standardization and modeling challenges [4]. These challenges hinder reuse, combination, and sharing of data and cause the need to improve how the data is stored and managed [5].

Simultaneously, as these challenges have arisen in the field of neuroscience, there has been a shift in the data management field: From almost exclusively representing data using relational models, NoSQL solutions have become increasingly popular [6]. Especially graph databases, a type of NoSQL database, have seen an increase in use due to their ability to manage large amounts of complex data and analytical abilities [6]. In 2019, graph analytics were identified by Gartner as the fifth "Data and Analytics Technology Trends"

[7]. Gartner predicts that the use of graph data stores will increase over the next few years due to the need to ask complex questions across complex data.

While technology has helped create vast amounts of existing brain-related data, partly causing barriers for understanding, it appears that technology also will contribute significantly to solving the challenges caused by the data volume and complexity. Research has suggested ways of working with new and existing data to make it usable for neuroscientists [8, 9]. There has been some research into creating common frameworks for neural data. For example, Hamilton et al. (2012) proposed an ontological approach for describing neurons and their relationships [10]. Due to the numerous ways research can identify neurons, it is unlikely that a standard naming format for the data can exist. Consequently, research and data initiatives have created guidelines on how to handle the data, with the central notion being the data must be made available and machine-readable [8, 11].

In computational neuroscience, researchers have investigated the use of graph databases for two primary objectives: knowledge graphs for organizing publications and data sets and direct representation of connectomes (neural connections in the brain) [12, 13, 14]. Still, there is little research on graph-based data representation as a mechanism for integration, analysis, and reuse of neuroscience data. This thesis places itself within the field of neuroinformatics and data management, exploring both neuroscience and data management aspects within the given context.

1.2 Motivation

Given the challenges with reuse and combination caused by the large volume and complexity of brain-related data currently existing, there is still a need to research effective mechanisms to manage brain-related data. Existing research in managing brain-related information works towards the standardization of metadata, aiming to make it easier for researchers to find and reuse data [15, 16, 17]. This research stream has mainly focused on metadata management for data sets and little on managing the actual data for a single data set. The existing research is an essential part of answering data management challenges, such as finding relevant data and standardization and modeling challenges. Nonetheless, there is a value in exploring novel approaches to managing neuroscience data and focus further research efforts on making specific brain-related data sets available and accessible, complementing the

current research. This thesis aims to investigate new ways of organizing brain-related data to provide helpful insight and improve researchers' understanding of the data and its usability, focusing on single data sets.

In 2019, Bjerke et al. published a database of quantitative neuroanatomical data about the murine basal ganglia [18]. The database consists of data from more than 200 research papers and data repositories, manually collected and gathered, and stored in a relational database [19]. The thesis will refer to the *murine basal ganglia database* when considering the database created by Bjerke et al. and the *murine basal ganglia data set* as the data in this database. This data set's significant relevance is that it gathers and integrates the available research of quantitative neuroanatomical measures on the murine basal ganglia produced over the past decades. Before, this research spread across multiple experiments and research projects with no common point of reference. Moreover, there is generally very little data about the basal ganglia in the well-renowned neuroscience research initiatives. These aspects make the data set provided by Bjerke et al. unique with great scientific relevance.

As the murine basal ganglia database is relational, there are limited possibilities to represent the data's relations, as such databases only allow single, undirected relationships [6]. The main contributor to the database, Ingvild E. Bjerke, a Ph.D. student at the faculty of Medicine at the University of Oslo, told through conversations that there are relations between the data that are not possible to represent. Further, as the research by Bjerke et al. focused on gathering murine basal ganglia data in a unified model rather than on opportunities in the specific data management solution, this data set poses an excellent example for investigating novel approaches for managing and accessing the data. Together with the scientific relevance, the data management opportunities with the murine basal ganglia data set make it a suitable base for this thesis.

Although there is a lot of ongoing research for understanding the different areas of the brain [1, 15], managing the existing brain-related data [15, 16, 17], and using graph databases [12, 13], there is little research on managing specific neuroscience data sets using graphs. Graph-based approaches to data management in neuroscience have focused on managing sets of data and modeling networks in the brain, but not on mechanisms for modeling and storing the actual data in data sets to facilitate integration and reuse, which is the aim of this thesis. As neuroscience data is available through data sets in different formats and structures based on the data set's purpose, there is a need for flexible data structures that manage such data and further facilitates

integration and understanding. A benefit of graph databases is their ability to flexibly store dynamic and interconnected data.

1.3 Problem statement

Based on the given context, there is a need to examine how neuroscience researchers can manage their data to make the data more accessible and reusable for others. The presented motivation includes arguments for graph-based data representation to store brain-related data in this context, as it allows flexible data models for highly connected data. Thus, the thesis presents the following hypothesis:

Hypothesis (H): Organizing neuroscience data in a graph model provides a better understanding of the data, facilitates data integration with other brain-related data sets, and improves the usability of the data.

To investigate the hypothesis, we look at three aspects of the graph model: data accessibility, understanding of the data, and the ability to integrate with other data sets. We specify these in the following research questions that guided the study of this thesis:

Research question 1 (RQ_1): *Can graph-based representation of brain-related data facilitate the integration of data from a variety of neuroscience data sets?*

Research question 2 (RQ_2): *Can a graph model provide a better understanding of the data in a brain-related data set?*

Research question 3 (RQ_3): *To what extent can a graph-based approach to neuroscience data management improve the usability of the data?*

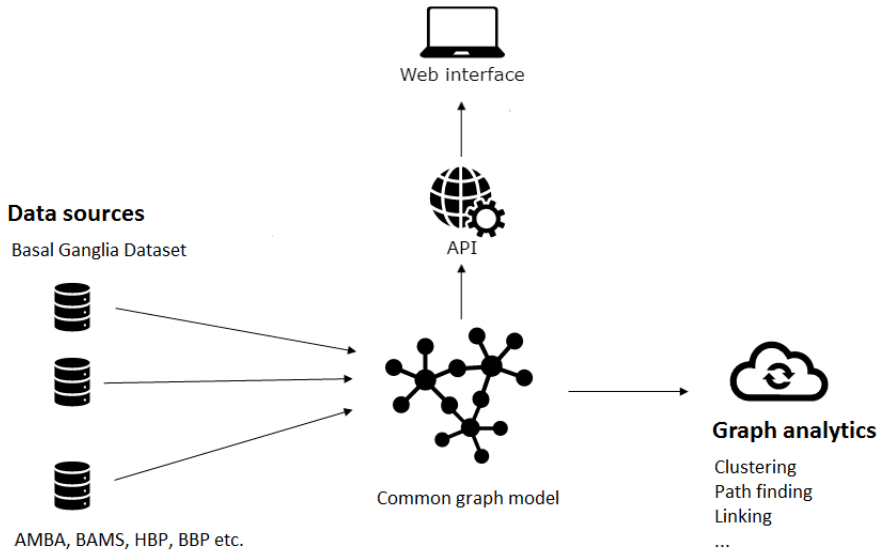

Figure 1.1: An overview of the graph-based data modeling components of this thesis.

1.4 Thesis scope

The scope of this thesis is to investigate graph-based data management for neuroscience data. In order to do so, we start with a unique data set in the neuroscience domain, the murine basal ganglia data set, which we define as the baseline data. With this data set, the thesis presents an approach for representing neuroscience data in a graph model, onboarding the data, integration with external data sets, storing, serving, and analyzing the integrated graph data.

Figure 1.1 presents the general idea of the thesis. This figure displays a high-level view of the thesis' key components, including graph-based data modeling, integration with external data sources, provisioning the integrated data, and data analysis. These components are employed to investigate graph-based data representation in the neuroscience domain.

The thesis provides insights into the presented research questions using quantitative and qualitative measures. These measurements include usability studies, survey results and interviews with neuroscience researchers, and interviews with Bjerke, one of the murine basal ganglia database creators.

The thesis does not cover approaches to brain modeling or simulation; rather, it focuses on how researchers better can work with the brain-related data they produce. It also does not cover a comparison between relational databases and other NoSQL solutions, as others widely cover this topic [20]. Instead, it focuses its research on why a graph model is beneficial for brain-related data and how such a model can help make the data more available and reusable.

1.5 Research design

Literature divides research methodologies into different paradigms. The Association for Computing Machinery (ACM) defines three paradigms; (1) Theory, (2) Abstraction, and (3) Design [21]. The ACM defines the *abstraction* paradigm as directed towards scientists investigating a phenomenon with the desire to obtain new knowledge. Further, the ACM defines the *design* paradigm as directed towards engineers who want to construct a system to solve a given problem. The abstraction and design paradigms, defined by ACM, correspond to what Solheim and Stølen (2007) describe as classical research and technology research, respectively [22]. Solheim and Stølen (2007) define *classical research* as formulating a hypothesis and verifying it using experiments and observations, aiming to answer "What is the real world like?" while they define *technology research* as "research for the purpose of producing new and better artifacts" [22].

Researchers should decide their research methodology based on the research setting, and in many cases, it is better to combine multiple methodologies to achieve the most accurate result [23, 24]. To facilitate the thesis research setting, we use a combination of classical and technology research.

Although this thesis presents research questions that try to answer facts about the (computing) world, we primarily did so by measuring artifacts, namely the graph model and its extensions. Accordingly, we applied technology research as the principal methodology, with classical research elements (experiments) in this thesis. The following steps represent the methodology used in this thesis to study the research questions:

1. Problem analysis: Specifying scope and defining requirements
2. Design and implementation

3. System evaluation: Evaluation of requirements and experiment design

4. Repeat from 1 or 2 based on the system evaluation result

Research methodologies employ research methods, divided into qualitative, quantitative, and mixed research methods [23]. *Qualitative* research measures the quality in some sense and produces data expressed as text, images, or forms, except numbers. *Quantitative* research produces data expressed as numbers, and researchers can use such data to create statistics. *Mixed-method* research combines these methods to draw from their strength, minimize their weaknesses, and better understand the researched topic [25, 26]. This thesis used mixed-method research with qualitative methods being dominant. The following list introduces the methods applied through this thesis:

- **Literature review:** We performed a literature review in the early phases of the thesis research to understand the neuroscience domain and the existing challenges and opportunities with brain-related data. Chapter 2 presents the outcome of the literature review.

- **Survey:** The thesis research employed surveys to understand how neuroscientists work with brain-related data and the challenges they experience. Chapter 3 describes the outcome of the survey.

- **Usability study:** The thesis research performs a usability study to evaluate the quality of the developed data user interface. In this study, we qualitatively assessed the users' experience of the interface and quantitatively measured if they could complete the given tasks. The evaluation chapter presents the outcome of the usability study in Section 5.3.2.

- **Qualitative interviews:** We utilized interviews to assess researchers' overall experience with the web-based user interface and evaluate how the graph model affects Bjerke's understanding of the data set data. The outcome of these interviews is presented in Chapter 5.

- **Experiments:** Experiments define studies where researchers introduce an intervention to observe the effects. We performed an experiment when connecting data from external sources with data in the graph-based murine basal ganglia data set. Further, we performed data analysis experiments to observe how it could extract new information from the thesis data set.

1.6 Thesis outline

Chapter 2: Background This chapter describes the background of the thesis, introducing the thesis graph and neuroscience perspectives. First, the chapter introduces graph-based data representation and analysis before presenting characteristics of neuroscience data. Finally, it presents the state of data management for brain-related data, including graph-based approaches.

Chapter 3: Problem analysis This chapter analyses the thesis problem space, starting with a review of the murine basal ganglia data set before describing the requirement specification process for the artifacts the thesis research developed to evaluate the research questions. Specifically, it presents the graph model's design requirements, analyzes data integration with external data sources, specifies the data analysis requirements, and defines the software applications for web-based data access. For each artifact, the chapter presents the resulting implementation requirements.

Chapter 4: Solution design and implementation This chapter describes the solution design and implementation of the thesis artifacts. The chapter starts by introducing the solution's high-level architecture before presenting the solution design, including the graph model, data onboarding, data integration, data analysis, and web and application programming interface (API) applications. Finally, the chapter describes the implementation of the proposed design, including technological decisions and component integration.

Chapter 5: Evaluation This chapter presents the evaluation of the requirements stated in the problem analysis chapter. First, the chapter presents an evaluation of the graph model and database before describing the data analysis experiments to derive new information about the data in the murine basal ganglia data set and an evaluation of these results. Finally, the chapter presents an evaluation of the user interface and the result of the usability study.

Chapter 6: Conclusion and further work The final chapter concludes the thesis research. First, it summarizes the thesis research before presenting the thesis contributions. Finally, the chapter proposes further work for data management in the neuroscience domain.

Chapter 2

Background

This chapter provides the thesis background, introducing perspectives of graph-based data management and brain-related data. As the thesis evaluates the use of graphs for neuroscience data, this chapter presents concepts of graph-based data representation in Section 2.1 and neuroscience data concepts in Section 2.2 before reviewing existing brain-related data management and graph-based approaches in Section 2.3.2.

2.1 Graph-based data representation

Many real-world scenarios are naturally structured as graphs, such as social networks and neuron connectivity. Graph-based data representation provides a way to represent such real-world structures directly. Graph-based data representation entails all representations of data that utilize a graph model. This section introduces graph definitions, and graph database features relevant for evaluating the graph database implementation in this thesis.

2.1.1 Graph definitions

Graph theory is a discipline within discrete mathematics regarding the study of graphs. This mathematical discipline roots back to the 18th century, when the mathematician Leonard Euler created a mathematical proof (*The seven bridges of Königsberg*) using a graph representation [27, 28]. With his proof, Euler displayed how a real-world scenario could map directly to a graph representation. To understand how researchers can utilize graph representations, we need to understand graph models.

There are various types of graphs. Discrete mathematics defines a graph, or a simple graph, as a set of *vertices* (nodes) and *edges* (relationships). Nodes connect through edges, and all edges in a graph go between two nodes in the node-set [29, 30]. In some definitions, a simple graph is not allowed to have self-loops. A *self-loop* is an edge that starts and ends in the same node. In a

broader definition employed in this thesis, a graph is allowed to have self-loops. A graph is either directed or undirected. In a *directed graph*, the edge exits one node and enters another. In an *undirected graph*, the edge has no direction [29]. A *hypergraph* is an extension of a simple graph, allowing the edge-set to be of any cardinality, meaning that an edge can connect more than two nodes. [30]. Another type of graph is multigraphs. A *multigraph* is a graph that allows multiple relationships between a node pair. If the edges between a node pair have the same direction, the edges are parallel. Hypergraphs and multigraphs are either directed or undirected. Figure 2.1 shows examples of a directed graph, an undirected graph, a directed hypergraph, and a directed multigraph.

A graph also has a set of properties. One such property is the node degree. The *degree of a node* defines how many edges connect to the node. In other words, a node's degree is the same as the number of neighbors of that node [30]. Node degrees are often a consideration when evaluating influential nodes in a graph [30]. Node degrees relate to the connectivity property. A graph is *connected* when all nodes in the graph connect so that a path from any node can lead to all other nodes [30]. Often, a graph that is not connected has connected sub-graphs. Many analysis methods and graph algorithms utilize the connectivity of a graph, such as community detection algorithms.

Statistical measures often have specific requirements for the graph or use specific graph properties for evaluation. Moreover, some graph types allow for specialized analysis, such as directed graphs. When representing data in a graph model, it is, in many cases, necessary to observe which type of graph is most suited to represent the data and which properties that graph has, as the type and properties affect how one can utilize a graph.

2.1.2 Graph databases

To store and represent graph models, computer scientists can utilize graph databases. Graph database management system (GDBMS), in this chapter referred to as graph databases, are a type of NoSQL databases. This thesis will not give a detailed background on NoSQL as many papers have done that before, such as Hecht and Jablonski (2011) and Cattell (2011) [20, 31]. However, a short introduction helps recognize the motivations behind graph databases. In the field of data management, there has been a shift in data storage forced by the changes in technology and the amount and types of data available [20]. As the internet evolved, so did the amount of data,

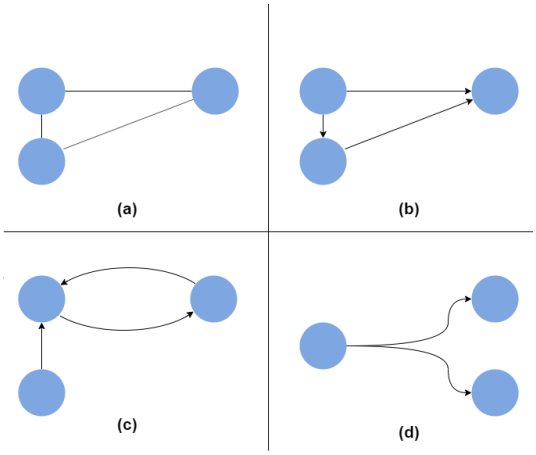

Figure 2.1: Various types of graphs: (a) An undirected graph; (b) A directed graph; (c) A multigraph (directed); (d) A hypergraph (directed).

and applications now need to handle large quanta and continually changing data. The need for features lacking in the traditional relational database management systems (RDBMS), due to these systems' normalized data models and full ACID support, gave rise to NoSQL databases [32]. These demands included fast concurrent read and write operations, efficient (big) data storage, and high availability and scalability [6]. NoSQL databases perform fast read and write procedures, support large data sets, and deal well with dynamic data, both changes in the schema and the data size [6, 20, 31, 32]. Providing these features comes at a cost, and each NoSQL solution differs in what they sacrifice to achieve them. By this, one must consider which properties are necessary and which might not be when selecting a NoSQL database system [32]. As mentioned, many real-world scenarios are represented more appropriately as graphs than classes to capture their nature [28]. The key features of such scenarios are that the data is highly interconnected and can come in high volumes, which are features that graph databases handle well [33].

There are multiple definitions of graph databases, but they all share the key concept of nodes and relationships [6, 34, 33]. Robinson et al. (2015) define a graph database as a database management system with Create, Read, Update, and Delete (CRUD) methods that expose a graph data model [6]. When defining graph databases, there is a separation between native and non-native implementations [35, 34]. In this thesis, we define a *native graph*

database as a graph database that has a graph data model in the underlying storage. It processes the data using index-free adjacency, meaning that the connected database entries (nodes) point to each other's physical location [6].

A relational data model can also be viewed as a graph, but with limitations. Entity-relationship (ER) diagrams, commonly used to model and presents relational databases, are graphs where the tables represent nodes, and the foreign-keys define named relationships. Appendix A presents such a diagram of the relational murine basal ganglia database. These diagrams depict a limitation of relational data models; they only allow single, undirected relationships between the nodes [6]. With this limitation, relational databases are ill-suited for representing domain models with numerous, diverse relationships between entities.

The key advantages of graph databases boil down to performance, flexibility, and agility [34]. The performance advantage appears when querying deep data as the performance stays roughly constant even when the amount of data increase over time [34]. The graph databases' ease of changing schemas provides flexibility; graph databases do not have strict predefined schemas that all nodes of a particular type need to follow. Instead, one can define what needs to be there as the database and application evolve, representing the domain model. The non-strict schemas also supply the agility advantage, allowing the database to change with the domain requirements [34]. The key benefits, performance, flexibility, and agility fit well with the thought out use cases of graph databases; continually evolving, interconnected data.

2.1.3 Graph data models

A graph database exposes a graph data model. The previous section on graph definitions shows that a graph model in its purest form consists of nodes (vertices) and relationships (edges). Angles and Gutierrez (2008) define a graph database model as a model where the data structure (schema) is modeled as a graph and where the data manipulation uses graph-based operations [36]. There are many different graph data models, but the two most common are the property graph model and the RDF graph model.

A *property graph model* is a graph model with nodes and relationships where both the nodes and the relationships can have properties. The model categorizes the nodes with one or more labels, and the relationships are named and directed [6]. Figure 2.2 illustrates a property graph model where the nodes have one label, some of the nodes have properties, and one of the

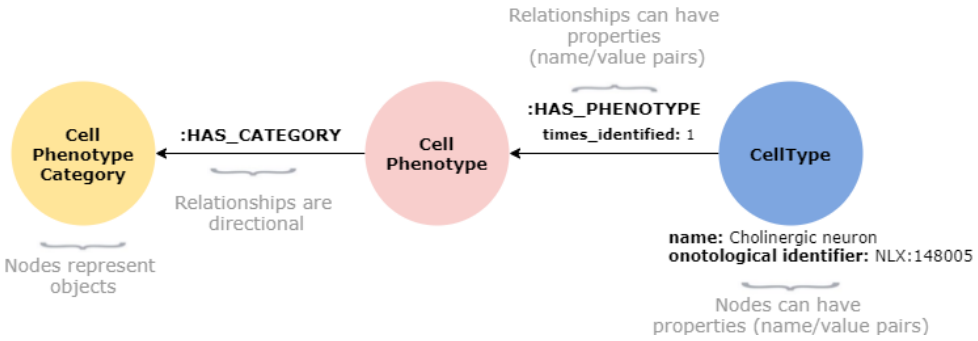

Figure 2.2: A property graph model of a subset of the murine basal ganglia data set.

relationships have a property [6].

Triple stores, or RDF stores, originating from the Semantic Web Movement, present a different graph model [6]. The Resource Description Framework (RDF) is a standard model, developed under the World Wide Web Consortium (W3C), enabling the encoding, exchange, and reuse of structured metadata [37]. The goal was to make a framework for all the World Wide Web resources to improve programmatic discovery and access to these sources [37, 38]. A triple store is a database that, as the name implies, stores triples. A triple is composed of a subject-predicate-object [38]. A set of RDF triples creates an RDF graph. Figure 2.3 illustrates a simple RDF graph. In the figure graph, the Wikipedia URL is the subject, the `dc:title` and `dc:publisher` are the predicates, and `Basal Ganglia` and `Wikipedia` are the objects. In an RDF graph, the nodes and edges do not contain properties; rather, the edges define the properties. Triple stores are graph databases as they expose graph data models [38]. However, they do not usually implement a graph data model in the underlying storage [6].

When selecting a graph database model, it is necessary to consider the data that the model represents, the structure of the data, including how it is interconnected, how the data evolves, and the graph model's features. Some graph models are better suited for data manipulation, while other models provide improved data analysis. Further, different graph models will allow different analytical possibilities. One should evaluate the scope of the data before choosing the appropriate graph model.

There are many available graph database implementations, and they utilize different graph models. Examples of graph database implementations

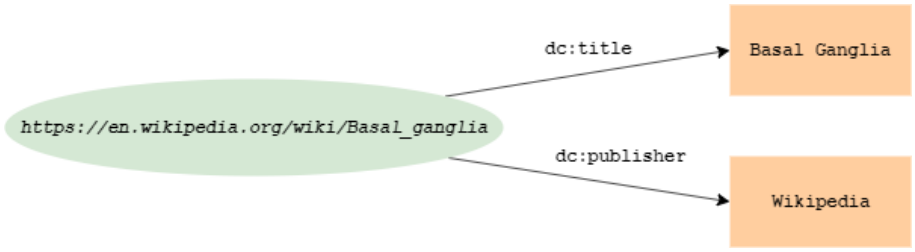

Figure 2.3: A simplistic RDF graph of the Wikipedia page of the basal ganglia.

that expose a property graph model are Neo4j[1], OrientDB[2], and ArangoDB[3]. Graph database implementations that expose an RDF graph includes AllegroGraph[4] and Apache Jena[5][34, 39]. Although sharing the same graph model, these implementations have different features necessary to evaluate when selecting an implementation. There are many graph database implementation comparisons available, such as Fernandes and Bernardino (2018) that compares the implementations referenced above, except Apace Jena [34].

Another aspect of graph databases is the query language. As the graph data model depends on the data it represents, the query language depends on the chosen graph data model. Cypher is a declarative query language developed by Neo4j to query property graph models, such as the Neo4j database engine [40]. SPARQL is the graph query language used to query RDF graphs [38]. The query languages are optimized to query their respective graph data model.

There are different graph patterns for querying graphs. Renzo et al. (2017) define *basic graph patterns* as graph queries that perform matches against the graph database, and *complex graph patterns* as basic patterns augmented with features, such as projection, union, optional, and difference [41]. Path queries are also essential for graph querying, to understand and navigate the topology of the data. Combined with basic queries, they are named *navigational graph patterns* [41]. The main SPARQL query building blocks are triple patterns, referencing the RDF triples [41]. By its long existence, SPARQL is

[1] `https://neo4j.com`
[2] `https://www.orientdb.org`
[3] `https://www.arangodb.com`
[4] `https://allegrograph.com`
[5] `https://jena.apache.org`

more studied than Cypher and has full navigational query support [41]. In Cypher, "patterns" are the main building blocks, and the language supports navigational queries over a property graph [41]. In summary, both Cypher and SPARQL support complex query patterns and navigational queries.

2.1.4 Graph analytics

Graph analytics includes all approaches to analyze graph-based data [35]. When working with large data sets, it is not customary to know all the data. If one has all of this information, graph analysis will not provide much interesting information. Nevertheless, this is usually not the case. For a minimal data set, graph visualization might provide enough information to give an overall understanding of the structure. For large amounts of data, that is not sufficient. Thus, analysts apply mathematical measures and use tools on the graph data set to boil down large amounts of data into simple numbers that are easy to understand [42]. As this thesis utilized graph data analysis to evaluate research question RQ_2, we describe in this section a set of graph analytics approaches.

State-of-the-art graph analysis approaches

Graph machine learning entails approaches that analyze graph data and creates machine-learning models that work on graph data. In 2009, Scarselli et al. proposed a graph neural network (GNN) model that utilized existing neural network methods on data represented in a graph model [43]. Before GNNs, computer scientists had to convert graph-represented data into other representational forms, such as vectors, to use neural networks. Scarselli et al. proposed the GNN as a neural network that inputs and returns data in a graph representation.

Graph neural networks have gained some use over the past decade [44]. There are many scenarios for using GNNs to predict and classify graph data models. Some are related to traditional machine-learning tasks, such as models for text and image classification. Other scenarios are more specific to data naturally structured as graphs, such as disease classification, protein interface prediction, and knowledge graph completion and alignment [44].

The significant benefit of using GNNs, when having data represented in graph format, is that they provide a way to obtain a machine-learning model that can predict the result for data added after the model creation.

Compared to traditional graph algorithms presented later in this section, the machine-learning model will work for inserted nodes. In contrast, a graph algorithm must be run on all the nodes at every insert to obtain the same result. The disadvantage of using such machine learning methods is that the model becomes a black box where the computer scientist cannot tell how the model predicts or classifies. The primary benefit of using machine learning approaches on graph data appears when the data set is so large that traditional analytics are insufficient.

Another way to analyze graphs is related to the concept of knowledge graphs. There are many definitions of knowledge graphs [45]. In this thesis, a knowledge graph describes a graph data model where the node labels and relationship types are predefined, limited, and has a concrete definition. Such a framework for labels and types is often referred to as an ontology. The data modeled in a knowledge graph must integrate into this formalization.

Knowledge graphs apply reasoners to extract new information or knowledge from the data [45]. A reasoner is software programs that can infer logical consequences from a set of given facts and rules. An example is Apache Jena, a toolkit for loading and processing information in an RDF graph, which contains inference-frameworks that can work directly on the data in an RDF graph [39].

Although powerful tools, this thesis does not utilize machine learning and reasoning approaches. Graph neural networks are well suited for graphs with a similar or equal set of properties and large data sets. As with all machine learning methods, the graph neural network removes some of the graph information to run on large data sets. Knowledge graphs using reasoners prove very powerful to derive new knowledge about data. However, to do so, there is a need for formal standardization. The thesis research found no ontology that the murine basal ganglia data set could integrate into where it was possible to define all of the different neuroscience research methods. Creating such an ontology was outside the scope of this thesis. As the goal was to analyze the data to obtain new knowledge, and based on the size of the murine basal ganglia data set, using graph algorithms sufficed.

Graph algorithms

Graph algorithms are procedures that provide mathematically based measures on large and complex data. Based on graph theory, graph algorithms use information about relationships and nodes to infer an understanding of

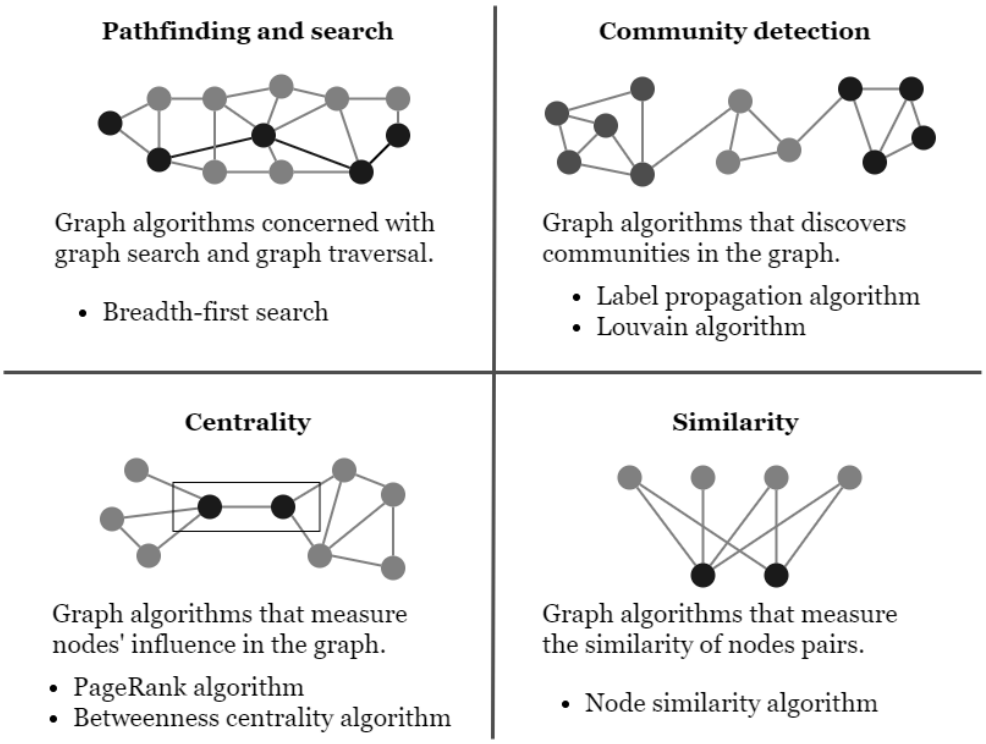

Figure 2.4: Four categories of graph algorithms.

the data [35]. Graph algorithms are typically categorized by the type of information they gather, and Figure 2.4 summarizes four such graph algorithm categories. The remainder of this section describes these four categories, including examples of algorithms.

A significant group of graph algorithms is pathfinding and search algorithms. Such algorithms are concerned with graph search by traversing the graph [35]. One such algorithm is breadth-first search (BFS). BFS traverses the graph in a breadth-first manner to find the shortest path between two nodes. The use of these algorithms is usually to find the shortest paths between node pairs and to specific nodes.

Community detection algorithms discover communities in a graph. Many graph representations, such as social networks, divide naturally into communities. These algorithms can help uncover the structure of the graph and group tendencies [35]. These algorithms define communities where nodes of a community have more relationships within the community than with nodes

outside of that community [42]. Examples of such algorithms are the Label propagation algorithm (LPA) and the Louvain algorithm. The LPA finds clusters based on labels, and the Louvain algorithm detects the communities by the concept of maximum modularity (to what extent equal nodes connect) [42].

Centrality algorithms measure which nodes are the most influential and have an extensive impact on the graph. There are multiple ways to measure the centrality of nodes. There are simplistic approaches, like counting the in- or out-degree of the nodes, and more advanced methods that take the dynamics of the connected nodes into account [42]. Examples of other centrality algorithms that take the entire graph's connectedness into account are the PageRank and the betweenness centrality algorithms. The PageRank algorithm, a previously central part of Google's web search engine, evaluates the nodes' direct influence while taking the influence of all the nodes into account [46]. The betweenness centrality algorithm measures the nodes' influence in the graph's information flow instead of measuring its direct influence [42]. These algorithms have many applications, such as finding relevant pages in a web search or influential scientists from publication databases [35].

The final group of graph algorithms described in this thesis is similarity algorithms. These algorithms measure the similarity of nodes by comparing node pairs [42]. There are many applications for finding similar nodes; for example, when studying a research paper, one might want to see similar papers. An intuitive similarity algorithm is the node similarity algorithm. This algorithm compares node pairs based on their neighboring nodes.

Many graph database implementations support some form of graph analysis. RDF graph implementations typically utilize reasoners to infer knowledge about the data [39]. For the property graph model databases, there are other solutions for data analysis. For example, the database implementation Neo4j provides a graph data science (GDS) library[6] that supports running graph algorithms directly on the graph data. ArangoDB, built to support big data, provides a machine learning infrastructure called ArrangoML[7]. In general, graph databases provide analytical tools that support analyses of the data they store.

[6] `https://neo4j.com/docs/graph-data-science/1.3`

[7] `https://www.arangodb.com/machine-learning`

2.1.5 Considerations for graph representation

When evaluating graph-based data representation, there are many considerations. The previous section has covered an introduction to graph database features, including graph data models and graph query languages, and approaches for graph data analysis. The following list summarizes the considerations of graph-based data representation:

- **The domain model**: Consider what the data represents and the structure of the data.
- **The graph data model**: Consider which data model the graph database exposes.
- **The graph query language:** Consider the language used to query the data for a given graph data model and database.
- **The graph database:** Consider the features of the graph database, including its implementation and its underlying storage, and possibly analytics capabilities.

From these considerations, we observe that the choice of a graph database management system depends on the requirements of the data model, query language, and potentially analytics capabilities. These requirements again depend on the domain model. As this thesis evaluates data management in the neuroscience domain, we need to understand the neuroscience aspects of the data.

2.2 Neuroscience data

To understand the brain's structure and function, researchers need data. In neuroscience, data primarily represent features of the brain and information related to brain-related research. To understand the domain of the murine basal ganglia data set, we present the relevant characteristics of neuroscience data in this section. Section 2.2.1 describes the anatomy of the brain and the structure of its cells to define what neuroanatomical data represent. Further,

the chapter presents how researchers work with brain-related data, including naming in Section 2.2.2, basal ganglia data in Section 2.2.3, and data quality in Section 2.2.4.

2.2.1 Anatomy of the brain

The brain is a large and complex organ that, together with the spinal cord, constitutes the central nervous system (CNS) [47]. Neuroscience typically divided the brain into different parts based on each region's functional, connectional, or structural properties. The exact division varies across the literature, but Kandel et al. (2000) specify six main parts [47]. These can be grouped into the three parts presented in Figure 2.5. One of these parts is the cerebrum, consisting of the cerebral cortex and subcortical nuclei. The cerebral cortex is the outer layer of the cerebrum and is responsible for most human cognitive abilities. The subcortical areas lie, as the name suggests, beneath the cortex. It consists of three compounds where one of them is the basal ganglia [47].

Many disease studies use rats or mice for their research as rodents have a shorter lifespan, and researchers can observe them in controlled environments. Although smaller in size, both the mouse and rat brain have a cerebral cortex and subcortical nuclei.

The brain includes numerous different cell types, broadly categorized as glial cells and neurons [48]. Neurons are the cells that process and transport information throughout the CNS. They communicate through connections

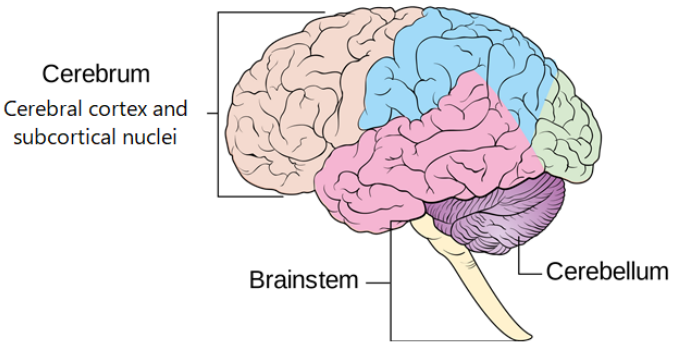

Figure 2.5: Anatomy of three main brain areas.

Credit: Cancer Research UK, CC BY-SA 4.0, via Wikimedia Commons (edited)

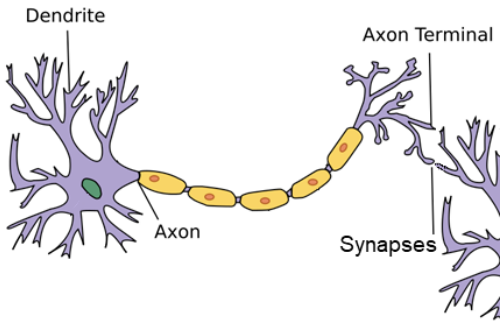

Figure 2.6: Central elements of a neuron.

Credit: User:Dhp1080, CC BY-SA 4.0, via Wikimedia Commons (edited)

called synapses. Figure 2.6 presents the central elements of a neuron. A neuron receives inputs, and when these reach a certain threshold, the neuron fires a signal through the axon that, via synapses, sends the signal to connected cells. The brain glial cells are non-neural, meaning they do not transfer signals directly. Instead, the glia cells provide support and regulate the functioning of the neurons [48]. Much of brain-related research investigates the cells in the brain.

2.2.2 Naming of brain regions and cell types

As presented in the previous section, neuroscience divides the brain into different regions. However, there are differences in the division, region naming, and which parts of the brain a defined region contain [49]. When a neuroscientist makes an observation, it is vital to communicate the observation's location in the brain [50]. In science, a nomenclature defines a system for naming within a specific area [51]. In neuroscience, a *brain region nomenclature* is a framework for naming and defining the areas of the brain. When studying the brain, such nomenclatures help researchers precisely define which region or part of the brain the data reference [52].

Neuroscience researchers utilize brain atlases for matching the location of their findings. This thesis refers to the term *brain atlas* as the more narrow description of atlases used for reference, also called reference atlases. A *reference brain atlas* is a map of the brain for a specific species, containing images of the brain and borders between regions in the context of those images [53]. In relevance to anatomical naming, reference atlases employ a specified

nomenclature [53]. The nomenclatures of the most renowned brain atlases at a given time are what researchers usually choose as nomenclature in a study or research experiment [54]. For example, when measuring cell-counts in a region, researchers can report which atlas nomenclature they have used to specify the given region. That atlas nomenclature is then the nomenclature used in that experiment or research. This reporting is essential for other researchers to obtain the correct location of the research observations.

Another area of anatomical naming considers cell types. Neuroscience research is often not concerned with counting or observing all neurons, but rather specific neurons, such as neurons which express particular neurotransmitters [55]. Researchers can name the neurons based on what they express, where they exist in the brain, or their structure, based on the research focus [10, 56, 55]. The many ways researchers can describe a cell type cause a lack of consensus on the criteria for defining neuron types [10]. For clarity, researchers should explicitly report what defines a specific cell type in their research [55].

2.2.3 Basal ganglia data

The basal ganglia are not a concrete part of the brain but a collective term for a group of nuclei. In humans and other mammals, the basal ganglia are significantly involved with movement and, to some degree, emotions, and memories [57, 58]. Figure 2.7 presents the nuclei of the basal ganglia together with related structures.

Much of the basal ganglia's clinical significance is related to movement disorders like Huntington's disease and Parkinson's disease [59, 60]. Basal ganglia studies are often related to specific diseases, producing a predominance of data about brain regions and cell types relevant to the disease.

Scientific researchers generally process data at different levels, which is also the case for basal ganglia data. This thesis categorizes the research data into three levels: (1) raw data, (2) derived data, and (3) metadata. Raw data entails direct research measurements, while derived data denotes the processed results drawn from the raw data. Metadata defines the characteristics of this data, being the "data about data." The *raw data* is the non-processed result of an experiment measure. Raw data can be neuroimages, electrode recordings, or other direct measurements. Researchers either base their results on the raw data, or this data is the result itself. Researchers are often interested in more than raw data and thus analyze the raw data to provide insights. This

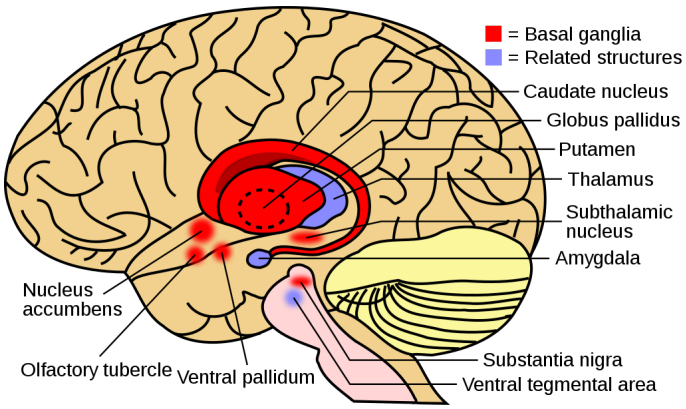

Figure 2.7: The nuclei of the basal ganglia and related structures.

Credit: File:BrainCaudatePutamen.svg: User:LeevanjacksonDerivative work: User:SUM1, CC BY-SA 4.0, via Wikimedia Commons

process yields the *derived data*. Examples of derived data are quantitations (objects of interest counts), distributions, or morphologies (an object's physical structure). In neuroscience, the *metadata* covers all the information related to the experiment and can include data about the methodology, specimens, and specific chemical solutions of the research. Published basal ganglia research typically presents the results (derived data), some degree of metadata, and in a few cases, the raw data [19].

Before the murine basal ganglia database, the existing basal ganglia related data was mostly available in research papers. These research papers were the basis for the data that Bjerke et al. (2019) have gathered into a collective database [19]. Of the data levels presented in Section 2.2.3, basal ganglia related data can be at all levels. However, the quality of the data varies. There can be a lack of metadata and also appropriate result data. The basal ganglia related research papers often present the results in either text, tables, or graphs. In the murine basal ganglia data set, Bjerke et al. excluded the papers that represented the results in graphs and the papers where certain predefined metadata were missing [19].

2.2.4 Data quality

The quality of research data is essential for researchers' ability to utilize the data. This thesis defines data quality as the data's ability to be used in the

intended context. Specifically, for neuroscience research, this entails being understood and possible to reuse by other researchers.

Data quality is essential in all the data levels presented in Section 2.2.3, but with different impacts. With raw data, the data quality impacts the extent to which other researchers can process the data. For high quality, the raw data should, whenever achievable, be machine-accessible and in a standardized format; unfortunately, this is often not the case [8]. To have high quality derived data, it must be clear how the researcher obtained the result. Researchers can achieve this by presenting the raw data and the specific features of the experiment or study. Thus, the quality of the derived data highly depends on the quality of the raw data and metadata. The metadata should include all the information relevant to understand, combine, and reproduce the result. Hence, all three data levels, raw, derived, and metadata, are interdependent and should be provided together to obtain high data quality.

The data quality in neuroscience is affected by the naming aspects. As stated in Section 2.2.2, there are different ways to divide the brain into regions, and researchers can employ brain region nomenclatures to clarify which region division they employ in their research. However, much research presents unstandardized terms when referring to brain regions [61]. As brain atlases continually update due to the progress in neuroscience, the older nomenclatures become outdated. There is no complete mapping data between different brain region nomenclatures, and there is still no standard format for cell naming, although research efforts investigate this [10, 50]. In summary, the challenges with data quality in neuroscience entail the lack of standard formats, the possibility to map between various nomenclatures, incomplete information, and a lack of proper metadata [61].

2.3 Brain-related data management

Although there are challenges related to the understanding and reuse of existing neuroscience data, several initiatives work to share data, collaborate, and advance brain research. Many of the initiatives are complementary and attempt to build on each other's data. Moreover, there are areas in neuroscience that utilize graph-based data representation. Section 2.3.1 presents initiatives types that provide neuroscience data before Section 2.3.2 evaluates graph-based approaches in neuroscience.

2.3.1 Types of initiatives for neuroscience data

The first initiative type considered in this thesis is the *repositories of data sets*. This type entails initiatives that collect and provide available research from multiple sources, including publications and data sets. There are many such initiatives where some are general-purpose, and others are specialized for a specific research area or set of data [15, 62, 16, 13]. Some of these initiatives also work with managing metadata to make it easier to navigate their data. What is common amongst these initiatives is their goal to make neuroscience research more available and accessible to deal with the data quality challenges.

Another essential brain-related initiative type is *brain atlases*. As presented in Section 2.2.2, we narrow the atlas scope down to reference brain atlases, which are maps of the brain, including defined brain region borders. Researchers use these atlases as reference tools to answer questions about location in the brain [53]. In comparison to the repositories of data sets, atlases come from one data source. Like the data repository initiatives, brain atlases function as a tool for researchers to analyze their research. Although brain atlases do not integrate research data directly, they are essential for neuroscience data integration as they provide location references for research and standardization of these locations [50].

The final initiative type we consider is *neuroscience databases*. A neuroscience database is broadly a database consisting of brain-related data. These initiatives combine data from one or many sources, such as research papers or other databases, and provide this data integrated into a common database. The data can be at any or all of the data levels presented in the previous section. The murine basal ganglia database is an example of such an initiative [18].

2.3.2 Graph-based approaches

As this thesis investigates graph-based data representation in the neuroscience domain, it is relevant to evaluate the currently existing graph-based approaches. Graph-based data representation in neuroscience has primarily focused on knowledge graphs for organizing research and networks for the brain's neural connections.

The brain-related initiatives EBRAINS and KnowledgeSpace utilize knowledge graphs to enrich the data and improve their search engines that retrieve

research papers and data sets [12, 13]. The EBRAINS Knowledge Graph, previously known as Human Brain Project (HBP) Knowledge Graph, is a metadata management system. The system utilizes the knowledge graph by adding metadata to neuroscience data sets so that researchers can categorize, filter, and search by keywords regarding features like data types, species, publication year, and experiment methods [12]. The returned result provides a description of the source together with metadata information in a standardized format. KnowledgeSpace is another initiative that employs a knowledge graph for research data. Differing from EBRAINS, KnowledgeSpace combines brain research concepts from multiple sources with data, models, and literature. KnowledgeSpace collects concepts from, among others, InterLex and Wikipedia [13]. They combine these concepts with data from other neuroscience initiatives. KnowledgeSpace presents an architecture diagram[8] displaying that they use Neo4j, amongst others, for graph data management. EBRAINS does not present the knowledge graph's technology or architecture but reveals that they use a graph database. In summary, both initiatives provide powerful graph models that simplify data discovery. However, they do not change or connect the data in the papers, data sets, and models contained in the knowledge base.

Another direction of computational neuroscience that utilizes graph principles is the study of neural connections in the brain, called connectomics. Connectomics is an extensive research field, including numerous research papers and large initiatives, such as the five-year Human Connectome Project [63]. In this research field, the goal is to create a map of how neurons connect in the brain, and researchers can investigate subsets, or the entire brain, and investigate functional or structural connections. Figure 2.8 displays a human connectome collected from a research project from 2014, investigating the difference between the brain's structural and functional networks [64]. Connectomics presents an example of data that naturally structure as a graph, and that can benefit from graph-based data representation.

Connectomics use a wide range of methods to obtain data, from tract-tracing techniques in animal models to functional magnetic resonance imaging (fMRI) in humans. For mice and rats, the methods can achieve spatial accuracy down to single neurons, but the techniques used on humans produce less specific data. Due to numerous techniques that produce data at different accuracy levels with varying spatial and temporal resolution, there are some

[8] `https://knowledge-space.org/ks-architecture`

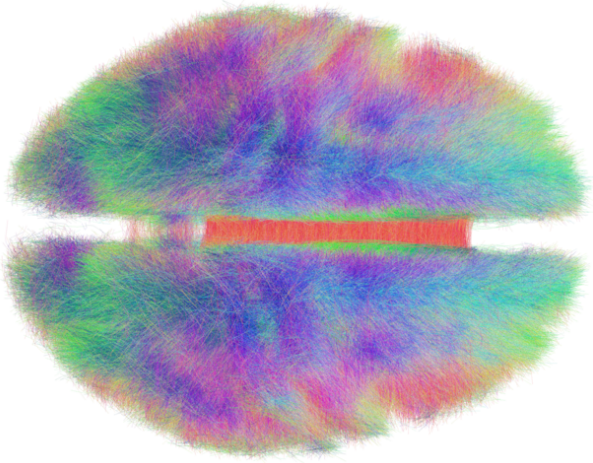

Figure 2.8: A human brain connectome.

Credit: Andreashorn, CC BY-SA 4.0, via Wikimedia Commons

disputes around the data quality and results in connectomics [65]. A connectome has no direct reference to the employed techniques, as the creator of the model has fully integrated the raw data into a complete model.

We can imagine the two main approaches to graph-based data representation in neuroscience as two sides of a scale. On the one side, we have the knowledge graph approaches that utilize graph models for managing research metadata to integrate multiple data sets. On the other side, we have connectomics that integrates research data into a complete model and remove all metadata references. On this scale, this thesis places its research somewhere in the middle, somewhat closer to the first point, as the focus is on the data of a specific research set, not purely metadata, but not purely the research findings either.

Many neuroscience data initiatives share the same purpose as this thesis; to make it easier to access, integrate, analyze, and share data. Some also utilize graph models, such as EBRAINS and KnowledgeSpace, that integrate data from multiple sources into a unified repository. However, none of the models integrate the research data from data sets while maintaining the metadata, which is the aim of this thesis. By this, the related work in these initiatives is complementary to the research of this thesis. Hopefully, the thesis research can define a proof-of-concept on how other initiatives could achieve the same goal.

Chapter 3

Problem analysis

Based on the background from Chapter 2, this chapter analyses the thesis problem space guided by the three research questions presented in Section 1.3, resulting in the thesis artifact requirements. After presenting an overview of the problem analysis in Section 3.1, this chapter chronologically describes the process of defining the requirements for the software solutions developed to evaluate the research questions.

3.1 Overview of problem analysis

This thesis aims to evaluate graph-based data management in the neuroscience domain, exemplifying the process with the murine basal ganglia data set, consisting of quantitative neuroanatomical data of the basal ganglia in rats and mice. We have gained relevant information to specify a graph model, define an evaluation guided by the thesis research questions, and understand the domain challenges. We know there are multiple sources for neuroscience data, multiple formats to store research data, and data standardization and quality challenges. As a basis for the problem analysis, we desired to understand these challenges from the researchers' perspective.

We created a survey to understand how researchers work with publicly available neuroscience data, including which challenges they experience. Appendix C presents this survey, together with the survey results. In the survey, we asked for the researchers' background to separate the desires of different research areas. The survey lists a range of data repositories, asking on a Likert scale how often the researchers use them. Furthermore, the survey asked which tasks the researchers usually perform with the data before listing a range of potential challenges and asking, on a Likert scale, to what extent they find them challenging. The following list presents the challenges of using public neuroscience data we posed in the survey:

1. Finding relevant data
2. Reusing data
3. Connecting data across multiple sources
4. Understanding the data structure and format
5. Understanding the method used to produce the data
6. Making use of all the available data
7. Finding enough data
8. Finding data of high enough quality

Fourteen neuroscience researchers with various backgrounds answered the survey. To obtain suitable respondents, we forwarded the survey to neuroscience institutions at three Norwegian universities. The responding neuroscientists had a background mainly in biology, medicine, and physics (see Appendix C). Although the number of respondents does not provide statistical results, it provides pointers to what these researchers find challenging. Further, as we evaluate these results combined with their use of data and data sources, and as the respondents come from independent institutes, the survey provided valid results.

From the survey responses, we observed that most of the respondents use the data to compare results or findings and address new hypotheses. Further, most respondents found finding data of high enough quality and making use of all the available data very challenging. When analyzing the thesis problem space and defining the solution requirements in this chapter, we considered the survey findings.

Summary of survey findings

- The respondent researchers predominantly use publicly available data to *address new hypotheses* and to *compare results and findings*.
- *Finding data of high enough quality*, *connecting data from multiple sources*, and *understanding the data structure and format* were found the most challenging.
- The remaining data challenges proposed in the survey were found *very challenging* by at least one respondent.

The thesis hypothesis posits that presenting the data in a graph model will give researchers a better understanding of the data, improve the usability of the data, and provide an intuitive way of integrating the model with existing data, answering the challenges researchers experience. The first step to evaluate the hypothesis was to understand the thesis data set and define the graph model and the database system's requirements. Section 3.2 displays our analysis of the data set, and Section 3.3 presents our analysis and defined requirements for the graph model and suggestions for a graph database implementation. Further, the chapter presents the analysis of each component we developed to evaluate the graph model, based on the three research questions:

RQ_1 Section 3.4 presents our analysis of related neuroscience initiatives, and of which sources and what data we could integrate with the thesis data set.

RQ_2 Section 3.5 presents the approach and requirements for graph data analysis, which we employed to observe if it is possible to obtain new information about the data.

RQ_3 We desired to evaluate how web-based data access affects the thesis data set's usability. Section 3.6 presents our analysis of the data set's usage and users and the software artifacts' functional requirements, together with requirements for measuring the solution's usability.

Summary of how the thesis research analyzed the thesis problem space

- Analyze the data in the murine basal ganglia data set.
- Define requirements for the graph model.
- Define suggestions for a graph database management system.
- Define external data that can integrate with the thesis data set by analyzing initiatives that publicly provide neuroscience data.
- Define requirements for data exploration using graph analysis.
- Define requirements for web-based data access and a usability study.

3.2 The murine basal ganglia data set

The murine basal ganglia data set is the data in a database created by Bjerke et al. in 2019, consisting of quantitative neuroanatomical data about the healthy rat and mouse basal ganglia, collected from more than 200 research papers and data repositories [19]. The data set contains three distinct information types: quantitations (counts), distributions, and cell morphologies. The counts and distributions regard either entire cells or specific parts of the cell, while the morphologies describe the cell's physical structure. The data set is publicly available through EBRAINS as an Access database (.accdb) and as comma-separated values (CSV)-files [18]. The data set's primary usage is for researchers to find and compare neuroanatomical information about the basal ganglia brain regions. In addition to a data set, Bjerke et al. (2020) published a paper describing the data set development process and their findings [19].

Referencing the basal ganglia data presented in Section 2.2.3, the data set contains metadata and derived data. The murine basal ganglia data set contains *metadata* about the experiments, analyses, and specimens, connected to experimental results and *derived data*, representing cell counts and cell structures in specific regions. Moreover, it contains general cell types and brain regions used to reference the derived research results. All the data in the murine basal ganglia data set are in a tabular format. The brain regions in the data set come from two nomenclatures, provided by the Waxholm Space (WHS) rat brain atlas for the rat species and the Allen Mouse Brain Atlas

Figure 3.1: The structure of the original murine basal ganglia database, presented as a graph.

(AMBA) for the mouse species. These brain atlases are further described in Section 3.4. The data set does not contain raw data or externally referenced files with data.

Figure 3.1 illustrates the murine basal ganglia database directly represented as a conceptual graph, where the nodes represent the database tables, and the edges represent the foreign keys. In this figure, the graph is simplified by omitting the table columns. The figure denotes the table names using

pascal-case, directly transformed from the original table names written in snake-case. This thesis will refer to the tables in the pascal case format and, from here on, refer to the original tables as nodes.

The murine basal ganglia data set consists of specific experiment results connected to predefined brain regions and cell types and detailed information about the experiment [19]. The data set has a hierarchical structure: A source reports one or many experiments. An experiment has one or many derived data records describing a specific analysis performed in the experiment. Moreover, the derived data records, from now called analyses, relate to one or more data types. These relations between analyses and data types describe specific measures of the analysis. The data types are either quantitations, distributions, or cell morphologies. For a complete entity-relationship (ER) diagram of the original database and for more details about the data set data, see Appendix A.

From Figure 3.1, we can observe the data structure, including the connectivity. The nodes are marked with one of four colors. These colors represent four node-categories that we derived from investigating the data and from discussions with Bjerke. The following list presents these categories with associated colors:

- *Experiment data* (purple): The nodes representing experiments and the related experiment data.
- *Sources of information* (green): The nodes representing external sources of information. This category includes the sources (journals) that published the experiments and the nomenclatures used to define the brain regions.
- *Specimen data* (yellow): The nodes representing the experiment's specimens and the properties of these.
- *Neuroanatomical data* (orange): The nodes representing neuroanatomical data about brain regions and cells with classifications and areas.

This data set is suitable for a graph model due to its connectedness that cannot be fully represented in a relational model, for example, direct relations between nodes to promote easier data access, navigation, and analysis.

Further, a graph model is flexible and can easily adjust to new data. This flexibility benefits the murine basal ganglia data set as its developers want the data set to be continually expanded with new data. Thus, we decided to use this data set as the basis for investigating graph-based data representation.

3.3 Requirements for the basal ganglia data set graph model

Remembering the general idea of the thesis, the graph model is the foundation for all the other artifacts, and this is where we began the requirement specification. This section describes the graph model requirements and suggestions for data storage, based on the analysis of the murine basal ganglia data set.

When specifying a graph model, the structure is fundamental, as it affects the searchability and usability of the data and the results of different graph analysis. As stated in Section 2.1.2, a graph data model stored in a graph database is dynamic, compared to a relational model, as the graph data model can be isomorphic to the domain model, implying that the model is satisfactory as long as it covers the users' needs [6]. This fact yields that the first step for designing the graph model was to define the potential user objectives and scenarios for accessing the data.

In the paper presenting the murine basal ganglia database, Bjerke et al. (2020) proposed a user workflow for the database that includes three user research scenarios [19]. Appendix A presents these. From these user scenarios, we list three specific use cases of the data:

- Researchers who want to find, explore, and use the data to model the basal ganglia.
- Researchers who want to find, explore, and use the data to compare it with their experiment results.
- Researchers who want to extend the data set with data from their experiments.

Based on these use cases and interviews with Bjerke, we defined the following user objectives for accessing the data in the murine basal ganglia data set:

- When studying a specific brain region or investigating a specific cell type.
- In search of similar experiments for comparing results.
- To extend the data set with new data.

The previous section categorized the nodes in the basal ganglia data set into one of four categories. For faster retrieval and a better overview of the data, the graph model can group the nodes of the same category to facilitate combined retrieval. Based on interviews with Bjerke and the presented categories, it seems reasonable to assume that when a user accesses a node from a category, the user will also access more nodes from the same category. For example, if a user is looking at an experiment, the information that details the experiment would naturally be associated and relevant. Thus, the final user objective formulates that researchers desire to find data within the same category together.

Based on the graph model use cases, we defined the graph model's requirements, presented in the following list. These requirements state that the graph model should represent the domain model, including connecting the same category nodes and provide multiple entry points to access the data. For the final requirement, we specified the analyses as a primary entry point rather than experiments, as the experiments in the data set are represented through analyses.

1. The graph model must follow the domain model such that a researcher can easily find and compare experiment data.
2. The model should connect nodes within the same category together.
3. Data should be easily reachable from three primary access nodes: cell types, brain regions, and analyses.

The second aspect of the graph model regards the graph database management system (GDBMS) that stores and represents the data. The database system itself is not the most crucial factor; instead, it is the selected system's features. Although this thesis evaluates the generic use of a graph model, rather than a specific management system, the selected system plays a role

in the graph model's development and use. As an outcome, we defined the following set of suggestions for the GDBMS:

- The GDBMS should implement native graph storage (see Section 2.1.2).
- The GDBMS should include programmatic procedures for accessing and inserting data.
- The GDBMS should incorporate graph analytics capabilities.
- The GDBMS should support dynamic updates and data retrieval.
- The GDBMS should be well maintained and have proper documentation.

3.4 Analysis of integration with related data

Will a graph representation of brain-related data facilitate the integration of data from a variety of neuroscience data sets? The first research question, RQ_1, asks how the graph model facilitates the integration of data from external neuroscience data sets. This section presents an evaluation of related initiatives that publicly provide neuroscience data to identify data that overlaps with the murine basal ganglia data set for integration with the graph model. This section does not include a complete overview of all available neuroscience data initiatives and does not perform an in-depth examination of each source. Instead, it provides an examination of what data we can obtain and integrate from such initiatives.

3.4.1 Review of initiatives for neuroscience data

The first step in this process was to define potential sources of data. First, we evaluated the repositories of data sets, as these contain large amounts of brain-related data. Next, as the data set has defined nomenclatures for the rat and mouse brain, we investigated the atlases providing these. Furthermore, the data set references two neuroscience databases as sources of information that are relevant to consider. Table 3.1 presents the initiatives this thesis examined, together with the relevance of each initiative.

Initiative	Description	Relevance
EBRAINS[1] *Repository Multiple species*	The Human Brain Project (HBP), initiated in 2013 and funded by the European Union (EU), aims to build infrastructures that aids and improves neuroscience, computing, and brain-related medicine research [15]. The HBP has delivered EBRAINS, a brain research infrastructure. EBRAINS includes tools that aim to address challenges in the field by assisting in collecting, analyzing, and sharing data and brain function modeling and simulation [12]. To promote data accessibility, EBRAINS provides a knowledge graph for searching all available data.	Relevant as it is a well-known platform for finding neuroscience data.
Neuroscience Information Framework[2] *Repository Multiple species*	Available since 2006, the Neuroscience Information Framework (NIF) offers services to search among an extensive collection of neuroscience information [16]. The NIF has gathered multiple data sets and allows the researcher to search across all available data, where the data is clearly categorized. The data includes information from other initiatives, like the Allen Institute and Brain Architecture Management System (BAMS). For example, NIF hosts a version of the BAMS database. NIF provides the database from BAMS containing tables with brain regions and cell type metadata, with data about the name of brain regions and cells, including the nomenclature.	Relevant as it is a well-known platform for finding neuroscience data.
Zenodo[3] *Repository Multiple species*	Zenodo is an online repository of research (publications and data), launched in May 2013 by CERN [62]. Although this initiative is not specific to neuroscience research, we presented it as a source for neuroscience data because it might provide relevant data in any research field. However, as Zenodo is not specific to a domain, it does not contain domain-specific metadata for navigating the data.	Relevant as it is a well-known platform for finding research data in any field.

[1] `https://ebrains.eu`
[2] `https://neuinfo.org`
[3] `https://zenodo.org`

Knowledge-Space[4] *Repository Multiple species*	KnowledgeSpace is an encyclopedia for neuroscience and combines general descriptions of neuroscience concepts found from Wikipedia and other neuroscience specific sources [13]. The initiative combines content from these sources with neuroscience research found in other repositories. KnowledgeSpace contains a broad range of neuroscience data, including the definition of neuroscience concepts and cell types, and incorporates the NIF ontology for metadata.	Relevant as it is a source for neuro-anatomical definitions.
Waxholm Space (WHS) rat brain atlas[5] *Atlas Rat*	WHS is a brain atlas of the Sprague Dawley rat brain [66]. The entire set of images that comprise the WHS atlas is available through EBRAINS. Additionally, the EBRAINS platform offers an interactive Atlas Viewer to explore the 3D WHS atlas. The latest version of the WHS data set consists of anatomical delineations of 118 brain regions observed in neuroimages [67].	The murine basal ganglia data set uses the nomen-clature provided by WHS for the rat brain.
Allen Institute for Brain Science[6] *Atlas and repository Human, Mouse*	This initiative provides multiple brain atlases for both the human and mouse brain, including a mouse brain atlas (AMBA) and a mouse brain connectivity atlas [68, 69, 70]. The mouse brain atlas provides data about mouse brain structure. Further, they provide many developer tools in the Allen SDK.	For mice, the thesis data set uses the nomencla-ture provided by AMBA.
The Blue Brain Cell Atlas (BBCA)[7] *Data set Multiple species*	BBCA is a part of the Blue Brain Project and provides an interactive cell atlas that includes information about cell densities and positions in the brain regions of the mouse brain [71, 72]. Although it is called an atlas, it is not a reference brain atlas used for brain location. We refer to BBCA as a data set as it offers cell counts for the mouse brain. The BBCA provides information about cellular compositions in the mouse brain.	Relevant as it contains cell data and uses the nomencla-ture from AMBA.

[4] `https://knowledge-space.org`
[5] `https://www.nitrc.org/projects/whs-sd-atlas`
[6] `https://alleninstitute.org/what-we-do/brain-science`
[7] `https://portal.bluebrain.epfl.ch/resources/models/cell-atlas`

Brain Architecture Management System (BAMS)[8] *Data set* *Rat*	BAMS aimed to be an online knowledge management system to store and infer relationships between data about the structural organization of nervous system circuitry [73]. The website, although no longer maintained, contains information about rat brain regions and structural brain region connectivity.	A data set with brain region data, suggested by neuroscientists at the University of Oslo.
NeuroMorpho.Org[9] *Data set* *Multiple species*	NeuroMorpho.Org aims to provide access and overview of available morphological reconstructions and has a database containing multiple data sets with digitally reconstructed neuronal morphologies with unique identifiers [74]. More than 500 laboratories have contributed data, and the database is continually updated with new reconstructions.	Some cell morphologies in the murine basal ganglia data set have such identifiers.
InterLex through SciCrunch[10] *Repository/ Data set* *Multiple species*	SciCrunch comprises tools, resources, and databases to make data FAIR (Findable, Accessible, Interoperable, and Reusable) [75]. The data sets found through SciCrunch are much of the same as the ones found in the NIF, as SciCrunch also provides the NIF data. SciCrunch contains InterLex, a dynamic lexicon of biomedical terms aiming to improve the researcher's communication about data. It works as an ontology on top of existing terminologies and ontologies, and is incorporated by both SciCrunch and the NIF [76].	The cell types in the murine basal ganglia data set have ontological identifiers that are accessible through InterLex.

Table 3.1: Review of initiatives the publicly provide neuroscience data, including their relevance to this thesis

[8] `https://bams1.org`
[9] `http://neuromorpho.org`
[10] `https://scicrunch.org/scicrunch/interlex/dashboard`

3.4.2 Analysis of initiatives

The next step of the process was to analyze the initiatives presented in Table 3.1 for integration with the thesis data set. We investigated each of these initiatives to find potential overlap with the murine basal ganglia data set. The overall methodology to investigate the initiatives was to visit the data source and look for available data sets and programmatic data access. For the initiatives that provided data programmatically, we searched with the term "basal ganglia" and some specified basal ganglia related regions. Where we managed to obtain relevant data, we consulted Bjerke to evaluate if the data was related to the basal ganglia. Further, we evaluated if the data was overlapping with the murine basal ganglia data set. If all of this was verified, the data was applicable to be integrated into the murine basal ganglia graph model. In summary, we analyzed each initiative against the following criteria:

1. Serve data programmatically
2. Have data related to the basal ganglia
3. Provide data that is extendable with the data in the thesis data set

Figure 3.2 summarizes our investigation, presenting which data sources we can collect data from to extend the graph. It is important to note that this is not an in-depth analysis. Some of the initiatives that we found unsuited for integration might fulfill all the criteria, but not in a way we managed to observe at the time of the thesis research.

Analysis of the initiatives that satisfied the criteria

Brain Architecture Management System (BAMS): The BAMS website, although no longer maintained, contains information about brain regions and structural connectivity information in the rat brain. A researcher first selects the direction of the connection on this website, either input (afferent) or output (efferent). Next, the researcher searches for a brain region, and the website returns all the connected regions. The returned connectivity results contain a reference to the article where BAMS' creators collected the connectivity information.

As presented in Section 2.2.2, data sources can define a nomenclature for a species to specify brain region location. In BAMS, the chosen nomenclatures

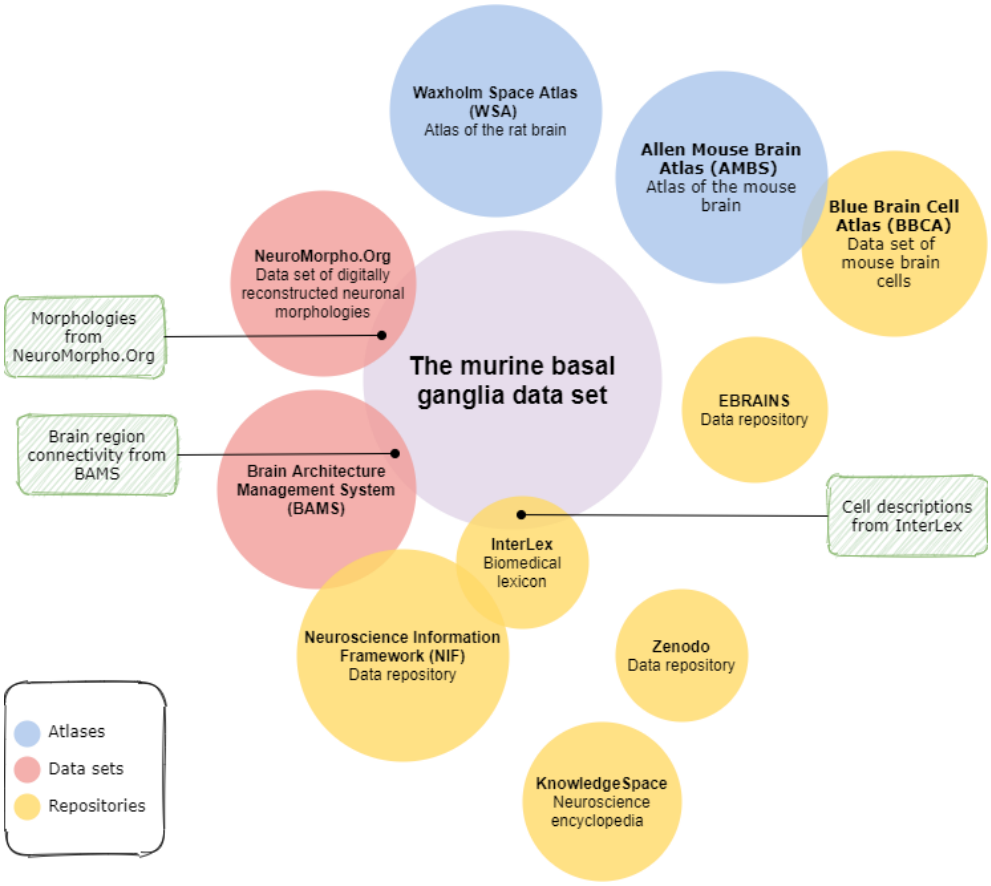

Figure 3.2: Overview and result of initiatives investigated for data overlap with the murine basal ganglia data set.

differ from the rat nomenclature in the murine basal ganglia data set. When we found data overlapping with the murine basal ganglia data set, we also needed to map between the nomenclatures.

Analyzed against the presented criteria, BAMS complies with two of the three. It contains basal ganglia related information extendable with the murine basal ganglia data set; however, it does not provide the data programmatically. For the scope of this thesis, we wanted to obtain the overlapping data, and as the BAMS website presented the data in tables, we could access the website programmatically and extract the data from these tables. With the adjusted method, BAMS answered all the presented criteria.

InterLex through SciCrunch: SciCrunch provides an application pro-

gramming interface (API)[11] where one can get descriptions based on InterLex's ontological identifiers. In the murine basal ganglia data set, many of the cell types have ontological identifiers recorded. We could use these identifiers to connect the information to the cell types in the database. This data source then complied with all the presented criteria. They provided an API and had data related to the basal ganglia overlapping with the thesis data set.

NeuroMorpho.Org: NeuroMorpho.Org provides an API[12] where researchers can find a neuron by id or name. With the inspiration of looking for identifiers in the murine basal ganglia data set, we observed that the nodes with cell morphologies (the structure of the cell) also had identifiers for the neurons mapping to NeuroMorpho.Org. By this, NeuroMorpho.Org complied with the presented criteria.

Evaluation of the initiatives that did not satisfy the criteria

EBRAINS: The EBRAINS platform provides a knowledge graph search where researchers can find data based on multiple metadata properties and download data. When this thesis looked into EBRAINS in early 2020, searching with the term "basal ganglia" only returned the murine basal ganglia database by Bjerke et al. (2019). At the time of the research, EBRAINS did not answer the final criterion; however, other researchers could look into newly added research data sets in further investigations.

The Neuroscience Information Framework (NIF): Searching the NIF for connectivity information about the basal ganglia did not return any relevant results. Although the NIF hosts a version of the BAMS database, it was not the databases with connectivity information. The initiative did not comply with the third criterion, as there was no data we could combine with the murine basal ganglia data set available in early 2020.

Zenodo: Zenodo provides an API where researchers can perform advanced searches for records. In search of relevant records, we performed API calls with the search term "basal ganglia" and record type "dataset." This search returned many records. However, looking more closely at the results, most of the hits only contained the word "basal" and were not related to neuroscience. Changing the search term to be only "ganglia" returned only one hit per April 2020. This data set contained RNA sequences and was not overlapping with

[11] `https://scicrunch.org/browse/api-docs/index.html?url=https://scicrunch.org/swagger-docs/swagger.json`

[12] `http://neuromorpho.org/apiReference.html`

the cell types in the murine basal ganglia data set. We excluded Zenodo for further investigation as it did not have any data related to the basal ganglia at the time of this research.

KnowledgeSpace: Searching the term "ganglia" returned 56 records per August 2020. Unfortunately, KnowledgeSpace did not provide any API to access the data programmatically. However, they did provide their source code on GitHub[13], and the entity client code displayed an endpoint for searching by "slugs." Testing this endpoint showed that it returned results for standard terms, such as "neuron" and "dopamine." However, as naming is not consistent in neuroscience, it was unsuitable to use the endpoint to collect definitions about regions and cell types without unique identifiers. Therefore, KnowledgeSpace did not satisfy the first criterion presented in the context of this thesis.

Waxholm Space (WHS) rat brain atlas: From the WHS website, it was possible to download the complete atlas as neuroimage files. There was, however, no other data available. Neuroimages of rat brain sections do not overlap with the data in the thesis data set. By this, Waxholm space atlas did not meet the defined criteria of providing data extendable with the murine basal ganglia data set.

Allen Institute for Brain Science: Using a Jupyter Notebook provided by Allen Insitute, we retrieved cell type data that included the brain regions where researchers have observed the cell type. We filtered out the unique brain regions, and Bjerke examined this list to see if any of these regions were related to the basal ganglia. Unfortunately, they did not. Allen Institute did not meet the defined criteria of providing data related to the basal ganglia and overlapping with the murine basal ganglia data set.

The Blue Brain Cell Atlas (BBCA): The Blue Brain Cell Atlas was constructed using data from the Allen Mouse Brain Atlas, both for region information and to derive cell counts. Nonetheless, the initiative did not provide any data programmatically as far as we could find, thereby not complying with the first criterion.

[13] `https://github.com/OpenKnowledgeSpace/KnowledgeSpace`

3.5 Requirements for data analysis

The second research question, RQ_2, asks about data understanding in a brain-related data set. Understanding is subjective, so we define the baseline knowledge about the database on the findings found by Bjerke et al. (2020) [19]. To examine the research question, we investigated what new information we can extract from the data by utilizing graph-based data analysis. The assumption is that if the graph model can provide findings that are not evident in the relational database, the model improves the understanding of the data.

The first step in specifying the data analysis requirements was to define the thesis' data analysis approach. Data analysis defines all methods that break down data into significant components. Many data analysis approaches involve statistics, but this is a traditional approach rather than a data analysis requirement [77]. Data analysis approaches can be separated into *confirmatory* and *exploratory* data analysis. *Confirmatory data analysis* is concerned with proving or answering a specific hypothesis or question [77]. The researcher or analyst has a question about the data, such as answering if the data proves a correlation between two explicit factors. Then data analysis is used to answer that specific question. *Exploratory data analysis* is concerned with exploring the data, looking for general information and clues [77]. Instead of asking if the data proves something, the approach seeks to investigate what the data can tell us. This approach utilizes any method that can provide information about the data, such as data visualization. As this thesis is interested in all information possible to obtain using graph-based methods, both exploratory and confirmatory data analysis can prove useful.

As the data volume in the murine basal ganglia data set is relatively small, we chose to apply graph algorithms as the primary data analysis tool. Section 2.1.4 presented four graph algorithm groups that all might provide useful information about the data. Because path-finding and search algorithms are more suitable for specific scenarios than data exploration, we decided that the exploratory data analysis approach should employ the remaining three algorithm categories: clustering, centrality, and similarity algorithms.

Further, the thesis research utilized specific use cases for data analysis based on the confirmatory approach. In search of specific answers, we needed to define relevant questions about the data, noting what information would be valuable to retrieve. From the research survey, we observed that researchers

use publicly available data to validate findings, compare results, and address new hypotheses. Based on this, we defined the first use case to find similar analyses. It seemed reasonable that researchers would be interested in finding other research similar to what they are investigating in their search. A second use case came from an interview with Bjerke. When she created the murine basal ganglia data set, one of the original research objectives was to investigate a correlation between method and quantitative results. Her question was: "*Is it possible to say anything about the methods the researchers used in the experiments when counting a cell type in a specific region that correlates to the result?*". This question defined the second use case. The approach for the confirmatory data analysis part of this thesis was to answer these two use cases, aiming to evaluate how a graph-based data representation aids in answering them.

Summary of thesis data analysis approaches

Exploratory data analysis approach: Aiming to acquire any information about the data.

1. Investigate clusters in the data
2. Find central (influential) nodes in the data
3. Compare similarity between nodes

Confirmatory data analysis approach: Aiming to obtain specific information defined in use cases.

Use case 1: Obtain similar analyses, based on brain region, cell type, and object of interest.

Use case 2: Observe what we can say about the correlation between research methods and results in quantitative experiments.

To evaluate the extent to which the data analysis findings provide new information about the data, we needed to evaluate the findings. We decided that Bjerke, who knows the data well, was suitable for such an evaluation.

Further, we defined three categories for the evaluation, where each finding should be categorized as either:

- *Already known*: The findings that were already clear from the original database.
- *Expected, but now known*: The findings that were not evident in the data, but information that researchers might assume from other domain knowledge.
- *Unexpected*: The findings that were not known and not expected.

3.6 Web-based data access requirements

The third research question, RQ_3, asks whether we can improve the data set's usability by developing web-based access to the data. To evaluate this research question, we chose to specify applications for web-based data access through both a user interface and programmatic access. To evaluate the research question, we also decided to analyze the usability of the user interface. Section 3.6.1 describes our approach for understanding public neuroscience data usage based on the survey results, and Section 3.6.2 and Section 3.6.3 present the resulting functional requirements and usability study requirements.

We recognize that applications that provide web-based data access do not directly represent the graph model and could retrieve data from a relational database and a graph database with the same end-result. However, we attempted to visualize the graph navigation through the website and API such that the data follow the graph model's structure, including the connectedness. Additionally, this thesis demonstrates the steps of building a web and an API application to provide web-based access to the graph database, aiming to illustrate how other researchers can do the same.

3.6.1 Understanding the usage

To define the user interface requirements, we needed to understand the usage of the data. In this effort, we used the responses from the thesis survey, fully described in Appendix C. Section 3.1 presented the areas that researchers find challenging based on this survey. In the survey, we also asked what publicly

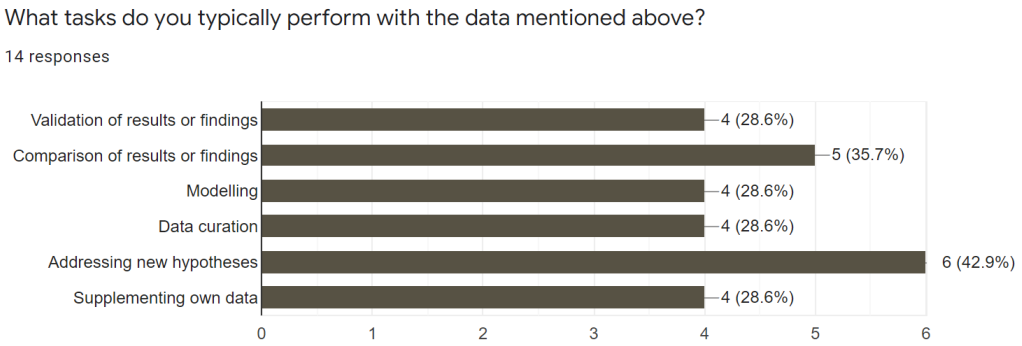

Figure 3.3: Results from the survey for understanding the usage of publicly available neuroscience data.

available data they work with and how frequently. Figure 3.3 presents the survey outcome of the question asking what tasks researchers typically perform with publicly available neuroscience data. From this survey finding, we observe that many researchers use the data for further research, and in addition to reusing the data, they might want to integrate the data into their work.

Before defining the graph data user interface and evaluation requirements, it is necessary to define usability. A widely used standard for usability testing is the ISO 9241-11 standard [78, 79]. This standard defines usability as

> *"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use."*

This definition specifies three essential elements for usage: the specified users, the specified goals, and the specified context of use. In this thesis, the *context* defines all three aspects, while the *context of use* defines the environment in which the user uses the product. Following the ISO definition, one can only validate the usability within a given context. The developers of a product have to define this context as a part of the requirement specification before evaluating the product. By this, an important first step of the problem analysis was to define the context.

A useful approach to establish the context is personas. Personas are fictional characters that are useful to understand the user and capture the users' goals [79]. In order to create a persona, one needs information about real users. In the case of the murine basal ganglia application, all researchers within

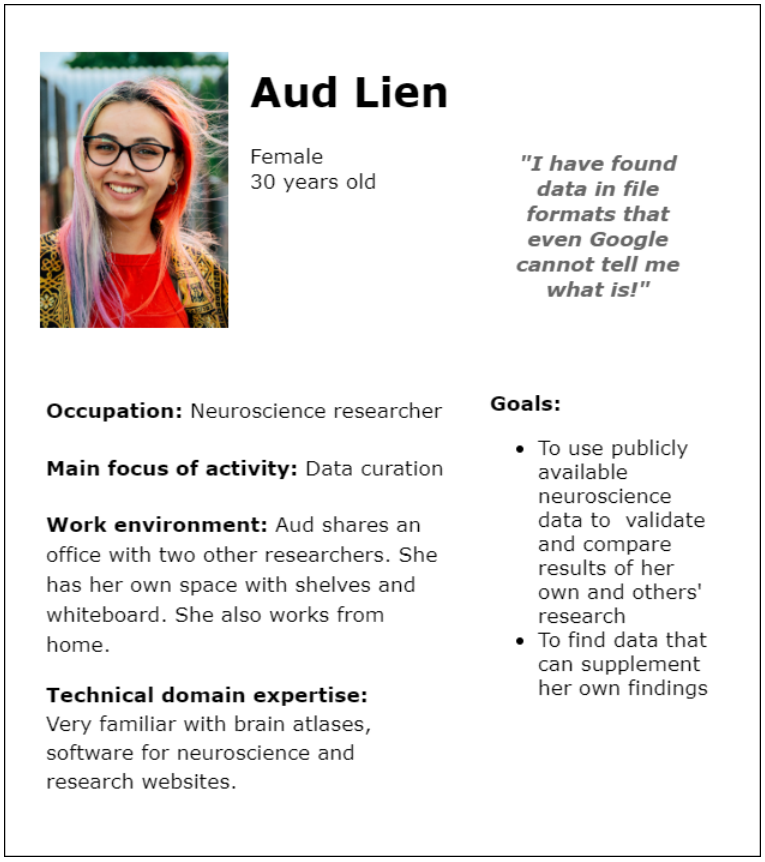

Figure 3.4: User persona.

neuroscience are potential users. However, as the number of participants in the usability study was limited, we narrowed it down to one sub-group of users: data curators. We collected information about these users, including their goals and work environment, from the survey and interviews with Bjerke. Figure 3.4 presents a fictional neuroscientist that describes the murine basal ganglia data set's primary user.

3.6.2 Functional requirements for the user interface

Based on the context defined through the persona, we specified a set of requirements for the user interface. These requirements also apply to the applications as both the web and API applications have to incorporate these

features. To create a focus on the user and their goals, we defined a set of functional requirements for the applications presented as user stories:

US1: As a researcher, I want to find analyses performed on a specific brain region.

US2: As a researcher, I want to find analyses performed on a specific cell type.

US3: As a researcher, I want to find analyses based on species, strain, and other available analysis properties.

US4: As a researcher, I want to be able to find the original publications that exhibit the data.

US5: As a researcher, I want to study detailed information about the methodology used in an analysis.

US6: As a researcher, I want to be able to find the original publications that exhibit the data.

3.6.3 Usability study requirements

The usability test strategy depends on where in the process developers perform the usability test. Barnum (2010) divides usability tests into two sub-categories: Formative and summative testing. Developers perform formative testing when they are still working on the product to find and fix problems. Developers perform summative testing when the product is complete, aiming to validate that they meet the requirements [79]. We defined that the usability study should apply formative testing as it was relevant to obtain user feedback during development.

As stated in the ISO definition, a usability test should observe how users interact with the service within the given context. The persona, shown in Figure 3.4, specified the user and the user's goals. The final part of the context is the user's work environment. As many researchers work from home this year, a remote usability test could represent the researchers' natural work environment. We defined that the test must validate the user interface's

usability within the context of a neuroscientist, performing tasks related to finding and understanding brain-related research data, using a remote-communication tool familiar to the user.

The ISO definition defines three metrics of usability, namely *effectiveness*, *efficiency*, and *satisfaction*. According to Barnum (2010), effectiveness and efficiency is the part that adds value to the user by assisting the user's needs in a better way compared to the current way [79]. Satisfaction regards the user-perceived satisfaction stating if the user will desire to use the product. Thus, the final requirement was that the thesis usability study measured these metrics.

Summary of usability study requirements

1. This usability study should apply formative testing as it is relevant to obtain user feedback during development.
2. The study must validate the user interface's usability as perceived by neuroscientists.
3. The study must provide tasks related to finding and understanding brain-related research data.
4. This usability study should perform usability tests using a remote-communication tool that is familiar to the user.
5. The study must measure the users' *effectiveness*, *efficiency*, and *satisfaction* when using the product.

Chapter 4

Solution design and implementation

Based on the requirements presented in the previous chapter, this chapter describes the design and implementation of the thesis artifacts. The first section describes an overview of the high-level solution architecture, referencing the section that describe each element of the solution. Section 4.2 presents the solution design which entails the final design of an artifact and the decisions leading to it, while Section 4.3 describes the implementation details, putting the decisions from the solution design section into effect.

4.1 High-level solution architecture

Figure 4.1 presents the refined scope of the thesis, revised from Figure 1.1 presented in the introduction chapter. This figure depicts a collective overview of the artifacts designed and implemented in the thesis research. We designed and implemented a graph model for the murine basal ganglia data set, chose a GDBMS, and migrated the data from the relational database into the graph database. Further, we designed and implemented the integration of data from related neuroscience data sources and the technical solution for graph-based data analysis. To provide web-based access to the graph data, we designed and implemented a web application and an API application. The following list presents the main components in Figure 4.1, referencing the sections that describe each component:

1. **The common graph model:** In the middle of the figure, we have the common graph model. Sections 4.2.1 and 4.2.2 present the solution design of the model and the data onboarding, based on the requirements listed in Section 3.3, while Section 4.3.1 presents the data migration implementation.

2. **Integration of external data:** In Section 3.4, we analyzed and found data from three initiatives that integrates the graph model. Sec-

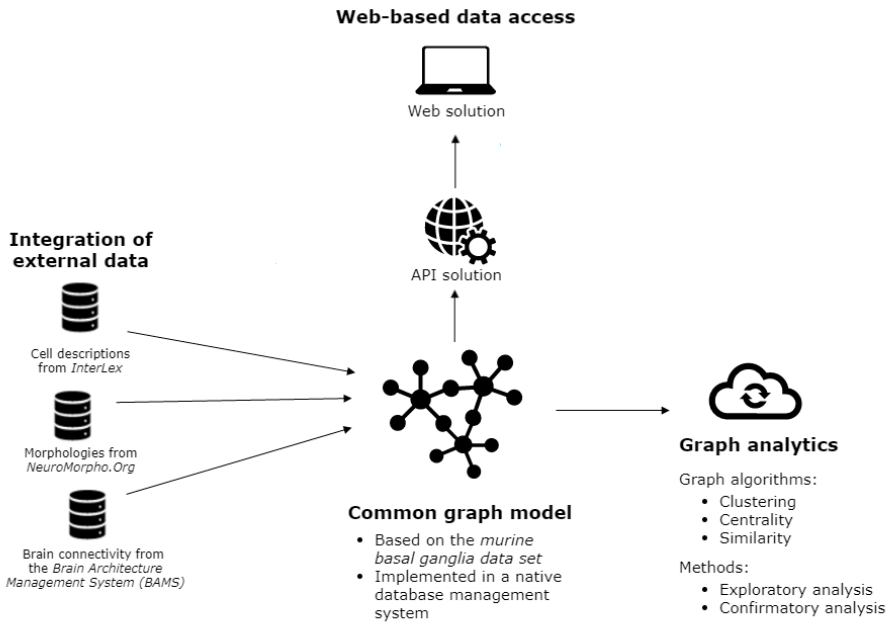

Figure 4.1: High-level architecture of the proposed solution.

tions 4.2.3 and 4.3.2 respectively present the design and implementation of the data integration.

3. **Graph analytics:** On the left side, the figure displays the graph analysis part of the thesis. Section 4.2.4 presents the design of and tools used to analyze the data, based on the approaches listed in Section 3.5, while Section 4.3.3 presents the algorithm set-up.

4. **Web-based data access:** To improve the data usability and accessibility, we developed web-based access to the graph model data based on the requirements presented in Section 3.6.2. Section 4.2.5 presents the technological and user interface design of these applications, and Section 4.3.4 details the implementation.

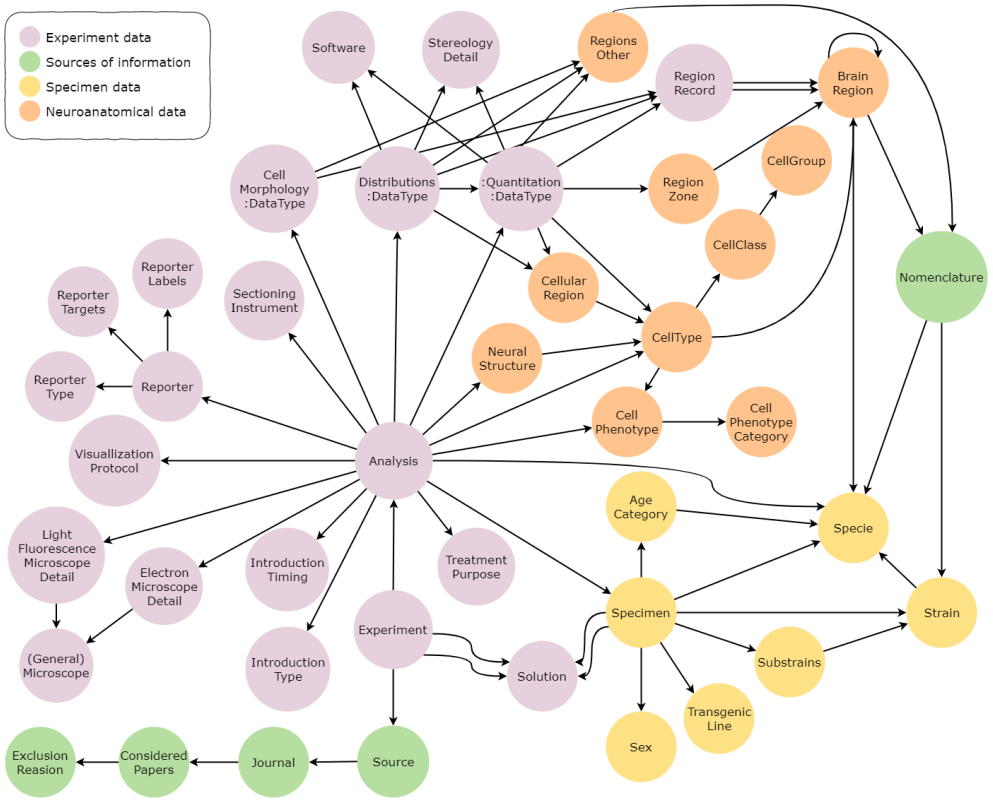

Figure 4.2: The high-level design of the graph model of the murine basal ganglia data set.

4.2 Solution design

4.2.1 Graph-based data modeling

Based on the requirements stated in Section 3.3, we designed a graph model for the thesis data set. The requirements specify that the model should represent the domain model, group together the nodes within the same category, and connect the data to the cell type, brain region, and analysis nodes. Figure 4.2 presents the high-level design of the graph model based on the murine basal ganglia data set.

We based the graph model on the original relational database, conceptually presented in the problem analysis chapter, Section 3.2, and performed changes on this model to satisfy the model requirements. To get an overview of the

domain model and the data usage, we involved researchers from the faculty of medicine at the University of Oslo. The decisions we made for the new structure are outcomes from such discussion. The choice not to keep the original structure came from the benefits the data model could obtain in a new structure, presented in Section 3.3.

Table 4.1 presents the significant design decisions when modeling the original relational structure to the graph model, together with the change's decision basis. For the remaining tables, we followed a general approach[1] for converting a relational database model into a graph model:

- A table becomes a node label.
- Each row in the table becomes a node of that label.
- Each column of the row becomes a property of the node.
- Foreign keys become relationships.
- Join-tables become relationships with properties.

Of the graph type and properties presented in Section 2.1.1, the resulting graph model is a directed multigraph because the relationships have direction and some node pairs have multiple relationships. It also contains a self-loop on the brain region node type. Further, the graph is connected, as there is a path between all the nodes in the graph. We decided to model the graph after what was appropriate for the domain model and requirements and chose to adjust analysis methods instead to fit with this.

Model change	**Decision basis**	**Details**
The "Derived data records" table maps to the "Analysis" label.	To make the node label more descriptive.	Bjerke, who decided the original name, stated that the term "analysis" more precisely describes the data.
Direct relationship between the experiments and analyses.	To clarify the relationship between an experiment and an analysis.	One derives the new relationship through the specimen node.

[1] `https://neo4j.com/developer/relational-to-graph-modeling`

New relationship between the brain region nodes and cell types, presenting the cell types observed in the region.	To provide direct information about cell types in brain regions.	Derived through analyses: The primary brain region of an analysis (through a region record and data type), connects to the analysis' cell type.
New relationships between the specimen nodes and specimen-related nodes (age, strain, sub-strain, sex, and transgenic line).	To connect the same category nodes, specifically the specimen related nodes.	Derived through the experiment table.
New relationship between the region zone nodes and brain region nodes.	To connect the same category nodes and obtain information about zones in brain regions.	Derived through the data type tables.
New relationship between method information and the analysis node.	To easily obtain the method information from an analysis.	Derived through the experiment and specimen table.
Replaced the sectioning detail table with a relationship with properties between the analysis and sectioning details nodes.	To simplify the graph model.	Tables joining two other tables can be replaced with a relationship in a graph model.
Replaced the reporter incubation table, with a relationship between analysis and reporter nodes.	To simplify the graph model.	Tables joining two other tables can be replaced with a relationship.
Three new node labels: IntroductionType, IntroductionTime, and TreatmentPurpose that originally were properties on the specimen table.	To connect analyses directly with method information.	Having these methods in new nodes makes it easy to retrieve data based on the types and possible for graph analytics to evaluate the specific nodes.

Table 4.1: The main design decisions when converting the relational database to the graph data model.

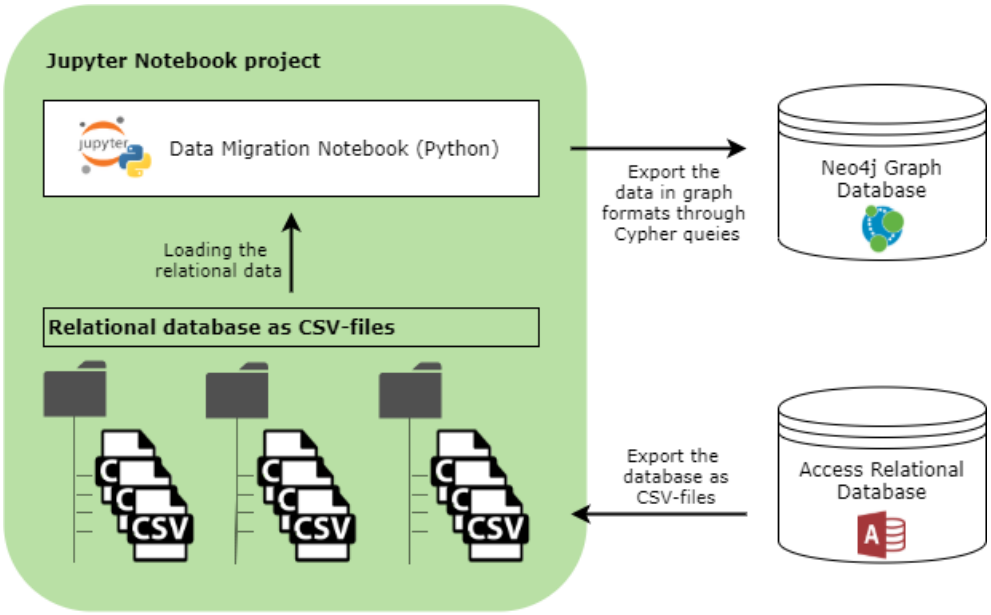

Figure 4.3: Solution design of data migration from the relational database to the Neo4j graph database.

4.2.2 Data onboarding

After designing the graph model, the next step was to specify the data-onboarding from the relational database to a graph database. As this is a research project, we desired the work performed in this thesis to be replicable, which affected the data migration design. Further, we wished to migrate the data on multiple occasions during the design of the graph model. Figure 4.3 presents the design of the data migration that answers these concerns. The following paragraphs present the decisions behind the tool selections and methods used in the presented design.

The first step for moving the data into the new structure was to choose a GDBMS. Section 3.3 presented a set of requirements for the selected system. The principal requirement of a GDBMS is that it fulfills the data and usage requirements. We chose Neo4j as it implements native graph storage, provides integration with many programming languages, has fast create-retrieve-update-delete procedures, is popular and well documented, and can apply graph algorithms on the data.

After choosing a graph database system, we needed to decide how to load

the data into the chosen graph database. Extract, Load, Transform (ELT) tools help extract data from a source, transform it to fit the destination database's schema, and load it into the destination database [80]. Neo4j provides such as tool, Neo4j ELT[2], that automatically maps a relational database to a graph database. However, our graph model did not directly map from the relational model, and as previously stated, we desired that the work performed in this thesis should be possible to replicate. As a result, we decided to implement a data migration solution rather than directly mapping the data to perform the migration on multiple occasions and promote reuse by others.

To make the data migration reproducible and documented, we chose to create a Jupyter Notebook project containing the relational database as CSV-files that provides a Notebook to migrate the data into the graph model format in a Neo4j database instance. The Jupyter Notebook[3] is a popular, free, open-source, interactive web tool. With this tool, researchers can combine software code with additional information and descriptive text in one document [81]. We decided to develop the data migration process using a Jupyter Notebook as it provides a good overview of the code and promotes code narration. We chose Python as the programming language in this solution because it has broad support for scientific computing. There is also a library for running Neo4j in Python, called neo4j[4]. This library supports connecting to a Neo4j database so that developers can run Cypher queries towards the data. Finally, we decided to make the data onboarding solution publicly available on GitHub[5], documented so that any researcher interested could download the code and migrate the data into a Neo4j graph database instance.

4.2.3 Data integration

The next component of the high-level solution design entails extending the standard graph model with external sources. From the initiative analysis presented in Section 3.4, we found three initiatives with overlapping data: BAMS, NeuroMorpho.Org, and InterLex. Figure 4.4 presents how data from these sources connect to the murine basal ganglia data. The following

[2] `https://neo4j.com/developer/neo4j-etl`
[3] `https://jupyter.org`
[4] `https://github.com/neo4j/neo4j-python-driver`
[5] `https://github.com`

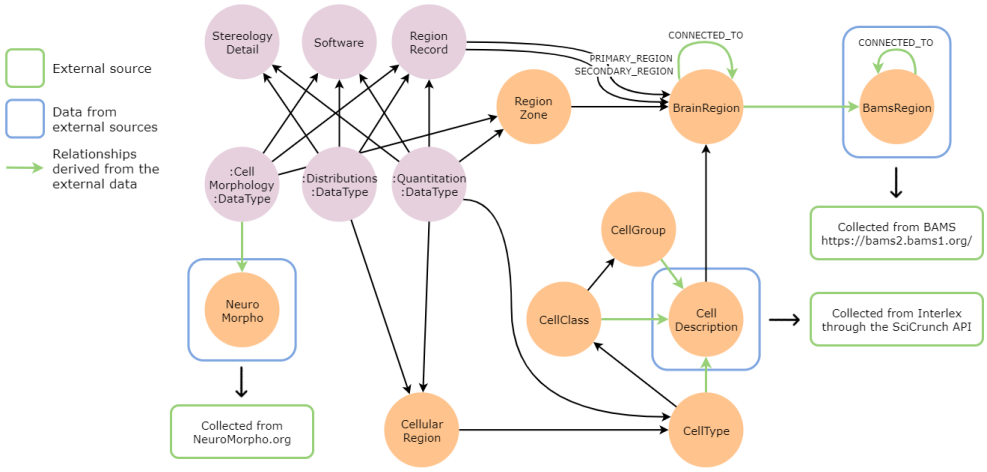

Figure 4.4: Integration of data from external sources.

paragraphs detail what the figure presents.

From BAMS, we found brain region connectivity information possible to integrate with the murine basal ganglia data. We concluded that the retrieved BAMS connectivity information should be stored in new nodes to clarify the data's origin. We decided to store the BAMS brain regions in nodes with a designated node label and to present the connectivity information through relationships between the BAMS region and basal ganglia data set's regions. To connect these regions, we also needed to perform a manual mapping between them. Bjerke mapped the brain regions defined in BAMS with the brain regions defined in the murine basal ganglia data set, and we stored the mapping in the data migration solution. Further, we chose to extract new relationships between the basal ganglia brain regions while maintaining a direct reference to the original data source.

For the integration of cell descriptions from InterLex with the murine basal ganglia data set, we also chose to store the descriptions in new nodes with a designated label. The cell types, cell groups, and cell classes in the thesis data set contain unique identifiers from different neuroanatomical ontologies. InterLex has cell descriptions connected to multiple ontological identifiers. Based on this, we decided to connect cell types, cell groups, and cell classes to cell descriptions based on the cell type's ontological identification attribute.

The final part of the data integration extended the data set with digital cell reconstructions from NeuroMorpho.Org. NeuroMorpho.Org provides

identifiers to the digital reconstructions, and some of the cell morphology nodes in the thesis data set have an attribute for such an identifier. As with the other two initiative's data, we decided to create a new node label to store these constructions and connect the cell morphology nodes with the digital reconstructions by matching the morphology identifiers.

4.2.4 Graph analytics

Following the high-level architecture of the thesis solution, the third component to design was the graph analytics. There are many tools available to analyze graph data. The chosen GDBMS, Neo4j, provides a graph data science (GDS) library[6] that can be added to the database system to analyze and modify data entries. Many Python packages can also perform graph analytics. In this thesis, we chose to use the Python package NetworkX[7]. Another useful tool to analyze graph data is graph visualization. Both Neo4j and NetworkX provide ways to visualize the data, but the visualization has size limitations. Thereby this thesis research utilized Gephi, an open-source software program for exploring and manipulating networks through visualization [82]. Figure 4.5 presents the solution design of graph data analysis in this thesis. The following paragraphs describe each of the mentioned tools and present their relevance in this thesis.

Neo4j Graph Data Science Library[6]: The Neo4j graph data science library provides a wide range of algorithms to run on projected graphs. A graph projection is a subset of the graph and can be created in Neo4j by either specifying node labels and relationship labels or Cypher queries. The general process of running graph algorithms with this tool is to load the desired graph projection, run the algorithms on the projection, and finally output the result and optionally write the values back to the database. As the murine basal ganglia database is in Neo4j, Neo4j's graph data science library is a natural choice of tool to run algorithms on the data set.

NetworkX[7]: NetworkX is a Python package where one can create, manipulate, and study networks. It provides a wide range of algorithms and supports many graph file formats. Multiple Python packages provide implementations of graph algorithms. However, the Python package manager, PyPi[8], provides an API that can tell how many times Python projects have

[6] `https://neo4j.com/docs/graph-data-science/1.3`
[7] `https://networkx.org`
[8] `https://pypi.org`

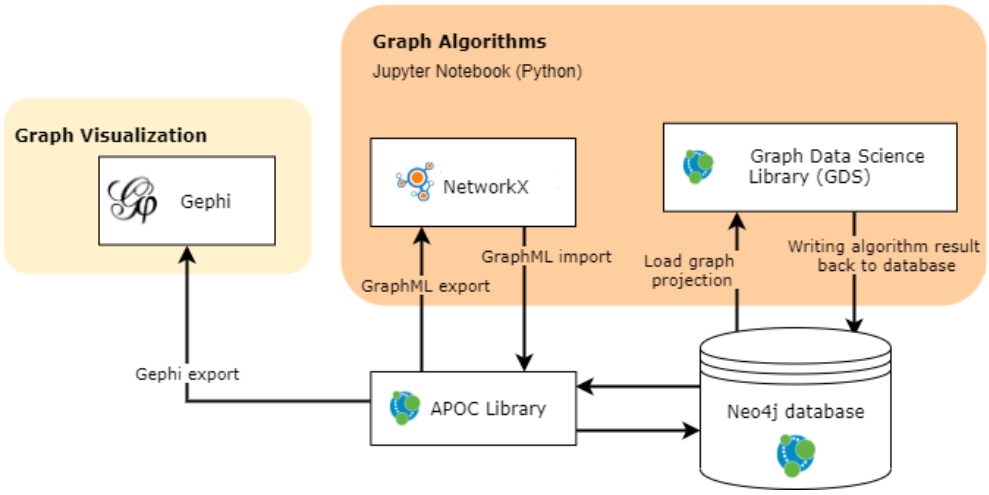

Figure 4.5: Solution design of the graph data analysis.

downloaded a package over the past 365 days. Searching this list for the term "network" returns the package NetworkX as the most popular. By being popular, a large community uses the package, and we can quickly obtain documentation and support. Therefore, we chose to use this package in the research.

Gephi: Gephi is an open-source software program for exploring and manipulating networks [82]. The program provides both advanced visualization and the possibility to manipulate the data directly in the program. In this thesis, the primary use of Gephi was to provide visualizations of the findings provided by the other tools. As presented above, it is possible to visualize data in both NetworkX and Neo4j. However, Gephi is very powerful in handling large amounts of data and provides multiple graph data layout algorithms. We choose to use Gephi for data visualization based on the ease of use, community support, and powerful visualization.

4.2.5 Web-based data access

The final part of the solution design was to specify the web-based data access that intends to improve the data set usability. The developed software includes an API and web application to improve researchers' access to the data programmatically and through a user interface. The next sections present the technological decisions and user interface design.

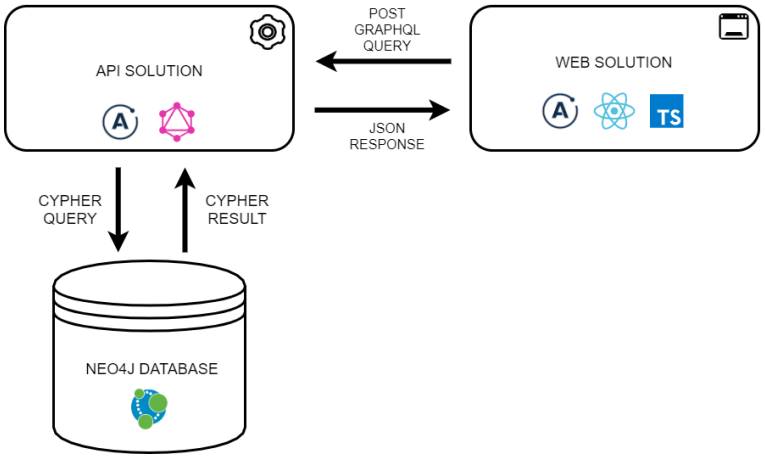

Figure 4.6: Application architecture for web-based access.

Technological design

When choosing web technologies, one should consider the requirements of the developed system and the purpose of the application. There are many API standards and website frameworks to choose from in modern web development, each with their advantages and disadvantages. In this thesis, the web and API application was built based on the GRANDStack[9] architecture, which consists of GraphQL, React, Apollo, and Neo4j. We desired to develop a product that other researchers can understand while using the least possible effort to integrate the different components. Therefore, we chose to use the architecture of GRANDStack as it provides libraries that simplify system integration. Further, each GRANDStack component is matured and has development and community support. Figure 4.6 presents the solution architecture of the API and web application.

GraphQL[10] is a schema-based API query language that fits well with highly interconnected data where the user of the API often needs data of multiple types simultaneously [83]. As the structure allows flexible and customized queries, it is also appropriate when the use cases differ between users or are not clearly defined. We decided to build the API application with GraphQL as it is well suited to represent the data in the murine basal ganglia database for programmatic access, reflecting the structure of the database model well.

[9] `https://grandstack.io`
[10] `https://graphql.org`

React[11] is a popular JavaScript library developed by Facebook for building user interfaces [83]. React is the most popular JavaScript library per August 2020, based on downloads from the JavaScript package manager npm[12]. As React is widely popular and well-known, the library is an appropriate choice when building the web application and user interface.

Apollo[13] works as the connector between the client and GraphQL API applications by offering plenty of libraries assisting effective implementation in a development stack that utilizes GraphQL [83]. Thus, Apollo was the natural choice for these applications.

User interface design

We decided that the web application user interface should be similar to the graph and domain model and desired it to be usable by researchers. From the user interface requirements, presented in Section 5.3.1, we observe that the user stories collectively represent the graph model's requirement that the data should be accessible from three main entry points. Following this, we decided that the application should consist of three top-level pages: one for cell types, one for brain regions, and one for analyses. Figure 4.7 presents a sketch of the user's high-level user interface we designed.

In addition to the pages presented in the figure, we designed a page for each distinct cell type, brain region, and analysis. We chose to represent the data interconnectedness by linking the endpoints. A user can start at any entry point and find data regarding all three areas. A cell type links to brain regions and analyses, a brain region link to cell types and analyses, and a specific analysis links to at least one brain region and cell type. With these choices, the user interface followed the structure of the graph model. The following paragraphs describe the final content and usage of each of these entry points.

The analyses page: This page displays a table of the analyses reported in the data set. On this page, the researcher can filter the results or search for analyses through a search field. The filter includes the data type, which is either a quantitation, distribution, or morphology. When a researcher selects an analysis, it opens a page with information about the selected analysis displayed in tabs. The information on the first tab differs for the three data

[11] `https://reactjs.org`
[12] `https://www.npmjs.com`
[13] `https://www.apollographql.com`

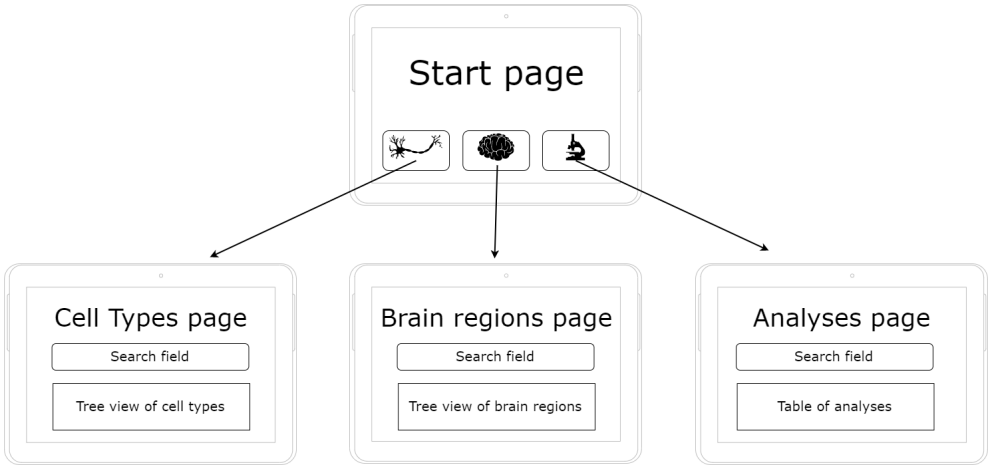

Figure 4.7: A sketch of the basal ganglia web application user interface.

types. For quantitations, it presents the quantification or counting information. For example, how many investigated cell types the researcher observed in the investigated region or regions. For a distribution, the tab presents how the object of interest distributes. Finally, for morphologies, the tab presents an illustration of the cell morphology, collected from NeuroMorpho.org, together with detailed information about the investigated cell morphology.

The remaining tabs on the analysis page are alike for all three data types. There is a tab with animal information that presents specimen information, including weight, age, species, and strain. The next tab displays data acquisition, presenting the research methods used to extract the analysis result, including information about the microscope used, the antibody used, and sectioning details. The anatomical metadata tab contains information about the investigated brain region and region zone and metadata about what the researchers have included in the original publication. Next, the source tab includes a reference to the original publication, including publication year and journal. Finally, there is a tab with similar analyses. The analyses presented in this tab are results from the graph data analysis.

The cell type page: On this page, the user can search for cells, or select them by their cell class or cell group, presented in a tree structure. When the researcher selects a cell type, a pane opens up and presents the cell type with a definition from InterLex, if one exists. In this pane, there are two tabs, one for brain regions and one for analyses. The brain region

pane presents all the regions where experiments have recorded the cell type. Selecting a region directs the researcher to the information page of that region. In the analyses tab, the researcher can see all analyses that investigate the selected cell type. Selecting an analysis from the list navigates the user to the information page of that analysis.

The brain regions page: This page presents all regions related to the basal ganglia for mice and rats in two tree structures. The researcher can search for a region and filter on species. Selecting a region opens up a side pane like with the cell types. There are two tabs for the mouse brain regions and three for the rat brain regions in this pane. The third tab presents connectivity information derived from BAMS. In this tab, the user can see the regions connected to the selected region and filter on direction and strength. Selecting a connection displays the original connectivity information from BAMS with citations and links to where we have collected the connection. The two other tabs are cell types and analyses. The analyses tab displays a list of the analyses that have investigated the selected brain region. The cell type tab presents all the cell types that experiments have recorded in the selected region.

4.3 Implementation

Implementation is the process of putting the decisions made in the previous section into effect. For some artifacts, the previous section presented some technological choices as they were necessary for the design. This section complements the previous by presenting how we implemented each of the presented components, including technological decisions and component integration. Most of the implementation details presented in this section are available through Notebooks in a common Jupyter Notebook project [84].

4.3.1 Data migration

We created a Notebook at the root of the common Jupyter Notebook project for migrating the data from CSV-files into a Neo4j database instance [84]. The following list presents the main steps of the Notebook, and the next paragraphs describe each of these steps with an example.

1. Connect to the Neo4j instance through a driver.

2. Delete all nodes and constraints if any exist.

3. Create all the nodes from the CSV files.

4. Create all the relationships from the CSV files.

5. Create relationships based on the existing relationships.

The first step generates the connection to the Neo4j database instance. Listing 4.1 presents the connection set-up. In this example, the variable `boltUrl` refers to the Bolt URL of the Neo4j database instance, and the variables `user` and `pwd` represent the database login username and password. The variable `driver` is the driver object used to create sessions towards the database instance for running queries on the database.

```
1 from neo4j import GraphDatabase, basic_auth
2
3 driver = GraphDatabase.driver(boltUrl,auth=basic_auth(user, pwd))
```

Listing 4.1: Python code to create a Neo4j Python Driver.

The second step in the Notebook deletes the existing nodes, indices, and constraints on the database. We decided to have this step to simplify database regeneration by making it possible to run the migration on instances with existing data. Listing 4.2 presents the Cypher query to remove all the nodes in the database.

```
1 MATCH (n) DETACH DELETE n
```

Listing 4.2: Cypher query to remove all the nodes in the database.

The third step entails converting the data from the relational database presented in CSV-files into the graph structure. Listing 4.3 presents an example of the code to insert the data from a table into the database using Python. Specifically, it displays moving the data of the `regions.csv` file into `BrainRegion` nodes. In this example, the first line calls a method that returns the full path to the CSV-file passed as an argument. The next line declares a variable with the Cypher query. This query loads the CSV-file and makes the data available in the `row` object. The query then has a `CREATE`-statement, first declaring the node label, and in the square brackets, placing the properties of the node. The `row` object contains the value of the properties. In the seventh line, the code connects to the Neo4j database through the driver object. Line

8 and 9 create an index on the brain regions' name for faster retrievals and a constraint stating that the brain region node's name must be unique. Finally, the code runs the query on the database. We moved all the data from the original tables into the graph database as nodes with multiple similar code blocks.

```
csv_file_path = get_csv_file_path("regions/regions.csv")
query="""
        LOAD CSV WITH HEADERS FROM "%s" AS row
        CREATE (:BrainRegion {id: row.ID, name: row.Region_name, abbreviation: row.
    Abbreviation, comments: row.Comments})
    """ % csv_file_path

with driver.session() as session:
    session.run("CREATE CONSTRAINT ON (n:BrainRegion) ASSERT n.id IS UNIQUE")
    session.run("CREATE INDEX ON :BrainRegion(name)")
    session.run(query)
```

Listing 4.3: Python code to run a query to insert a table from a CSV file into nodes in the Neo4j graph database.

With all the nodes created, the fourth step was to create the relationships. For some of the relations described in the graph model, we needed to create joined tables from the existing database. We did this mainly to the nodes related to the specimen, experiment, or analysis nodes that should connect to another of these nodes. These relationships were extracted by converting the Access 2016 database into an MSSQL database and running join queries. We converted the resulting table to a CSV-file for the graph database query.

Listing 4.4 exemplifies the code for relationship creation from CSV-files. It presents the code to create relationships between `BrainRegion` nodes and `Nomenclature` nodes. In this example, the Cypher query again loads data from a CSV-file and uses it to match the desired nodes. The fifth code line connects the matched node with the relationship `NAMED_BY`. Finally, the code executes the query on the graph database.

```
query="""
        LOAD CSV WITH HEADERS FROM "%s" AS row
        MATCH (brainRegion:BrainRegion { id: row.ID})
        MATCH (nomenclature:Nomenclature { id: row.Nomenclature })
        MERGE (brainRegion)-[:NAMED_BY]->(nomenclature)
    """ % csv_file_path
with driver.session() as session:
    session.run(query)
```

Listing 4.4: Python code to run a query to create relationships between nodes from a CSV-file.

Once the initial relationships and nodes were in place, the final step was to produce relationships derived through the graph. Listing 4.5 presents such a query. This example creates a relationship from the brain regions to cell types informing which cell types researchers have observed in that region. The symmetric relationship would be the brain regions researchers have observed for a specific cell type. However, there is no need to add symmetric relationships, as graph traversal in Neo4j is equally fast in either direction. An analysis is only linked to one cell type, while it can examine multiple regions. An analysis node has a primary and secondary region, and we cannot be sure that analysis researchers observed the cell type in the secondary region. In the example query, we derive the relationship for cell type and brain region using only the `PRIMARY_REGION` relationship between the `BrainRegion` and `RegionRecord` nodes. The last line in the query will then match all regions and cell types connected through the presented path and connect them directly.

```
query = """
        MATCH (region:BrainRegion)<-[:PRIMARY_REGION]-(r:RegionRecord)<-[:REGION_RECORD]-()
     <-[:DATA_TYPE]-(:Analysis)-[:CELL_TYPE_PUTATIVE]->(cell:CellType)
        MERGE (region)-[:CELLS_IN_REGION]->(cell)
    """
with driver.session() as session:
    session.run(query)
```

Listing 4.5: Python code to run a query to create relationships derived from existing nodes and relations.

The Jupyter Notebook project with the code to migrate the data and necessary database CSV-files is available in a GitHub repository under the account of the author of this thesis [84]. With this code, anyone interested can clone the repository, connect to a Neo4j database instance, and insert the basal ganglia data.

4.3.2 Extending the data set

The implementation of data integration followed the same structure as the data migration presented in the previous section. As we needed to regenerate the database on multiple occasions, we chose to store the external data in the common Jupyter Notebook project instead of directly integrating the data. Consequently, the Jupyter Notebook project also contains the data necessary to extend the database with data from the brain-related initiatives BAMS,

InterLex, and NeuroMorpho.org. The following list presents the general steps used to integrate data from these sources into the murine basal ganglia data set. For each of the integrated data sources, the following paragraphs present the specific details of each step.

1. Retrieve the data from the external source.
2. Convert the data to CSV format.
3. Store the data in the Jupyter Notebook project.
4. Migrate the data to the Neo4j database instance.

As presented in Section 4.2.3, we integrated brain region connectivity information from BAMS. The first step, retrieving the BAMS data, was set up by calling the BAMS website, with the specified input and output regions defined in the URL. We stored the output of these calls in temporary HTML-files. In the second step, we converted the HTML data into CSV-files. This method used the Python library BeautifulSoup[14] to retrieve the table-element in the HTML-file and read out the table data. When looping through the table rows, the method stores the data in a 2D list and converts it to a CSV-file. In the third step, we stored the CSV-file with connectivity information from all the basal ganglia-related BAMS regions in the Jupyter Notebook project. Listing 4.6 presents the final integration step, where we inserted the BAMS brain regions, connectivity information, and mapping information into the database, based on the generated CSV-files.

```
1  bams_regions_csv = get_csv_file_path("regions/bams2_regions.csv")
2  # Query to create the BAMS brain region nodes
3  bams_region_query= """
4      LOAD CSV WITH HEADERS FROM "%s" AS row
5      CREATE (:BamsRegion {id: row.id, name: row.name, description: row.desciption})
6  """ % bams_regions_csv
7
8  connectivity_csv = get_csv_file_path("regions/region_connectivity.csv")
9  # Create relationship CONNECTS_TO between BamsRegion and BamsRegion
10 bams_rel_query= """ LOAD CSV WITH HEADERS FROM "%s" AS row
11         MATCH (a:BamsRegion { id: row.bams_id_from})
12         MATCH (c:BamsRegion { id: row.bams_id_to})
13         MERGE (a)-[:CONNECTS_TO]->(c)
14     """ % connectivity_csv
15 # Add properties to the relationship
16 bams_rel_prop_query="""
```

[14] https://pypi.org/project/beautifulsoup4

```
        LOAD CSV WITH HEADERS FROM "%s" AS row
        MATCH (a:BamsRegion {id: row.bams_id_from})-[r:CONNECTS_TO]->(b:BamsRegion { id:
    row.bams_id_to})
        SET r.strength = row.strength
        SET r.technique = row.technique
        SET r.description = row.description
        SET r.reference = row.reference
""" % connectivity_csv

mapping_csv = get_csv_file_path("regions/bams2_mapping_regions.csv")
# Query that maps the BAMS brain regions to the brain regions in the data set
mapping_query = """
        LOAD CSV WITH HEADERS FROM "%s" AS row
        MATCH (a:BamsRegion { id: row.bams_id})
        MATCH (c:BrainRegion { id: row.bg_id})
        MERGE (a)-[:RELATES_TO]->(c)
    """ % mapping_csv

with driver.session() as session:
    session.run(bams_region_query)
    session.run(bams_rel_query)
    session.run(bams_rel_prop_query)
    session.run(mapping_query)
```

Listing 4.6: Python code to integrate the BAMS brain region connectivity data.

The second data integration extended the murine basal ganglia data set with cell descriptions from InterLex, connected to the cell type nodes through ontological identifiers. For the first step, to load the data from InterLex, we wrote a program that calls the SciCruch API and retrieves the description for each cell type with an ontological identifier. Listing 4.7 presents an excerpt from the program. In this program, we used the Python library `urllib`[15] to call the SciCrunch API. We performed the second and third step by storing the cell descriptions with their identifier in a CSV-file that the program saved to the project. We performed the final step, adding the information to the graph database, in a similar manner as for the BAMS data, by loading the CSV-file and adding all the cell descriptions to the database before matching cell type nodes with description nodes by the identifier and connecting them with a relationship.

```
import urllib, json
def getDescription(identifier):
    url = "https://scicrunch.org/api/1/ilx/search/curie/%s" % identifier
    req = urllib.request.Request(url, headers=headers)
    response = urllib.request.urlopen(req)
    data = json.loads(response.read())
```

[15] `https://docs.python.org/3/library/urllib.html`

```
    resData = data["data"]

    definition = resData["definition"]
    return definition
```

Listing 4.7: An extraction of the program that retrieves cell descriptions from InterLex.

The steps to integrate the morphologies from NeuroMorpho were relatively similar to the InterLex data integration. The main difference is that the NeuroMorpho data are images instead of text. We stored the image data as base64 data. Listing 4.8 presents the properties of the integrated morphology nodes. We included all of these properties to cite the morphology in the graph and web application properly.

```
LOAD CSV WITH HEADERS FROM "<path to csv file>" AS row
CREATE (:Neuromorpho { id: row.neuromorphoId, href: row.href, base64: row.base64, archive:
    row.archive, dois: row.original_paper_doi })
```

Listing 4.8: Cypher query to migrate the morphology data from NeuroMorpho to the graph database.

4.3.3 Overview of graph algorithms set-up

As with the other implemented artifacts, the code that runs the graph algorithms is available in a Notebook in the common Jupyter Notebook project, aiming to make the research reproducible and verifiable [84]. Section 4.2.4 presented the solution design of the graph analysis effort, including the analysis tools, specifying that we used Neo4j's GDS Library, the Python package NetworkX and the visualization tool Gephi to analyze the data. This section presents how we technically performed data analysis with the three selected tools.

When using the Neo4j GDS Library to run graph algorithms, it is necessary to create graph projections. Listing 4.9 presents how one can create such a projection. In this example, the projection contains the entire graph. We primarily used this projection containing all the nodes. It is possible to write the algorithm results back to the graph by setting the configuration parameter `writeProperty`.

```
CALL gds.graph.create("whole-graph", "*", "*")
```

Listing 4.9: Procedure call to create a projection of all nodes and relationships.

To integrate the NetworkX data analysis with the Neo4j graph database, we exported the graph database as a GraphML-file, a standard graph format, that NetworkX can import as a graph object. For the NetworkX data analysis, we had to remove some relationships, as the algorithm we used in NetworkX did not support multigraphs. We only included the `PRIMARY_REGION` relationship between the brain region and region record nodes, and the experiment's chemical perfusion fix medium solution and specimen's chemical treatment solution. We deleted the excluded relationships from the graph, returning a pure directed graph model in the graph database.

Listing 4.10 presents the procedure call statement to load the Neo4j database into a GraphML-file. In this call, we have specified the `format` and `useTypes`. The `format` property `"gephi"` makes the label type export properly, and `useTypes` set to `True` provides types that NetworkX desire. To get the results back into the database, we can either store the algorithm's results and insert it for specific nodes or write the result back to the NetworkX graph object and export it to a GraphML-file and load the exported file into the Neo4j database.

```
CALL apoc.export.graphml.all("graphml_digraph.graphml", {format: "gephi", useTypes: True})
```

Listing 4.10: Procedure call to load the Neo4j database into a GraphML file format.

When the graph algorithms returned results of interest, we loaded the relevant data into Gephi to visualize the result. Gephi presents the results visually by using one of the program's many layout algorithms. Neo4j provides a library, APOC[16], that allows the user to stream data directly from the database into Gephi. To allow data to come into Gephi, we configured the server mode to be "Streaming." Listing 4.11 presents the Cypher statement we used to load data from Neo4j to Gephi. The `<gephi-workspace>` is the name of the Gephi workspace, and the `<nodes-and-relationship-query>` represents the Cypher query that matches the nodes and relationships that streams to Gephi.

[16] `https://neo4j.com/developer/neo4j-apoc`

```
MATCH path = <nodes-and-relationship-query> as paths
CALL apoc.gephi.add(null,"<gephi-workspace>", paths) YIELD nodes, relationships, time
RETURN nodes, relationships, time
```

Listing 4.11: Cypher statement to load data from Neo4j to Gephi.

4.3.4 Web-based access implementation

The final artifacts we implemented were applications that provide web-based data access. When implementing applications to have high usability, it is vital to consider the application's users. Multiple software development processes take the user into account. In recent years, many development projects use the Lean Methodology as it has a high user-focus. The Lean Methodology is iterative, meaning that the development happens in cycles consisting of development, deployment, and feedback [85]. As the solutions in this thesis did not have any current users, Bjerke was the user that came with feedback in each development iteration. In the final phases of the development, we considered the feedback from the usability tests. In developing the API and web applications, we followed lean development principles, using feedback and metrics from users to prioritize features while continuously integrating and deploying versions of the applications.

As the GRANDStack architecture inspires the application architecture, we initialized the applications using the GRANDStack starter project on GitHub[17]. We extracted the GraphQL API application and the React web application from the starter project and moved these to a separate repository on GitHub [86, 87]. When we had developed a prototype of the web and API applications, we needed them to be available online to receive feedback from users. To host the applications, we used the Heroku platform[18]. Heroku is a cloud platform that allows users to deploy and host apps and is suitable for prototypes [83]. As the Neo4j Desktop version is not reachable by the hosted API, and as Neo4j does not offer any other free database servers, we set up a Neo4j Sandbox for demonstration purposes and ran the Python code that loads the data from the CSV files into the database. With this, the API and web applications were available for users[19][20].

[17] `https://github.com/grand-stack/grand-stack-starter`
[18] `https://www.heroku.com`
[19] `https://basal-ganglia.herokuapp.com`
[20] `https://basal-ganglia-graphql.herokuapp.com/graphql`

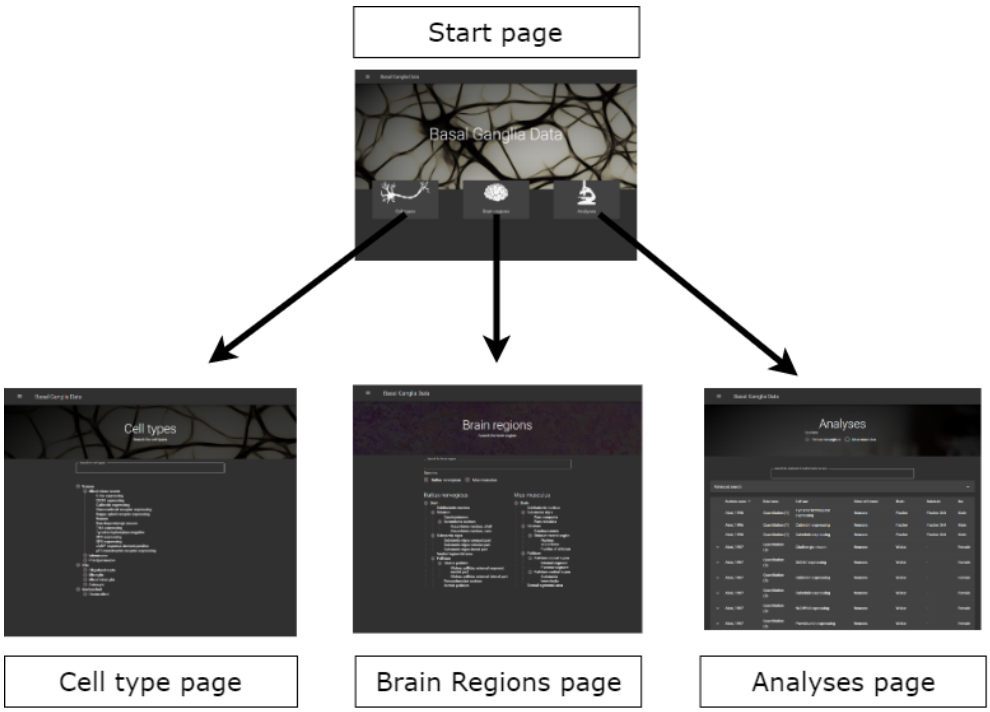

Figure 4.8: The basal ganglia web application top-level user interface.

The key feature of a GraphQL API is its schema. The schema works as a contract between the API and the client and defines what data can be retrieved (queried) and changed (mutated) and the data types. In developing the schema, we followed the Neo4j guide on how to expose a Neo4j database through a GraphQL API. In addition to the schema file, there is an `index.js` file. This file is responsible for serving the application and connecting it to the Neo4j database. In short, the code in the `index.js` file creates an executable GraphQL schema object from the schema file, a Neo4j driver instance to connect to the database, and a new ApolloServer instance, serving the GraphQL schema, and finally, runs the app on a specified path and port. The repository also includes a README-file that describes the primary application usage, including how to run the application locally. The code for the API application is available on GitHub [87].

Figure 4.8 presents the resulting top-level entry points of the web application, and Appendix B presents the complete sitemap of the website. We implemented the web application as a standard React application using the

programming language TypeScript. The application uses React's Context API to maintain state across the application and the Apollo client package to get data from the GraphQL API. The application consists of three primary entry points, as defined in Section 4.2.5. As with the API, the source code of the web application is available on GitHub [86].

4.3.5 Summary of artifacts and used technologies

We summarize here the implemented artifacts and the used technologies of this thesis. Table 4.2 summarizes the produced software artifacts, and Table 4.3 presents the technologies used to develop these artifacts.

The first artifact that we provided is a Jupyter Notebook project that generates the data set, including the data from external sources. This project is available in a GitHub repository licensed under the Creative Commons Attribution 4.0 International license[21] [84]. With these Notebooks, anyone can generate the murine basal ganglia data set in a Neo4j graph database instance. Although not required, we recommend that other researchers use this data set and solution under the same license. Additionally, this solution contains Notebooks to perform data analysis, as described in the evaluation chapter, Section 5.2. The README.md file at the root of the GitHub repository provides documentation of the scripts in the solution.

The two other artifacts produced through the work of this thesis are the web application and the GraphQL API application that provide web-based access to the data set. The web application provides a graphical user interface to the data and can be accessed online. The source code is available through a public GitHub repository [86]. The GraphQL API application, providing programmatic access to the data, is also available online, and the source code is available on GitHub [87]. We have licensed both code repositories under the Apache License 2.0 license[22]. Anyone can download the code, run, and distribute the API and web solutions or provide implementation suggestions directly in these repositories. For each code repository, there is a README.md file documenting the solution set-up.

[21] `https://creativecommons.org/licenses/by/4.0`
[22] `https://www.apache.org/licenses/LICENSE-2.0`

Artifact	Description	Purpose
Jupyter Notebook project	(1) A Notebook for onboarding the murine basal ganglia data set into a Neo4j database instance. (2) Notebooks for data integration and data analysis.	(1) Provide a way for other researchers to obtain the murine basal ganglia graph database. (2) Provides reuse of the research methods for data integration and graph algorithm data analysis.
GraphQL API	Integrates with the murine basal ganglia graph database, using the graph data query language GraphQL and optimization from Apollo.	Provides programmatic access to the murine basal ganglia graph database.
Web application	Developed with the React user interface and integrates with the GraphQL API.	Provides a user interface for the murine basal ganglia data where researchers can interact with the data .

Table 4.2: Summary of implemented software artifacts.

Technology	Type	Purpose
Python	Language	The thesis programming language for data onboarding, data integration, and graph data analysis.
TypeScript	Language	The programming language for the developed web and API applications.
Jupyter Notebook	Application	To create a document for the data onboarding to the Neo4j database instance and to create data integration and graph data analysis documents.
Neo4j GDS library	Library	To run graph algorithms on the Neo4j database.
Neo4j APOC library	Library	To stream sub-graphs from Neo4j to Gephi and to export the Neo4j graph in the GraphML file format.
NetworkX	Python package	The run graph algorithms on the graph data, using an exported GraphML file of the data.
Gephi	Application	Visualize the graph data.
GraphQL	API query language	The developed API query language.
React	Library	Utilized for building the web application user interface.
Apollo	Data Graph Platform	Used to optimize the GraphQL API workflow.
Heroku	Cloud platform	Hosts the API and web application.

Table 4.3: Summary of the chosen technologies.

Chapter 5

Evaluation

This chapter presents an evaluation of the artifacts developed in this thesis based on the requirements from Chapter 3 and further research performed with the artifacts. Section 5.1 presents the evaluation of the graph model and graph database. Next, Section 5.2 describes the results of the graph analysis on the graph model data and the evaluation of the resulting findings. Finally, Section 5.3 presents the user interface evaluation with a review of the functional requirements in Section 5.3.1 and the usability study's set-up and results in Section 5.3.2.

5.1 Evaluation of the graph model and database

In Section 3.3, we defined a set of requirements for the graph model and a set of suggestions for the selected GDBMS. Table 5.1 displays the overall evaluation of the graph model requirements.

The first requirement is a general requirement and difficult to measure directly. However, we marked it as fulfilled because the domain model includes the use of the data set and the user requirements, and by grouping the nodes based on user needs, we have represented the domain model. The usability study presented in Section 5.3.2 further evaluates how the graph model satisfies the user requirements, and indirectly the domain model.

Figure 5.1 presents the original database structure as a conceptual graph, as presented in Section 3.2, and the updated graph model, as presented in Section 4.2.1. In this figure, node labels are omitted for clarity. Evaluating the second research question, we observe that most of the nodes within a category connect. The exception is the green nodes that contain sources of information. As the separation is between nomenclatures and experiment sources, we decided it was acceptable to divide these.

For the third requirement, we increased the connections to the three primary access nodes: cell types, brain regions, and analyses. The figure

#	Requirement	Status
1	The graph model must follow the domain model so a researcher can easily find and compare experiment data.	✓
2	The model should connect nodes within the same category together.	✓
3	Data should be easily reachable from three primary access nodes: cell types, brain regions, and analyses.	✓

Table 5.1: Evaluation of the graph model requirements.

denotes these nodes with text and outlines. Compared to the original database structure, the graph model presents higher connectivity for these nodes. We observe that the node label with the potentially highest node degree is the Analysis node. Further, the cell type and brain region node are node labels with high node degrees. Having a high node-node degree means these nodes connect to many others in the graph directly, and by this are good starting points for accessing data.

We are not evaluating the GDBMS in detail, as that is beyond the scope of this thesis, but we note that the selected system, Neo4j, satisfies the suggestions defined in Section 3.3. Instead, we provide some information about the generated data: The database generated for the research of this thesis consisted of 9539 distinct nodes with 46 distinct node labels, 29807 distinct relationships, and 66 distinct relationship types. Further, we extended the data set with 142 nodes with three labels from integration with external sources. Eighty of these nodes were brain regions from BAMS. The integration added 351 new distinct relationships, where 335 of these represent brain region connectivity.

5.2 Data analysis results

To answer research question RQ_2, we aimed to observe whether graph analytics can derive new information from the data, and by that, observe if a graph model can provide a better understanding of the data. We have separated the data analysis into exploratory data analysis and confirmatory data analysis.

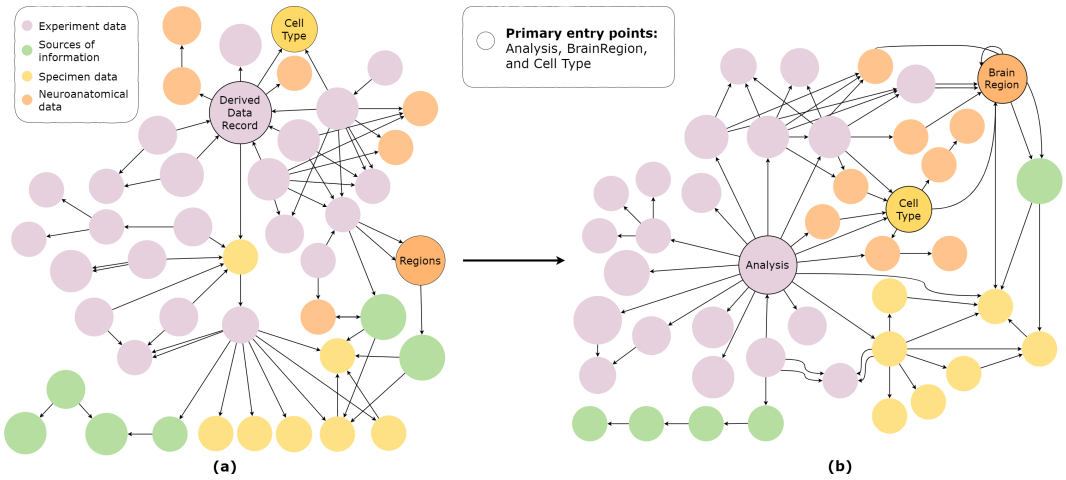

Figure 5.1: Comparison of (a) the original database structure as a conceptual graph and (b) the graph model.

The first was done to obtain general information about the data and the latter to answer specific questions. The Jupyter Notebook project contains the complete algorithm set-up for the graph data analysis [84]. This section presents technical experiment set-up in Section 5.2.1, the results of the exploratory data analysis in Section 5.2.2, and the confirmatory data analysis results in Section 5.2.3.

5.2.1 Exploratory data analysis set-up

As previously presented, exploratory data analysis looks for general information about the data. We used a combination of clustering algorithms and graph visualization to investigate the general data structure. First, we visualized the entire graph to see if we could observe any clustering. To visualize the entire graph, we loaded all the nodes and relationships into the visualization tool Gephi with the query presented in Listing 5.1. Figure 5.2 presents this visualization which uses the ForcedAtlas2 graph layout algorithm [88]. From the entire graph visualization, we observed that the data naturally groups into two almost separate clusters. There is a large cluster on the right side and a smaller cluster on the left side, with some nodes combining them. Investigating the smaller cluster, we identify that it solely consists of data concerning considered papers and their exclusion reason. It is only source

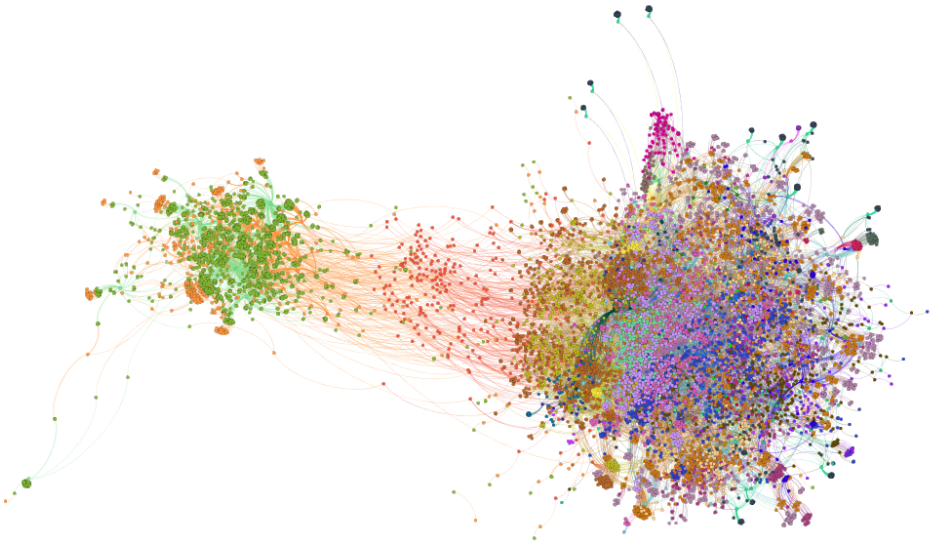

Figure 5.2: The murine basal ganglia data set visualized using the ForcedAtlas2 layout algorithm in Gephi.

nodes that combine the two clusters. Knowing this, we could investigate these clusters separately.

```
1 MATCH path = (n)-[r]-(m) as paths
2 CALL apoc.gephi.add(null,"whole-graph", paths) YIELD nodes, relationships, time
3 RETURN nodes, relationships, time
```

Listing 5.1: Procedure call to load the entire murine basal ganglia graph into Gephi.

Looking at the excluded papers, we only have three types of nodes: the papers, their exclusion reason, and their source. Each paper connects to precisely one exclusion reason and one source. For this reason, the only relevant feature to investigate is the influence of the source and exclusion reasons. After discussing these observations with Bjerke, we decided to exclude these nodes from the analysis. Figure 5.3 visualizes the largest node cluster. The nodes presented here are the basis for the remainder of the graph algorithm experiments.

This thesis utilized community detection algorithms to investigate the graph data structure, specifically the Label propagation algorithm (LPA) and

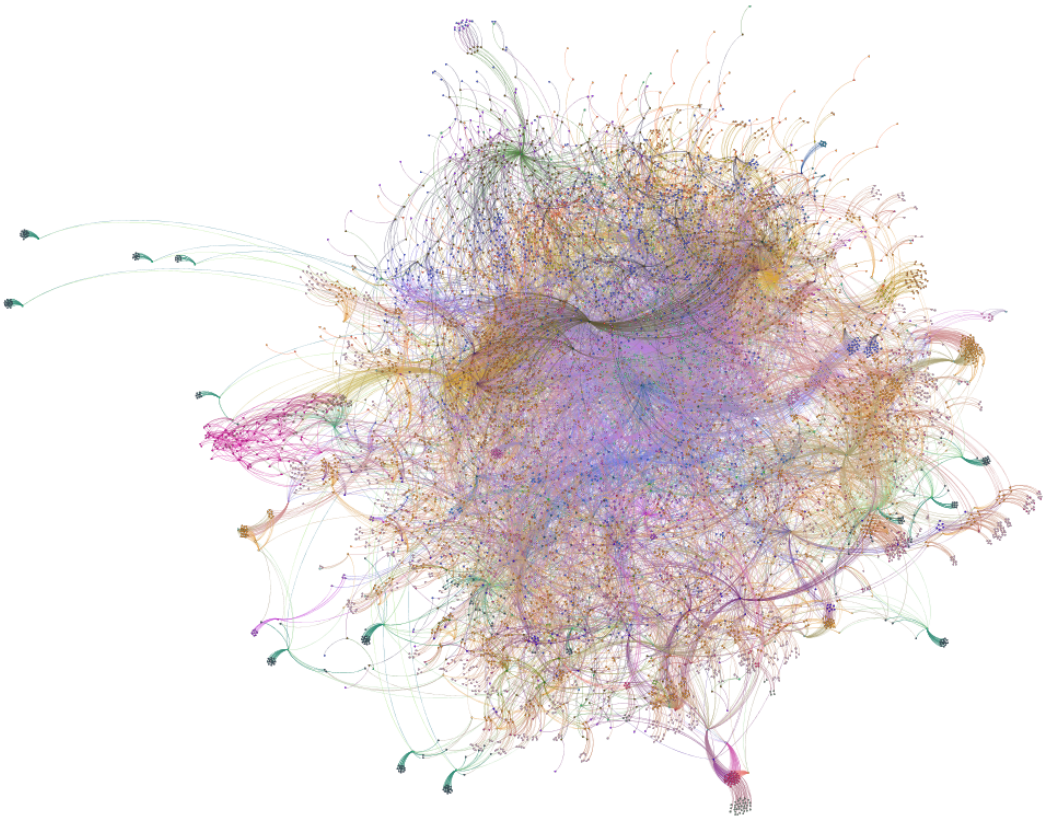

Figure 5.3: The murine basal ganglia data set with the excluded paper data removed, visualized using the ForcedAtlas2 layout algorithm in Gephi.

Louvain algorithm. We choose these algorithms as the LPA performs community detection based on the structure, while the Louvain algorithm applies heuristics based on the nodes' modularity. By applying both algorithms, the results displayed two approaches to community detection. We applied these algorithms directly on the graph database, using the Neo4j GDS library. The nature of community detection algorithms makes it necessary to write the result back to the graph, and the GDS library facilitates this. As we were interested in understanding the structure better, it was relevant to include the entire graph. Listings 5.2 and 5.3 present the procedure calls to run the algorithms on the whole-graph projection. As the LPA is iterative, we had to define the number of iterations. For the murine basal ganglia graph, the algorithm converged after four iterations. To observe the community detection

algorithms' results, we investigated the labels and names and visualized the largest communities.

```
CALL gds.labelPropagation.write( "whole-graph",
    { maxIterations: 5, writeProperty: "community" }
)
```

Listing 5.2: Procedure call to run the LPA in the projection of all nodes.

```
CALL gds.louvain.write("whole-graph",
  { writeProperty: "louvain" }
)
```

Listing 5.3: Procedure call to run the Louvain algorithm in the projection of all nodes.

A suitable method for finding influential nodes in a graph is to run centrality algorithms. There are multiple centrality algorithms. In this thesis, we applied the PageRank algorithm and betweenness centrality algorithm in Neo4j, and the closeness centrality, the betweenness centrality, and the HITS algorithm from NetworkX. We chose these algorithms as they implement differing measures for centrality, including direct and indirect influence. Listings 5.4 and 5.5 presents the algorithm set-up for the Neo4j procedure calls and Listing 5.6 presents the NetworkX code. We ran the algorithms on the whole graph as we were not looking for specific results. The centrality algorithms use information about other nodes, so we did not want to exclude nodes that could affect this.

```
CALL gds.pageRank.stream("whole-graph")
YIELD nodeId, score
RETURN gds.util.asNode(nodeId).name AS name, labels(gds.util.asNode(nodeId)) as label,
    score
ORDER BY score DESC
```

Listing 5.4: Procedure call to run the PageRank algorithm in Neo4j on the whole-graph projection.

```
CALL gds.betweenness.stream("whole-graph")
YIELD nodeId, score
RETURN gds.util.asNode(nodeId).name AS name, labels(gds.util.asNode(nodeId)) as label,
    score
ORDER BY score DESC
```

Listing 5.5: Procedure call to run the betweenness Centrality algorithm in Neo4j on the whole-graph projection.

```
G = nx.read_graphml("graphml_digraph.graphml")
betweenness_centrality = nx.betweenness_centrality(G)
hits = nx.hits(G)
closeness_centrality = nx.closeness_centrality(G)
```

Listing 5.6: Python code that uses NetworkX to run the betweenness and eigenvector centrality algorithms.

The final experiment for retrieving new information about the data was to investigate similar nodes. This thesis used the Node Similarity algorithm, described in Section 2.1.4, provided through the Neo4j GDS library to analyze similarities. We chose to use the Neo4j Node Similarity algorithm mainly due to its efficiency for comparing all of the graph's nodes. In the Neo4j algorithm implementation, one can define parameters that reduce the runtime. In comparison, the similarity algorithms provided by NetworkX do not support a search of the entire graph, only between a pair of nodes. This causes a worse runtime for the NetworkX similarity algorithms than the Neo4j implementation when evaluating the entire graph.

Listing 5.7 presents the procedure call to run the Node Similarity algorithm in Neo4j on the entire graph. Some configuration properties need to be set before running the algorithm. The degree cutoff defines the minimum degree a node can have in order to be considered by the algorithm. This cutoff was set to three, stating that nodes need at least three relationships for consideration. The similarity cutoff defines the minimum similarity requirement. This research sets the similarity cutoff to 0.5, which means that the nodes must have at least 50 percent of the compared nodes in common. It is possible to add a configuration, *topN*, that defines the number of nodes that the algorithm search returns. However, as we were interested in all the results, we did not specify it. The *topK* configuration specifies how many similarity matches each node can have. As the algorithm returned all the similarities, we configured each node to return a maximum of two similar nodes. We investigated the top similarities for each of the central node labels to collect results.

```
CALL gds.nodeSimilarity.stream( "whole-graph", {degreeCutoff: 3, similarityCutoff: 0.5,
    topK: 3})
```

Listing 5.7: Procedure call to run the node similarity algorithm in Neo4j on the "whole-graph" projection.

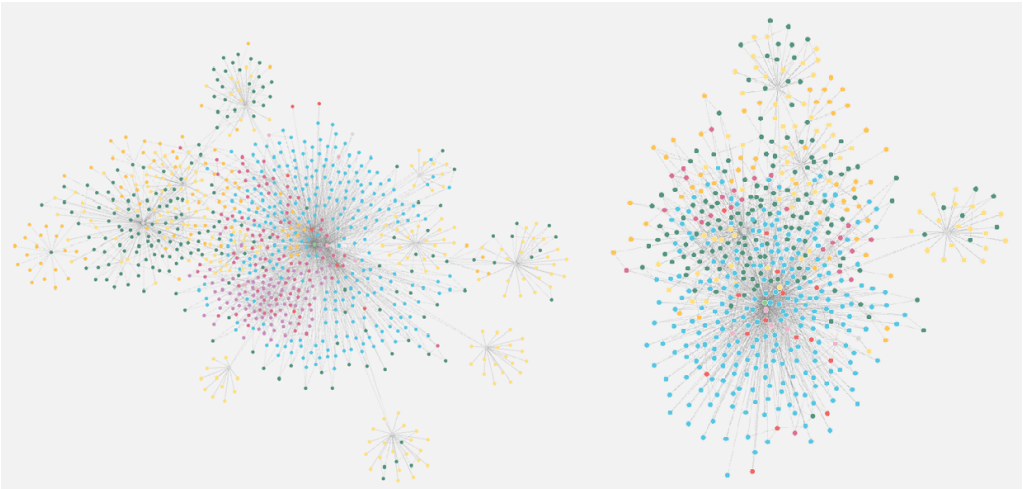

Figure 5.4: Two communities centered around the species Rattus norvegicus (left) and Mus musculus (right).

5.2.2 Evaluation of exploratory data analysis results

This section presents the results of the exploratory data analysis. The results include Bjerke's evaluation of the findings. Table 5.2 summarizes all the findings, including the algorithms used to obtain them and Bjerke's evaluation of the findings.

What can we say about the data structure?

The Neo4j GDS community detection algorithms outputted an overview of the communities in the data set, marked with a community identifier and the number of nodes in each community. Although the nodes and sizes of the clusters were different, both algorithms presented many of the same findings. The largest cluster, found by the LPA, revolved around the cell phenotype category *Expressing*. The second and third clusters gathered around the species Rattus norvegicus and Mus musculus, as visualized in Figure 5.4. The Louvain algorithm also found clusters around the cell phenotype *Expressing* and the species *Rattus norvegicus* and *Mus musculus*. However, these communities were more balanced in size compared to the LPA. The remaining communities clustered around influential cell types, brain regions, and methods. The paragraph presenting the influential nodes describes these.

Can we say something about data quality?

Running the graph algorithms, we found some results that might say something about the data quality. The PageRank algorithm yielded that the second most used anesthetic solution and fourth most used perfusion fix medium solution was *unspecified*. Further, this algorithm yielded that the second most used software was *custom*. Figure 5.5 presents a graph-visualization of the chemical solution nodes connected with the analysis nodes.

When presenting these results to Bjerke, she stated that although she added all the data and knows the quality of method reporting is low, she found the results intriguing. First, she was not aware that the anesthetics are worse than the reporting on perfusion fix mediums. Second, as researchers should report the perfusion fix medium (it affects the antibody penetration and tissue shrinkage), the fact that this is unspecified in many reports is not beneficial. About the reporting of custom software, she stated that it could imply that much research use self or company developed software, which again makes the research results challenging to reuse.

Influential nodes

We found that the most influential journals were Neuroscience, Brain Research, and Journal of Comparative Neurology, who contributed with 32, 20, and 17 experiments, respectively. The most studied brain region was the *caudoputamen*, both for rat and mouse. The study of the murine basal ganglia database by Bjerke et al. also presented these results, making them already known [19].

Further, we found that the rostral zone is the most studied part of the brain regions. Bjerke expected this, as she observed that many articles presented data about this part in the caudoputamen. The rostral region might be preferred in the caudoputamen because it is easier to separate from surrounding regions. However, it is an interesting result as it presents a bias in the research indicating that neuroscience largely bases its knowledge of the caudoputamen on one region zone.

In the evaluation of influential cell types, the algorithms found that the most studied cell object was the neuron as a whole. With a small margin, the most influential cell phenotype category was *expressing*. Further, the results presented that the most investigated cell type was *neurons*, followed by *Tyrosine hydroxylase expressing (TH)* cells. Figure 5.6 presents a graph-

Figure 5.5: Graph-visualization of the chemical solution nodes and analysis nodes in the data set.

visualization of influential cell types.

Bjerke stated that these findings are evident as well, although some are interesting. She is aware that most research investigates the more generic cell type *neurons* and that cell classification based on what they express is widespread. Bjerke expected the finding that much research defines neurons as the object of interest because it is the easiest for scientists to observe. However, she stated it was interesting as it tells us that neuroscience knows much more about the whole cells than the sub-cellular entities.

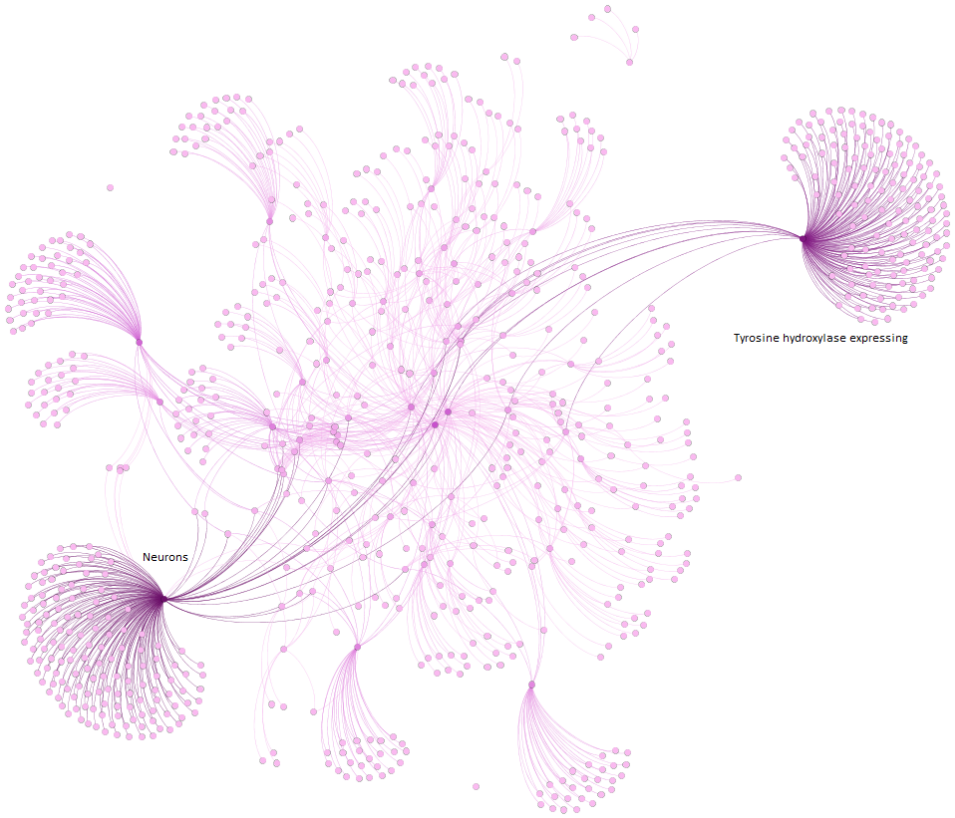

Figure 5.6: Graph-visualization of cell type nodes and analysis nodes from the murine basal ganglia data set.

The TH cells are the first cells to be affected by Parkinson's disease, and much of the basal ganglia research investigate this disease [57]. Thus, Bjerke states that it is expected that this data set contains much TH cell research due to their relevance in Parkinson's disease.

Of the experimental methods, we investigated the most influential visualization method, microscope, sectioning instrument, and reporters. For these categories, the algorithms found the *bright-field microscope* as the most used microscope type and *immunohistochemistry* as the most used visualization method. Further, it found *Cryostat* and *Freezing microtome* as the most used sectioning instruments. Finally, it found *tyrosine hydroxylase* and *Rabbit antibody* as the most used reporter targets and *Goat anti rabbit_ biotin* as the most used reporter.

Evaluating these findings, Bjerke stated that the microscope and visualization methods are common and thereby expected. Bjerke did not expect the sectioning instruments finding but stated that these instruments are mundane. Further, the Rabbit antibody is common in research, and the tyrosine hydroxylase reporter target is influential in this data set due to the amount of research on TH cells. According to Bjerke, the Goat anti rabbit_biotin represents a group of reporters, suggesting something about the data quality; it might be a unique antibody, but we cannot say based on the data reporting.

Finally, we investigated the influence of the nodes related to specimens. From this investigation, we found that in the data set, *Adult* was the most common age category, the most common strain of rat is *Wistar*, and the most common strain of mouse is *C57BL/6*. To Bjerke, these results were already known or highly expected.

We also found that *male* is in this data, by a large amount, the most studied sex. Figure 5.7 presents the nodes representing sexes and analyses and their relationships. From presenting this to Bjerke, we learned that it is common knowledge that most research uses male specimens. A research article evaluating the sex balance in cell and animal studies states that the preference towards males often comes from concerns about varying results due to the estrous cycle; however, this does not display any effect for most applications [89]. Bjerke stated that although this is known for research in general, it was interesting to observe visually and a significant finding as the murine basal ganglia data set was collected unbiased.

Are there any similarities?

We observed that many analyses that investigate low expressing cells also investigated high expressing cells. This similarity is natural as the study probably investigates the specific cell type, and that analysis investigates the different types. Further, for brain regions, we observed in the same manner that the internal segment of the brain region was similar to the external segment of the region. Bjerke stated that this finding is already known as the internal and external segments are sub-regions of the same region, Globus pallidus, and naturally studied in combination. The next section evaluating the use cases presents a more detailed evaluation of similar analyses.

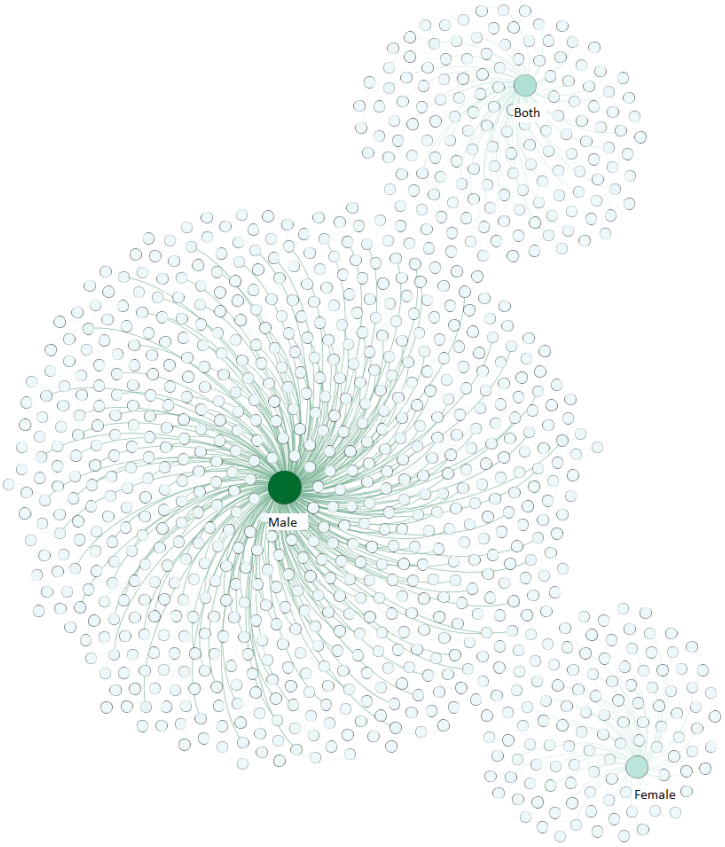

Figure 5.7: Graph-visualization of analyses and the sex they study from the murine basal ganglia data set.

Summary of findings

Table 5.2 summarizes the findings and evaluations from the exploratory graph data analysis. In summary, the information extracted with graph data analysis provided a good understanding of the structure and content of the data set. One should consider Bjerke's evaluation of the findings noting that she is the one who has gathered all of the data. Although she knows the data very well and evaluated many of the findings as expected or already known, she concluded that the graph algorithms combined with graph visualization yielded potent results that provide useful insights and data representations.

Result - information	Algorithms	Evaluation
Cell type information: The most investigated cell type category is "Expressing"	Louvain, LPA, PageRank	This is already known.
Data structure: Community around the data set species.	Louvain, LPA	This is already known.
Data quality: Chemical solution *Unspecified* is the second most used anesthetic and fourth most used perfusion fix medium.	PageRank, Louvain	Expected as it is known that solution information is poorly reported but fascinating to observe for this data.
Data quality: The second most used software is *Unspecified*.	PageRank	Unexpected and interesting as it will make the research results challenging to reuse.
Source information: The most influential publications are *Neuroscience*, *Brain Research*, and *Journal of Comparative Neurology*.	PageRank	Already known as the original database paper by Bjerke et al. also stated this.
Cell type information: *Neuron* is the most investigated cell type.	PageRank, Closeness centrality	Expected as it is the easiest for scientists to observe.
Cell type information: *Tyrosine hydroxylase expressing* (TH) cells are the second most investigated cell type.	PageRank, Closeness centrality	Already known due to TH-cells relevance in Parkinson's disease.
Cell type information: Most of the experiment investigate entire neurons.	PageRank, HITS	Interesting as it implies that neuroscience knows much more about the whole cells than the sub-cellular entities.
Method information: "Bright-field microscope" is the most used microscope type.	PageRank	Expected as it is a very common microscope.

Method information: Immunohistochemistry is the most used visualization method, histochemistry the second most. No difference between species.	PageRank	Expected based on the data Bjerke has collected.
Method information: Tyrosine hydroxylase and Rabbit antibody are the most used reporter targets.	PageRank	Expected as the data contains many TH studies.
Method information: "Goat anti rabbit_biotin" is the most used Reporter.	PageRank, HITS	Unexpected and interesting as it can tell us something about the data quality.
Method information: The most used sectioning instrument is "Cryostat", followed closely by "Freezing microtome".	PageRank	Not evident but of little interest as cutting instruments are mundane.
Brain region information: The data set contains the most information about the brain region caudoputamen for both species.	PageRank, Betweenness centrality, Node similarity	Already known from the data set paper.
Brain region information: Most of the data investigates the rostral region zone.	Betweenness centrality	Expected, but interesting as it displays a bias in the data.
Brain region information: When a study investigates the internal segment region, they also investigate the external segment region.	Node similarity	Already known as they are sub-regions of the same region.
Specimen information: "Adult" is the age category, most often used (big difference).	PageRank	Expected as researchers use other age categories mostly for research specific to the age category.

Specimen information: Most influential strain is Wistar for rats, $C57BL6$ for mice.	PageRank	Expected as these are common strains.
Specimen information: "Male" is the most influential sex.	PageRank	Expected, but interesting as it displays a bias in the research.

Table 5.2: Graph data analysis restuls and evaluation.

Having the results presented, we wished to evaluate which of these presented findings we could have obtained using a relational data model. We have investigated three analysis methods: clustering, centrality, and similarity.

Clustering methods would be challenging to replicate in a relational model as the method heavily bases itself on node connectivity. A relational database cannot fully represent the connectivity in the thesis data set. Even though one could fully represent the data structure properly with the limitations of relational models, the model must be converted to a graph for the clustering algorithms to analyze the data.

Evaluating the centrality measure, we could have obtained some of the graph algorithms' findings with a relational model using relational join and grouping. These are the cases where the nodes connect to only one other node type and when this node type is not highly connected. For example, in the thesis data set, the source nodes only connect to analysis nodes, and discovering which journal has published the most experiments in the data set is trivial. When the node connects to many other node labels, which are highly connected, the graph algorithms present a considerable advantage.

When it comes to the similarity analysis results, we could not easily have obtained these using only a relational database. One would have to compare each row in a table against all other nodes they connect to, which would require nested SQL queries that can become complex and prone to mistakes. Moreover, as with centrality measures, the more interconnected the data becomes, the more difficult it is to analyze in a relational model. In summary, except for some of the centrality measures, we could not have obtained the findings from most of the graph analysis approaches in this thesis using a relational database.

5.2.3 Evaluation of confirmatory data analysis results

In the confirmatory data analysis part, we aimed to answer the two use cases presented in Section 3.5, concerning specific inquiries for the murine basal ganglia data set.

Use case 1: Find similar analysis on specific criteria

The analysis nodes represent one of the data sets' three primary entry points and are what researchers often use when comparing results. For this use case, we were interested in finding analyses investigating the same *cell type* in the same *brain region* and having the same *object of interest*. As with the exploratory data analysis, we utilized the Neo4j implementation of the Node Similarity algorithm. Compared with the exploratory data analysis, we used a slightly adjusted graph projection because we only want the analyses that were entirely similar with respect to cell type, brain region, and object of interest.

We created a graph projection containing only the four relevant labels with a direct relationship between them. The node similarity algorithm ran on this projection with degree-cutoff set to 3 and similarity-cutoff set to 1 and configured to write the relationship back to the graph for the nodes that matched the criteria. These efforts created a relationship between the analyses with the same cell type, brain region, and object of interest. Figure 5.8 presents the analyses (in orange) in the data set connected to the specified nodes and species. The yellow nodes represent the two species in the data set, and the central node in the middle is the cell type "neurons."

We evaluated the result by querying nodes with their similar nodes and verifying that they match the requirements presented above. The result was that the use case was possible to answer and easily achieved with the graph model as it only required a few lines of code to retrieve.

Use case 2: Can the graph model facilitate an evaluation of methods and results in the data set?

In this use case, we aimed to examine how one could compare experimental methods in the rodent basal ganglia data set. There are two parts to this process. The first is to find analyses that research the same topic. Specifically, this entails the same cell type, brain region, and object-of-interest. The second part is to evaluate the methods and results of the analyses that investigate

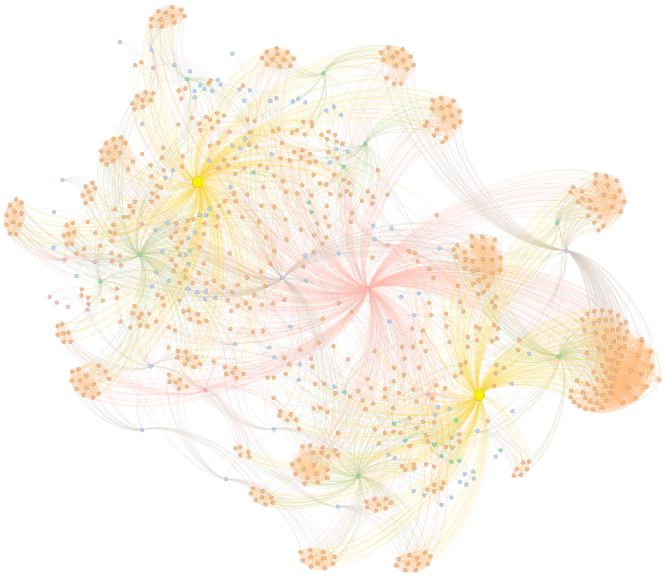

Figure 5.8: The data set analyses with related nodes.

the same topic. The following list presents our graph-based experiment to investigate this use case:

1. Connect analysis nodes directly to the topics they investigate and create a projection with only these nodes.
2. Run a community detection algorithm and store a community identifier on the analysis nodes.
3. Connect analysis nodes directly with relevant methods
4. Create a projection of analysis and connected method nodes for each of these communities.
5. Run a similarity algorithm on each projection.

In the first step, the analysis nodes were connected directly with the cell type, brain region, and object-of-interest nodes, and in the third step, we used the Louvain algorithm to create communities. In the fourth step, we connected the analysis nodes with the method nodes presented in purple in Figure 5.9. We derived some of the presented relationships through the

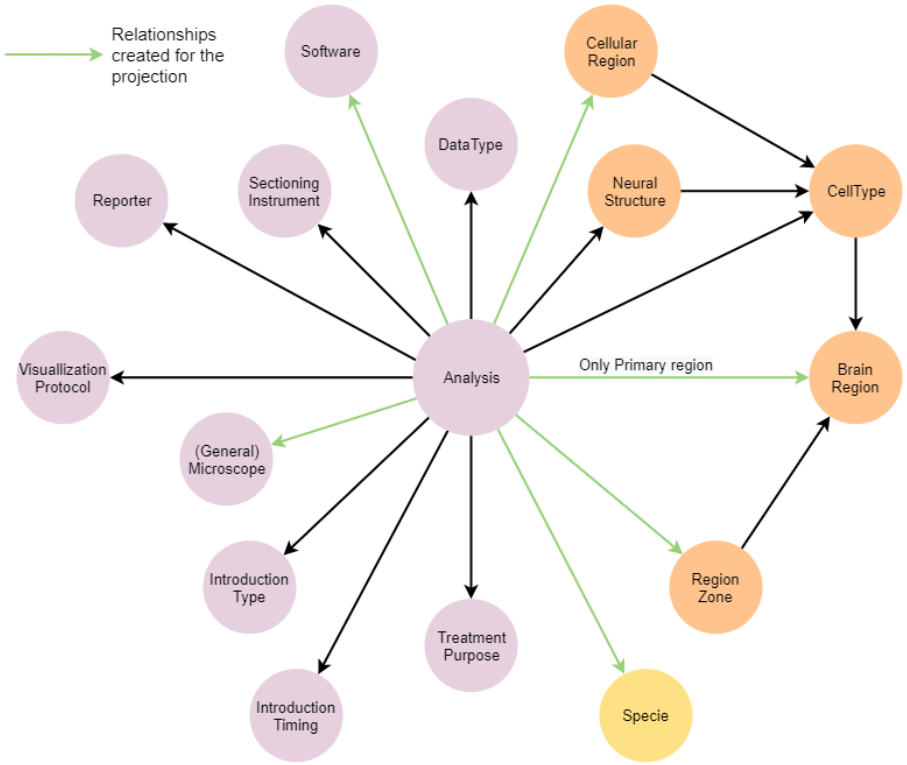

Figure 5.9: The projected graph model used for obtaining method similarity in the second use case experiment.

experiment and data type nodes and removed others that were distinct per analysis. In the fifth step, we set the similarity cutoff to 90 percent, aiming to find all analyses that have used the same methods, except one or two methods, and compare them afterward.

From this investigation, we could conclude that there is not enough data in this data set to compare methodology against the result. The largest community with the same cell type, brain region, and object of interest consisted of 73 nodes, and the second-largest consisted of 26 nodes. When looking for similar nodes having all methods in common except one, we could only find analyses from the same experiment, and those were not relevant for comparison. However, other researchers can use the proposed approach and available code to investigate the same use case.

5.3 User interface evaluation

To evaluate the third research question, RQ_3, we developed a web-based user interface for the graph model data as a basis for evaluating the usability of the graph model. This section presents an evaluation of the web application user interface's functional requirements, verifying that it supports the desired features, before presenting the user interface usability evaluation.

5.3.1 Fulfillment of functional requirements

Section 3.6.2 presented a list of functional requirements, formulated as user stories, for the graph model web application's user interface. The developed web application user interface fulfills these requirements, as presented in Table 5.3. The application implements the first user story through the brain region page, where a user can search and select a brain region and see all analyses that investigate the selected region. It fulfills the second user story in the same manner for cell types. The application completes the third user story through the analysis page, where the user can search and filter all analyses on the listed methods. It realizes the final user stories on the web page of a specific analysis. In the implementation of all the pages mentioned in this paragraph, the user can select a specific analysis and observe multiple analysis properties, including the requirements listed in user stories 4-6.

5.3.2 Usability study

In addition to evaluating the functional requirements, we wanted to evaluate the developed user interface's usability. Section 3.6.3 presented a set of requirements for the usability study. Based on the first requirement, we applied *formative testing*. Although formative testing does not provide any statistics, it gives a good indication of what the users like and dislike and a general impression of the product's usability. The following list presents the process of a formative usability study, as defined by Barnum (2010)[79]:

1. Define the user profile
2. Create task-based scenarios
3. Use the think-aloud process

Functional requirements: User stories	Status
US1: As a researcher, I want to find analyses performed on a specific brain region.	✓
US2: As a researcher, I want to find analyses performed on a specific cell type.	✓
US3: As a researcher, I want to find analyses based on species, strain, and other available analysis properties.	✓
US4: As a researcher, I want to see the anatomical findings and information of an analysis.	✓
US5: As a researcher, I want to study detailed information about the methodology used in an analysis.	✓
US6: As a researcher, I want to be able to find the original publications that exhibit the data.	✓

Table 5.3: Evaluation of the web application's functional requirements.

4. Make changes and test again

For the first step in this process, Section 3.6.1 presents the user and the context: We defined the specified user as a neuroscientist, using a persona, and the context as the neuroscientist's usual workplace. Answering the third requirement, we created tasks related to finding and understanding brain-related research data based on the persona. The set of tasks performed by the usability study participants are presented in its entirety in Appendix D.

In addition to observing how users completed the tasks, we wanted to evaluate the users' overall understanding and experience of data presented in the graph format through web-based access. Naturally, the ease with which they can perform the tasks indicates this. However, we decided to also interview the users after performing the tasks as part of the usability test. An interview is an appropriate method, as we aimed to understand the user's total understanding and experience interacting with the data. The interview questions are formulated not to guide the users into specific answers. The following list presents the general steps of the usability tests performed in this thesis:

1. The observer introduces the thesis and the web application.
2. The observer describes how to think-aloud and encourages the user to apply the technique.

3. The observer presents the tasks to the user and explains that there will be no communication during the tasks, and if the user can not complete a task, they should continue with the next.

4. The user performs the tasks while thinking out loud.

5. The observer interviews the user to evaluate the user's overall experience.

The second usability study requirement presented in Section 3.6.3 states that the participants must be neuroscientists. To find suitable participants, we contacted neuroscience and medicine faculties from multiple universities. However, all the researchers who expressed interest in participating were from the University of Oslo. We verified that the participants were researchers, often working with publicly available data.

The fourth usability study requirement states that the test must use a remote communication tool familiar to the user. For the remote communication tool, we used Zoom as all the researchers at UiO were familiar with it.

Table 5.4 presents the result of the usability study tasks. The order of the usability tests was as presented in the table. Between each usability test, we performed small adjustments to the applications. The users completed the tasks sequentially, in the order presented in the table. The red cells mark the tasks the user did not manage to complete, the yellow cells mark the tasks the users completed but were unsure of, and the green cells mark the tasks the user completed with satisfaction.

As presented in the fifth and final usability study requirement, we aimed to measure whether the applications assist the user's needs (effectiveness and efficiency) and if the users can complete the tasks with self-perceived satisfaction. Thus, we evaluated if the user managed to complete the task and separately evaluated if the user completed the task with satisfaction. The table separates this by marking green cells for tasks the user completed with satisfaction and yellow cells where the user completed the task but did not feel confident.

Evaluating the results, the first two participants struggled with task 5.1 and task 9, while the third participant experienced some struggles with the filter function. Task 5.1 regards finding the total number of axonal varicosities observed by Fujiyama (2016) in the substantia nigra. After observing the two first participants struggling with finding this number, we adjusted the interface to present this number more clearly, and according to the final usability tests, this was successful.

✓	Completed with satisfaction
∼	Completed, but not confident
×	Not completed

Task	User 1	User 2	User 3	User 4
1.1	✓	✓	✓	✓
1.2	✓	✓	✓	✓
2	✓	✓	∼	✓
3	✓	✓	✓	✓
4	✓	✓	✓	✓
5	✓	✓	✓	✓
5.1	∼	∼	✓	✓
5.2	✓	✓	✓	✓
5.3	✓	✓	✓	✓
6	✓	✓	∼	✓
6.1	✓	✓	✓	✓
7	✓	✓	✓	✓
8	✓	✓	✓	✓
8.1	✓	✓	✓	✓
8.2	✓	✓	✓	✓
9	×	∼	✓	✓
10	✓	✓	✓	✓
11	✓	✓	✓	✓
12	✓	✓	✓	✓
12.1	✓	✓	✓	✓
12.2	∼	✓	✓	✓
13	✓	✓	✓	✓
14	✓	✓	✓	✓
14.1	✓	✓	✓	✓
15	✓	✓	✓	✓
15.1	✓	✓	✓	✓
15.2	✓	✓	✓	✓
16	✓	✓	✓	✓

Table 5.4: Task completion in the usability study.

Task 9 asks the user to find the number of mixed-class neurons observed in the mouse caudoputamen. When the user selects a cell type, the resulting page presents all the cell types observed in that region. The goal was that the user should count the number of mixed-class neuron cell types on this page. For the first two participants, this was not clear. The first participant also struggled with task 12, where the user was to find a morphology illustration's source repository. We updated the page to reference the morphology repository more clearly, and the next participants found it with ease.

The third participant struggled with the filter-function at the beginning of the test, which caused the unsatisfactory completion of task 2 and task 6. However, the participant learned how it worked and managed all the subsequent tasks. In summary, the user feedback improved the applications to a point where the users managed to complete almost all tasks confidently. Further, the participants grew more confident throughout the usability test.

In the user interviews, performed right after the tasks, we asked the users about their overall user experience. All the participants had an overall good impression. They felt they understood the application and that the interface provided the necessary entry points for finding data relevant to them. One participant suggested the possibility for community building, such as having a contact page with more information and sharing data. However, as presented in the solution design, sharing data was not possible in the prototype, as we did not have a persistent database. We suggest that the developed applications should implement community functions in the further work chapter.

Chapter 6

Conclusion and further work

6.1 Summary

With technological advances over the past decades, the amount of data generated in the neuroscience domain has increased exponentially. Neuroscience research generates large amounts of brain-related data, and now a challenge is how this research field should deal with all the available data. Simultaneously, in the field of data management, graph databases experience increased popularity due to their ability to handle large data sets that are highly interconnected and dynamic. The research of this thesis investigated how graph-based data representation can improve neuroscience data management.

This thesis presented a graph-based approach for representing neuroscience data, exemplified with the murine basal ganglia data set. We addressed multiple ways of working with a graph model in the neuroscience domain from a data management perspective based on the proposed data set graph model. The thesis described how data from external sources can integrate with neuroscience data in a graph model, applications for web-based access to improve the usability of the data, and the use of graph analytics to extract new information and improve the understanding of the data. Further, the thesis presented evaluations of the developed software, the usability of the data, and the results obtained by applying graph algorithms.

Our goal with this thesis was to evaluate the benefits of graph-based data representation in the neuroscience domain with respect to usability, extendability, and understanding of the data. We presented definite advantages of graph-based data representation through our work, including ease of data analysis, support for data integration, and availability through web-based data access. In light of the thesis scope, we will not conclude that graph-based data representation provides the stated benefits for all neuroscience data. However, we presented a thorough example of how to work with graph-based data representation for neuroscience data. The technical approach from this thesis is practical and generic and can be applied to other data sets. We believe

that the developed artifacts and evaluations contribute valuable insights that promote further research into graph-based data representation in the field of neuroscience.

6.2 Contributions

The thesis research produced software artifacts and a graph model to represent the murine basal ganglia data set in a graph database. With this graph model as a basis, we addressed the following research questions and hypothesis:

RQ_1 *Can a graph representation of brain-related data facilitate the integration of data from a variety of neuroscience data sets?*

Multiple initiatives provide brain-related data, and Chapter 3 presented the related initiatives we analyzed for integration with the murine basal ganglia data set. Chapter 4 presented how we extended the graph model with data from three external data sources; BAMS, InterLex, and NeuroMorpho.Org. We found overlapping data for cell morphologies, cell types, and brain regions that were straightforward to integrate into the existing graph model due to the flexibility of graph database models and standardization of cells. However, the main challenges we experienced with data integration were the lack of data documentation from the initiatives and the lack of data related to the basal ganglia. We had to manually map the brain regions loaded from BAMS to the regions in our data set. By these efforts, we conclude that although a graph model facilitates data integration, other challenges in the neuroscience domain, such as low data availability and lacking documentation and standardization, are more blocking.

RQ_2 *Can a graph model provide a better understanding of the data in a brain-related data set?*

The thesis research contributed with a solution and evaluation for utilizing graph algorithms to improve the understanding of the data in the murine basal ganglia data set. Chapter 3 presented the research

methods and requirements for the data analysis, and Chapter 4 presented the technical aspects of designing and running the algorithms on the graph data model. Chapter 5 presented the specific experiments we used to retrieve new information, the resulting findings, and an evaluation of each finding. The data analysis performed in this thesis research is publicly available through the thesis Jupyter Notebook project. The analysis evaluation concluded that it was possible to extract new information about the data and that some of the information provided an increased understanding. Specifically, we observed that the analyses that evaluate multiple aspects of the node and node constitutions, such as data topology and similarity, are the areas in which graph analyses provide the most noticeable results in the case of the murine basal ganglia.

RQ$_3$ *To what extent can a graph-based approach to neuroscience data management improve the usability of the data?*

There were two primary objectives with this research question. First, we needed a way for researchers to interact with the data set. Second, we wanted to present how computer scientists can integrate the graph data model with applications that provide a user interface and programmatic data access. Chapter 3 presented the requirements of the developed applications and the background for the usability evaluation. Chapter 4 presented how we developed the applications, and Chapter 5 presented how the application user-interface satisfied the presented requirements and the usability evaluation results. We provided web-based access to the graph data through a web application user interface and an API application. These applications are publicly available on GitHub. Further, the usability study exhibited overall high usability of the user interface. The researchers who initially created the data set found the applications usable and wish to continue to use and maintain them. The produced artifacts and usability evaluations suggest that a graph model could increase the usability of the data, although we need further research into this to conclude.

H *Organizing neuroscience data in a graph model provides a better understanding of the data, facilitates data integration with other brain-related data sets, and improves the usability of the data.*

The thesis research exemplified the process of graph-based representation of brain-related data through the murine basal ganglia data set. Further, the research investigated multiple areas of the model guided by the research questions presented above. The research showed that although it does not solve all the challenges with data integration, a graph model facilitates the integration of brain-related data from external sources. In combination with graph analysis, graph-based data representation can provide new information about the data, and our results indicate that this data representation can improve the usability of the data.

Summary of thesis contributions

- A graph model for the murine basal ganglia data set.
- A solution for migrating the data in the murine basal ganglia data set from a relational model to the proposed graph model.
- An analysis of basal ganglia-related data in relevant neuroscience data initiatives.
- An approach for integrating data from multiple neuroscience data sources using the proposed graph model.
- A solution for performing graph data analysis on the murine basal ganglia data set.
- An improved understanding of the data in the murine basal ganglia data set.
- A user interface and programmatic endpoint to access the data, improving the usability of the data.

6.3 Further work

Many of the presented areas of graph-based data representation in the neuroscience domain are still uncharted terrain. It is relevant to continue evaluating the implications of graph-based data representation and work to solve the challenges with data management in the field of neuroscience.

Neuroscience data quality
In extending the murine basal ganglia graph model with data from other neuroscience data initiatives, it was challenging to obtain information about the content the initiatives provided, programmatic data access, and, occasionally, the data format. We suggest that further research performs a thorough review of neuroscience data initiatives and present what data are available from where and how the researchers can access the data, preferably including the data formats. Another approach could be to look at standardization for programmatic access to neuroscience data.

Neuroanatomical data standardization
As many before us have experienced, there are challenges related to integrating neuroanatomical data. There are many ways to name cell types and no standard format. One suggested approach is to create a naming standard for cell types, such as an ontological approach. Standardized naming will make it possible for researchers to find data and studies of relevant cell types across literature and facilitate programmatic and automatic data integration. From our work with cell type data, we believe this is an essential aspect for further research in the data management and neuroscience domain.

For brain regions, there are naming standards, namely the brain region nomenclatures. However, the existing nomenclatures are not compatible, and there is currently no overview that provides a complete mapping of terms between nomenclatures. Bjerke et al. (2019) started the work to create such a mapping between a set of nomenclatures. The data overview they created is a considerable contribution, but it needs to be adequately standardized and extended to all the relevant regions and nomenclatures that exist. An overview of the relations between different nomenclatures would greatly benefit data management in the field of neuroscience, and we recommend that researchers in data management collaborate with neuroscientists to continue this work in the future.

Further research on graph-representation of neuroscience data
This thesis presented an approach for modeling and storing a neuroscience data set in a graph model. We believe that the results of this thesis encourage further research into this area. Further research can investigate a graph-based approach for representing other data types to observe if the benefits and challenges are different for these and on a larges data set in combination with other graph analytics techniques to evaluate the performance and usability. With more research on graph-based data representation in the neuroscience domain, we encourage further research into developing an ontological framework for storing all types of neuroscience data, including the metadata with experimental and specie information. However, to do so, there is a need for further evaluations of multiple data sets to define such an ontology's scope and requirements.

Further development of the thesis artifacts
This thesis research produced a graph database with basal ganglia data, together with a web and API application for web-based data access. Researchers at the Faculty of Medicine at the University of Oslo, who created the relational murine basal ganglia database, see a great benefit of using the developed artifacts and express a desire to continue working with these solutions. By this, there is a need for maintenance and further development of the artifacts. One feature we could not implement, due to the non-persistent database, was the feature for researchers to share their research data. The next step in this context would be to provide a persistent database instance and allow the sharing of more data. This feature would provide a significant improvement as it will promote the solution's relevance for current and future research.

References

[1] Xue Fan and Henry Markram. "A Brief History of Simulation Neuroscience". In: *Frontiers in Neuroinformatics* 13 (2019), p. 32. ISSN: 1662-5196. DOI: 10.3389/fninf.2019.00032.

[2] Suzana Herculano-Houzel. "The human brain in numbers: a linearly scaled-up primate brain". In: *Frontiers in Human Neuroscience* 3 (2009), p. 31. ISSN: 1662-5161. DOI: 10.3389/neuro.09.031.2009.

[3] David A. Drachman. "Do we have brain to spare?" In: *Neurology* 64.12 (2005), pp. 2004–2005. ISSN: 0028-3878. DOI: 10.1212/01.WNL.0000166914.38327.BB.

[4] Danielle S. Bassett, Perry Zurn, and Joshua I. Gold. "On the nature and use of models in network neuroscience". In: *Nature Reviews Neuroscience* 19 (2018), 566–578. ISSN: 1471-0048. DOI: 10.1038/s41583-018-0038-8.

[5] Jeffrey L. Teeters et al. "Data Sharing for Computational Neuroscience". In: *Neuroinformatics* 47 (2008), p. 55. ISSN: 1559-0089. DOI: 10.1007/s12021-008-9009-y.

[6] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph databases*. O'Reilly Media, Inc., 2013. ISBN: 9781449356262.

[7] *Gartner Top 10 Data and Analytics Trends*. https://www.gartner.com/smarterwithgartner/gartner-top-10-data-analytics-trends/. Accessed: 2020-05-22.

[8] Huda Akil, Maryann E. Martone, and David C. Van Essen. "Challenges and Opportunities in Mining Neuroscience Data". In: *Science* 331.6018 (2011), pp. 708–712. ISSN: 0036-8075. DOI: 10.1126/science.1199305.

[9] Sten Grillner et al. "Worldwide initiatives to advance brain research". In: *Nature neuroscience* 19.9 (2016), pp. 1118–1122. DOI: 10.1038/nn.4371.

[10] David Hamilton et al. "An ontological approach to describing neurons and their relationships". In: *Frontiers in Neuroinformatics* 6 (2012), p. 15. ISSN: 1662-5196. DOI: 10.3389/fninf.2012.00015.

[11] Giorgio A. Ascoli. "Mobilizing the base of neuroscience data: the case of neuronal morphologies". In: *Nature Reviews Neuroscience* 4 (2012), pp. 318–324. ISSN: 1471-0048. DOI: 10.1038/nrn1885.

[12] *EBRAINS*. https://ebrains.eu/. Last accessed: 2020-11-04.

[13] *KnowledgeSpace*. https://knowledge-space.org/about. Last accessed: 2020-10-30.

[14] Stephen M. Smith et al. "Functional connectomics from resting-state fMRI". In: *Trends in cognitive sciences* 17.12 (2013), pp. 666–682. DOI: 10.1016/j.tics.2013.09.016.

[15] Katrin Amunts et al. "The human brain project: creating a European research infrastructure to decode the human brain". In: *Neuron* 92.3 (2016), pp. 574–581. DOI: 10.1016/j.neuron.2016.10.046.

[16] Daniel Gardner et al. "The Neuroscience Information Framework: A Data and Knowledge Environment for Neuroscience". In: *Neuroinformatics* 6 (3 2012), pp. 149–160. ISSN: 1559-0089. DOI: 10.1007/s12021-008-9024-z.

[17] Subhashini Sivagnanam et al. "Introducing the Neuroscience Gateway". In: *IWSG* 993 (2013).

[18] Ingvild E. Bjerke et al. "Database of quantitative cellular and subcellular morphological properties from rat and mouse basal ganglia [Data set]". In: *Human Brain Project Neuroinformatics Platform* (2019). DOI: 10.25493/DYXZ-76U.

[19] Ingvild E. Bjerke et al. "Database of literature derived cellular measurements from the murine basal ganglia". In: *Scientific data* 7.1 (2020), pp. 1–14. DOI: 10.1038/s41597-020-0550-3.

[20] Jing Han et al. "Survey on NoSQL database". In: *2011 6th international conference on pervasive computing and applications*. IEEE. 2011, pp. 363–366. DOI: 10.1109/ICPCA.2011.6106531.

[21] Peter J. Denning et al. "Computing as a discipline". In: *Computer* 22.2 (1989), pp. 63–70. DOI: 10.1109/2.19833.

[22] Ida Solheim and Ketil Stølen. *Technology research explained*. Technical report A313. p. 22. SINTEF, 2007.

[23] John Mingers. "Combining IS research methods: towards a pluralist methodology". In: *Information systems research* 12.3 (2001), pp. 240–259. DOI: 10.1287/isre.12.3.240.9709.

[24] Viswanath Venkatesh, Susan A Brown, and Hillol Bala. "Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems". In: *MIS quarterly* (2013), pp. 21–54. ISSN: 02767783.

[25] John W. Creswell and J. David Creswell. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.

[26] R. Burke Johnson and Anthony J. Onwuegbuzie. "Mixed methods research: A research paradigm whose time has come". In: *Educational researcher* 33.7 (2004), pp. 14–26. DOI: 10.3102/0013189X033007014.

[27] Leonhard Euler. "The seven bridges of Königsberg". In: *The world of mathematics* 1 (1956), pp. 573–580.

[28] Albert-László Barabási et al. *Network science*. Cambridge university press, 2016.

[29] Thomas H. Cormen et al. *Introduction to algorithms*. MIT press, 2009.

[30] Reinhard Diestel. *Graph Theory*. eng. 5th ed. 2017. Vol. 173. Graduate Texts in Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg : Imprint: Springer, 2017. ISBN: 9783662536216.

[31] Rick Cattell. "Scalable SQL and NoSQL data stores". In: *Acm Sigmod Record* 39.4 (2011), pp. 12–27. DOI: 10.1145/1978915.1978919.

[32] Robin Hecht and Stefan Jablonski. "NoSQL evaluation: A use case oriented survey". In: *2011 International Conference on Cloud and Service Computing*. IEEE. 2011, pp. 336–341. DOI: 10.1109/CSC.2011.6138544.

[33] Anil Pacaci et al. "Do we need specialized graph databases? Benchmarking real-time social networking applications". In: *Proceedings of the Fifth International Workshop on Graph Data-management Experiences & Systems*. 2017, pp. 1–7. DOI: 10.1145/3078447.3078459.

[34] Diego Fernandes and Jorge Bernardino. "Graph databases comparison: Allegrograph, arangoDB, infinitegraph, Neo4J, and orientDB". In: *DATA 2018 - Proceedings of the 7th International Conference on Data Science, Technology and Applications*. 2018, pp. 373–380. DOI: 10.5220/0006910203730380.

[35] Mark Needham and Amy E. Hodler. *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*. O'Reilly Media, 2019. ISBN: 9781492047681.

[36] Renzo Angles and Claudio Gutierrez. "Survey of graph database models". In: *ACM Computing Surveys (CSUR)* 40.1 (2008), pp. 1–39. DOI: 10.1145/1322432.1322433.

[37] Eric Miller. "An introduction to the resource description framework". In: *Bulletin of the American Society for Information Science and Technology* 25.1 (1998), pp. 15–19. DOI: 10.1002/bult.105.

[38] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. "Semantics and complexity of SPARQL". In: *ACM Transactions on Database Systems* 34.3 (2009). DOI: 10.1145/1567274.1567278.

[39] Jeremy. J. Carroll et al. "Jena: Implementing the semantic web recommendations". In: *Proceedings of the 13th International World Wide Web Conference on Alternate Track, Papers and Posters, WWW Alt. 2004*. 2004, pp. 74–83. DOI: 10.1145/1013367.1013381.

[40] Nadime Francis et al. "Cypher: An evolving query language for property graphs". In: *Proceedings of the 2018 International Conference on Management of Data*. 2018, pp. 1433–1445. DOI: 10.1145/3183713.3190657.

[41] Renzo Angles et al. "Foundations of modern query languages for graph databases". In: *ACM Computing Surveys (CSUR)* 50.5 (2017), pp. 1–40. DOI: 10.1145/3104031.

[42] Mark Newman. *Networks*. 2nd. Oxford university press, 2018.

[43] Franco Scarselli et al. "The graph neural network model". In: *IEEE Transactions on Neural Networks* 20.1 (2008), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

[44] Jie Zhou et al. *Graph Neural Networks: A Review of Methods and Applications*. 2018. arXiv: 1812.08434 [cs.LG].

[45] Lisa Ehrlinger and Wolfram Wöß. "Towards a Definition of Knowledge Graphs." In: *SEMANTiCS* 48 (2016), pp. 1–4.

[46] Sergey Brin and Lawrence Page. "The Anatomy of a Large-Scale Hypertextual Web Search Engine". In: *Computer Networks* 30 (1998), pp. 107–117. URL: http://www-db.stanford.edu/~backrub/google.html.

[47] Eric R. Kandel et al. *Principles of neural science*. Vol. 4. McGraw-hill New York, 2000.

[48] Neil A. Campbell et al. *Biology*. San Francisco, CA: Pearson Benjamin Cummings, 2011.

[49] Larry W. Swanson. "What is the brain?" In: *Trends in neurosciences* 23.11 (2000), pp. 519–527. DOI: 10.1016/s0166-2236(00)01639-8.

[50] Ingvild E. Bjerke et al. "Navigating the murine brain: toward best practices for determining and documenting neuroanatomical locations in experimental studies". In: *Frontiers in neuroanatomy* 12 (2018), p. 82. DOI: 10.3389/fnana.2018.00082.

[51] Merriam-Webster. *Nomenclature*. In: *Merriam-Webster.com dictionary*. URL: https://www.merriam-webster.com/dictionary/nomenclature (visited on 05/24/2020).

[52] Douglas M. Bowden et al. "NeuroNames: an ontology for the BrainInfo portal to neuroscience on the web". In: *Neuroinformatics* 10.1 (2012), pp. 97–114. DOI: 10.1007/s12021-011-9128-8.

[53] Jan G. Bjaalie. "Localization in the brain: new solutions emerging". In: *Nature reviews neuroscience* 3.4 (2002), pp. 322–325. DOI: 10.1038/nrn790.

[54] Mihail Bota and Larry W. Swanson. "Collating and curating neuroanatomical nomenclatures: principles and use of the Brain Architecture Knowledge Management System (BAMS)". In: *Frontiers in neuroinformatics* 4 (2010), p. 3. DOI: 10.3389/fninf.2010.00003.

[55] Gordon M. Shepherd et al. "Neuron Names: A Gene- and Property-Based Name Format, With Special Reference to Cortical Neurons". In: *Frontiers in Neuroanatomy* 13 (25 2019). DOI: 10.3389/fnana.2019.00025.

[56] Petilla Interneuron Nomenclature Group (PING et al. "Petilla terminology: nomenclature of features of GABAergic interneurons of the cerebral cortex". In: *Nature reviews. Neuroscience* 9.7 (2008), p. 557. DOI: 10.1038/nrn2402.

[57] Charles R. Gerfen and J. Paul Bolam. "The neuroanatomical organization of the basal ganglia". In: *Handbook of Behavioral Neuroscience*. Vol. 24. Elsevier, 2016, pp. 3–32. DOI: 10.1016/B978-0-12-802206-1.00001-5.

[58] Frank A. Middleton and Peter L. Strick. "Basal ganglia and cerebellar loops: motor and cognitive circuits". In: *Brain research reviews* 31.2-3 (2000), pp. 236–250.

[59] Jose A. Obeso et al. "The basal ganglia in Parkinson's disease: current concepts and unexplained observations". In: *Annals of Neurology: Official Journal of the American Neurological Association and the Child Neurology Society* 64.S2 (2008), S30–S46. DOI: 10.1002/ana.21481.

[60] Kendra D. Bunner and George V. Rebec. "Corticostriatal dysfunction in Huntington's disease: the basics". In: *Frontiers in human neuroscience* 10 (2016), p. 317. DOI: 10.3389/fnhum.2016.00317.

[61] Ingvild E. Bjerke et al. "Data integration through brain atlasing: Human Brain Project tools and strategies". In: *European psychiatry* 50 (2018), pp. 70–76. DOI: 10.1016/j.eurpsy.2018.02.004.

[62] Miguel-Angel Sicilia, Elena García-Barriocanal, and Salvador Sánchez-Alonso. "Community curation in open dataset repositories: Insights from Zenodo". In: *Procedia Computer Science* 106 (2017), pp. 54–60.

[63] David C. Van Essen et al. "The WU-Minn human connectome project: an overview". In: *Neuroimage* 80 (2013), pp. 62–79. DOI: 10.1016/j.neuroimage.2013.05.041.

[64] Andreas Horn et al. "The structural–functional connectome and the default mode network of the human brain". In: *Neuroimage* 102 (2014), pp. 142–151. DOI: 10.1016/j.neuroimage.2013.09.069.

[65] Joshua L. Morgan and Jeff W. Lichtman. "Why not connectomics?" In: *Nature methods* 10.6 (2013), p. 494. DOI: 10.1038/nmeth.2480.

[66] Eszter A. Papp et al. "Waxholm Space atlas of the Sprague Dawley rat brain". In: *Neuroimage* 97 (2014), pp. 374–386. DOI: 10.1016/j.neuroimage.2014.04.001.

[67] Kirsten K. Osen et al. "Waxholm Space atlas of the Sprague Dawley rat brain delineations v3". In: *Human Brain Project Neuroinformatics Platform* [Data set] (2019). DOI: 10.25493/2R2H-JG8.

[68] Michael J. Hawrylycz et al. "An anatomically comprehensive atlas of the adult human brain transcriptome". In: *Nature* 489.7416 (2012), pp. 391–399. DOI: 10.1038/nature11405.

[69] Ed S. Lein et al. "Genome-wide atlas of gene expression in the adult mouse brain". In: *Nature* 445.7124 (2007), pp. 168–176. DOI: 10.1038/nature05453.

[70] Seung Wook Oh et al. "A mesoscale connectome of the mouse brain". In: *Nature* 508.7495 (2014), pp. 207–214. DOI: 10.1038/nature05453.

[71] Henry Markram. "The Blue Brain Project". In: *Nat Rev Neurosci* 7 (2006), pp. 153–160. DOI: https://doi.org/10.1038/nrn1848.

[72] Csaba Erö et al. "A cell atlas for the mouse brain". In: *Frontiers in neuroinformatics* 12 (2018), p. 84. DOI: 10.3389/fninf.2018.00084.

[73] Mihail Bota, Hong-Wei Dong, and Larry W. Swanson. "Brain architecture management system". In: *Neuroinformatics* 3.1 (2005), pp. 15–47. DOI: 10.1385/NI:3:1:015.

[74] Giorgio A. Ascoli, Duncan E. Donohue, and Maryam Halavi. "NeuroMorpho. Org: a central resource for neuronal morphologies". In: *Journal of Neuroscience* 27.35 (2007), pp. 9247–9251. DOI: 10.1523/JNEUROSCI.2055-07.2007.

[75] Jeffrey S. Grethe et al. "SciCrunch: A cooperative and collaborative data and resource discovery platform for scientific communities". In: *Front. Neuroinform. Conference Abstract: Neuroinformatics*. 2014. DOI: 10.3389/conf.fninf.2014.18.00069.

[76] *FDI Lab - SciCrunch Infrastructure | InterLex | Dashboard*. https://scicrunch.org/scicrunch/interlex/dashboard. (Visited on 08/30/2020).

[77] Frederick Hartwig. *Exploratory data analysis*. Vol. no. 07-016. Quantitative applications in the social sciences ; SAGE, 1979. ISBN: 9781412984232.

[78] *ISO 9241-11:1998(en), Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability*. https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en. (Visited on 06/10/2020).

[79] Carol M. Barnum. *Usability Testing Essentials: Ready, Set... Test*. Elsevier Science & Technology, 2010. ISBN: 012375092X.

[80] M. Tamer Özsu and Patrick Valduriez. *Principles of distributed database systems*. 4th ed. Springer, 2019.

[81] Jeffrey M. Perkel. "Why Jupyter is data scientists' computational notebook of choice". In: *Nature* 563.7732 (2018), pp. 145–147. DOI: 10.1038/d41586-018-07196-1.

[82] Mathieu Bastian, Sebastien Heymann, Mathieu Jacomy, et al. "Gephi: an open source software for exploring and manipulating networks." In: *Icwsm* 8.2009 (2009), pp. 361–362.

[83] Robin Wieruch. *The Road to GraphQL: Your journey to master pragmatic GraphQL in JavaScript with React. js and Node. js*. Robin Wieruch, 2018.

[84] Maren P. Gulnes. *Murine Basal Ganglia Notebooks*. https://github.com/marenpg/jupyter_basal_ganglia. Version 1.0.2. 2020. DOI: 10.5281/zenodo.4141244.

[85] Craig Larman and Bas Vodde. "Scaling lean & agile development". In: *Organization* 230.11 (2009).

[86] Maren P. Gulnes. *Murine Basal Ganglia Web Solution - Client*. https://github.com/marenpg/basal_ganglia_client. Version 1.0.1. 2020. DOI: 10.5281/zenodo.4106445.

[87] Maren P. Gulnes. *Murine Basal Ganglia Web Solution - GraphQL API*. https://github.com/marenpg/basal_ganglia_api. Version 1.0.3. 2020. DOI: 10.5281/zenodo.4106461.

[88] Mathieu Jacomy et al. "ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software". In: *PloS one* 9.6 (2014), e98679. DOI: 10.1371/journal.pone.0098679.

[89] Janine A. Clayton and Francis S. Collins. "Policy: NIH to balance sex in cell and animal studies". In: *Nature News* 509.7500 (2014), p. 282. DOI: 10.1038/509282a.

Appendices

Appendix A

Summary of the murine basal ganglia database

This appendix presents the entity-relationship (ER) diagram of the relational of the murine basal ganglia database before it describes the content of this database in greater detail.Figure A.1 presents the ER diagram produced by the relational database. Figure A.2 presents the same diagram colored in the categories presented in Section 3.2.

Figure A.3, collected from Bjerke et al. (2020), presents a summary of the murine basal ganglia database [19]. In this figure, the hierarchical structure is visible. The figure denotes the data types as either quantitative estimates or cell morphologies. In addition to presenting the structure, Figure A.3 presents some key information about the data in the data set.

Figure A.4 presents the workflow proposed by Bjerke et al. that includes three researcher scenarios use; (1) researchers who want to find basal ganglia data for modeling, (2) researchers who want to update the database, and (3) researchers who want to share their data. For further inquiry about the data, please review the article "Database of literature derived cellular measurements from the murine basal ganglia" by Bjerke et al. [19].

The remainder of this appendix describes the nodes and how they relate to each other, for each of the categories presented in Section 3.2.

Experiment data: The experiment data has a hierarchical structure where an `Experiment` node connects to one or many `DerivedDataRecord` nodes, and a `DerivedDataRecord` node relates to only one `Experiment` node. A `DerivedDataRecord` node connects to one or more of the following data type nodes; `Distribution`, `Quantitation`, or `CellMorphology`, and each of these is only related to one `DerivedDataRecord` node.

Each of these three levels has related information. The `Experiment` nodes relate to information about chemical solutions and the experiment specimens, with all the specimen information directly connected to the `Experiment` nodes, rather than the `Specimen` nodes. The `Experiment` nodes connect to `DerivedDataRecord` nodes through a `Specimen` node. Some experiment-related nodes connect to the Specimen node, such as the specimen treatment, how the cells are labeled, the sectioning details, and the reporters used.

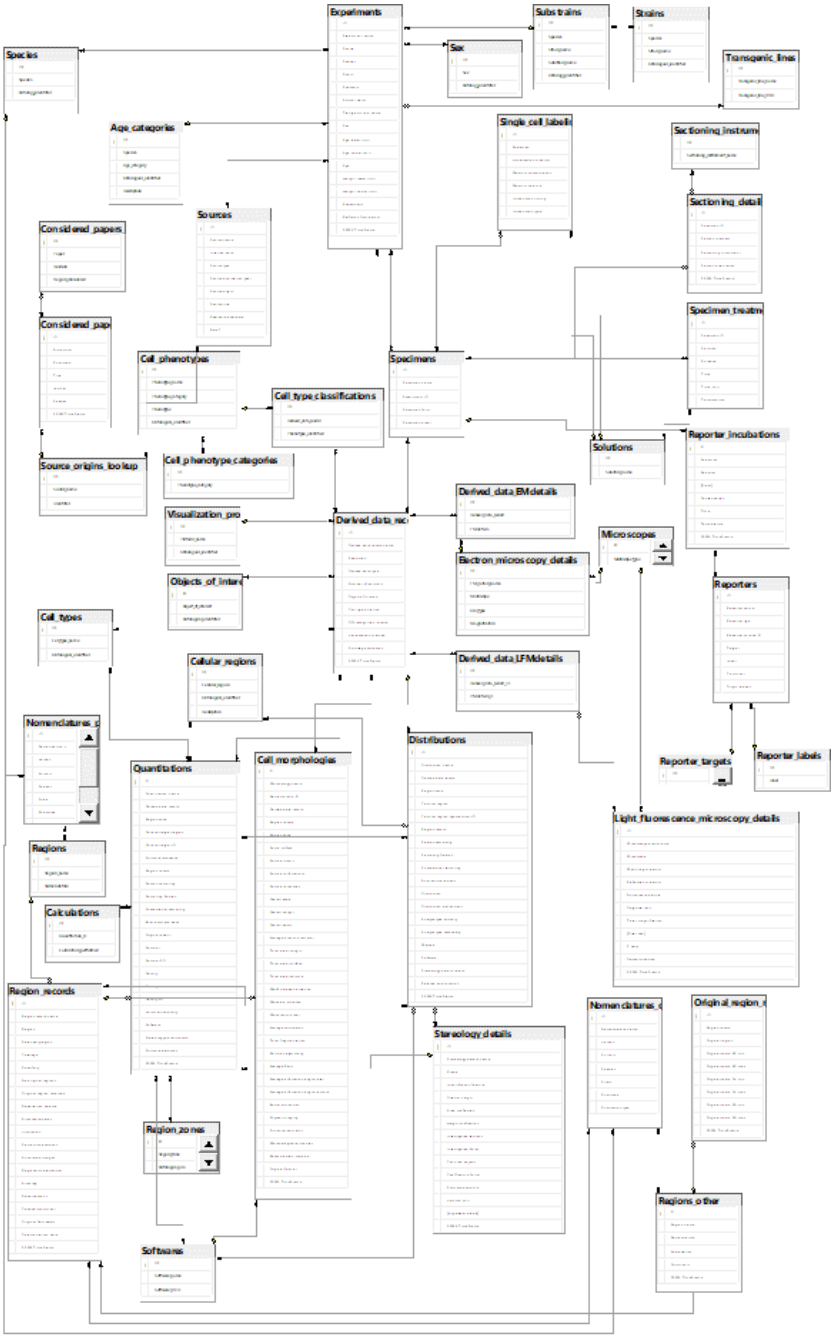

Figure A.1: ER diagram of the murine basal ganglia database.

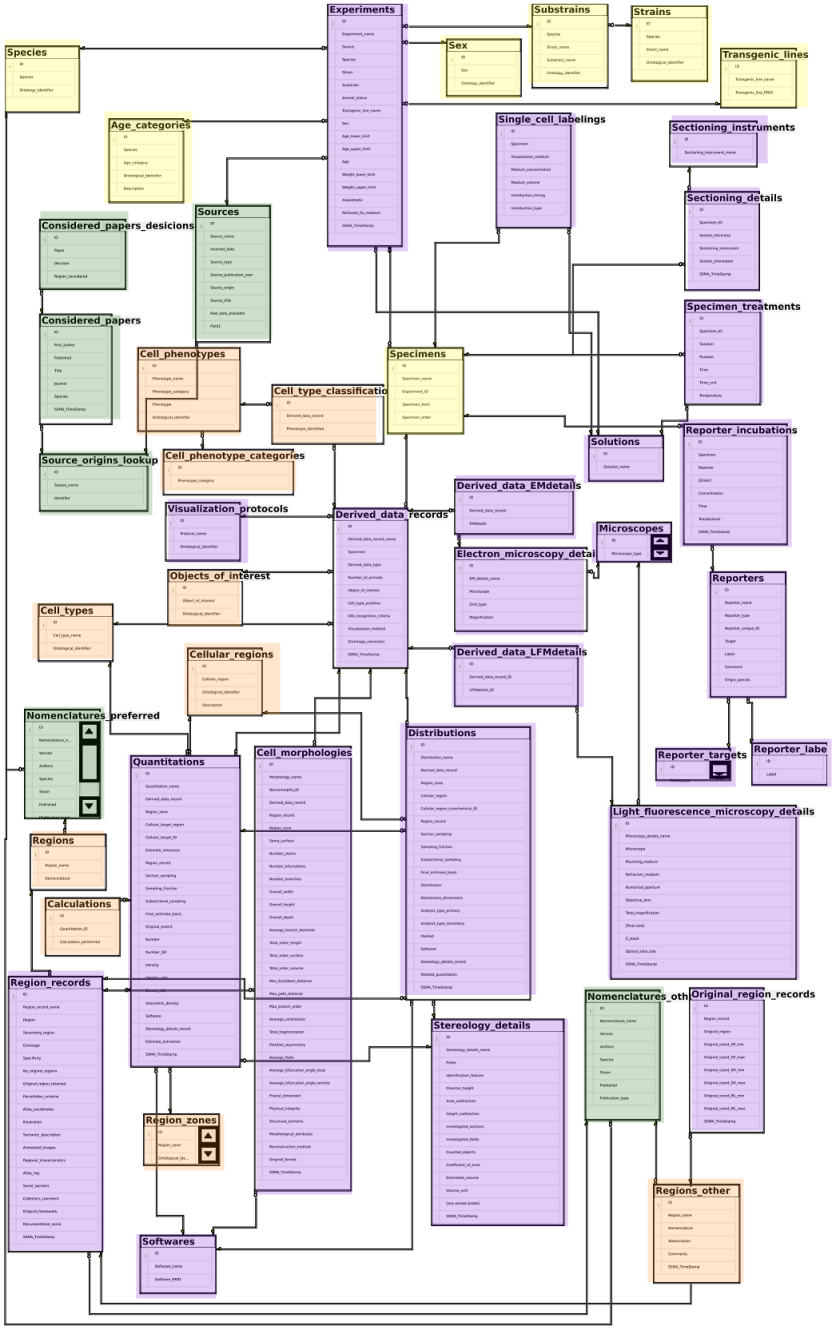

Figure A.2: Categorized ER diagram of the murine basal ganglia database.

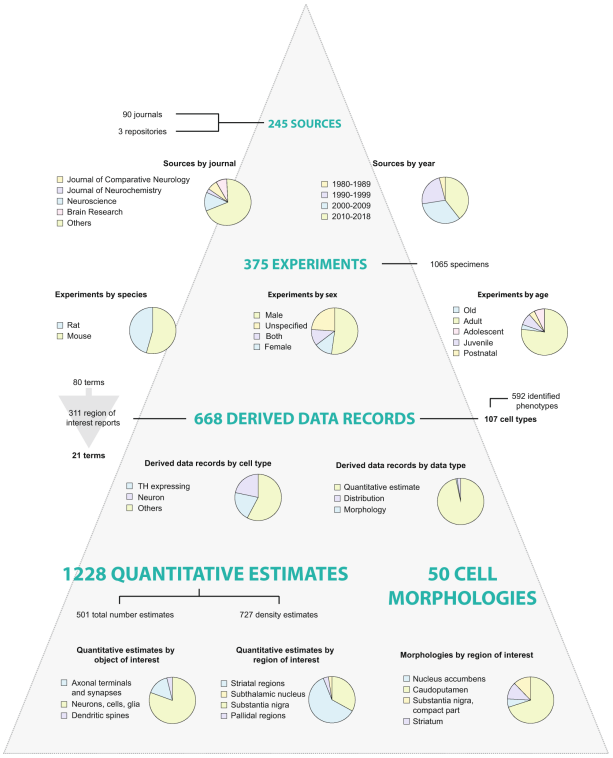

Figure A.3: Summary of the relational murine basal ganglia database.

Credit: Collected with permissions from Bjerke et al. (2020) [19].

Of experiment data, the `DerivedDataRecord` nodes have information about the microscope and visualization protocol used and the relation to the data types. The `DerivedDataRecord` nodes also connect to the cellular information, describing the investigated cell, including the cell type, object of interest, and cell phenotype. The data types listed above can have information about stereology and software used. The different data type nodes connect to the brain region nodes through a data type-specific `RegionRecord` node. They are also related to the nodes defining the zone of the region and the cellular region. Regarding the data types defined in Section 2.2, all the purple nodes are metadata, except the data types (`Distributions`, `Quantitations`, and `CellMorphologies`) that are the derived data type, containing the results.

Sources of information: The source nodes divide into two groups; the nomenclatures and the sources that initially contained the experiments. In

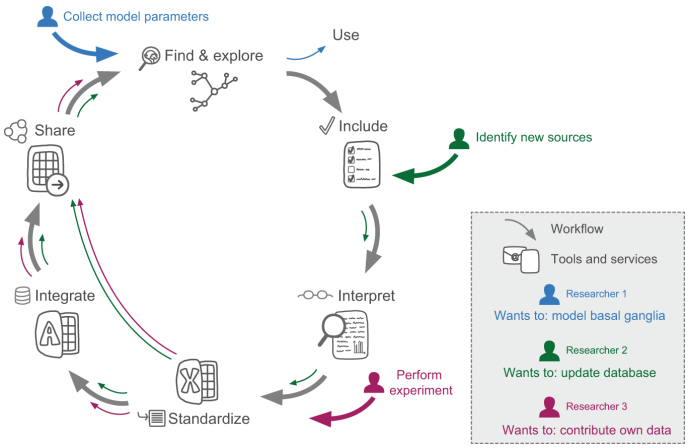

Figure A.4: Workflow for researchers to find, explore, and integrate derived data in the relational murine basal ganglia database.

Credit: Collected with permissions from Bjerke et al. (2020) [19].

the latter group, the `Source` nodes contain a specific publication or repository where Bjerke collected the experiment information. The `Source` node connects to one `SourceOriginLookup` node, containing the information about where the source was published. The `SourceOriginLookup` nodes are usually journals or larger initiatives. A `SourceOriginLookup` node connects to one or many `ConsideredPaper` nodes, containing all the papers Bjerke has considered. These again connect to a `ConsideredPaperDesicion` stating if the paper is included in the data set or not, and if not, it contains the exclusion reason. The second category contains nomenclatures. In the data set, Bjerke has mapped all the data to fit with the data set nomenclatures, referenced in the `NomenclaturesPreferred` nodes. The `NomenclaturesOther` nodes comprise the information about the original nomenclatures used in the source.

Specimen data: All the `Specimen` nodes related directly to the `Experiment` nodes. The `Specimen` nodes represent the experiment subject. Information about this specimen exists in the remaining specimen-related nodes, including the `Specie` nodes, `Strain` and `Substrains` nodes, and the nodes describing sex and age categories.

Neuroanatomical data: The brain region nodes contain the region's name and the nomenclature that provided it. The `Regions` nodes refer to this data set's nomenclatures, and the `RegionsOther` nodes refer to the original

experiment's nomenclature. If an experiment only observes a specific part of the region, it relates to the `RegionZone` nodes that describe parts of brain regions. The same goes for the `CellularRegion` nodes that are zones, or regions, of the cell. The `CellType` nodes are different cell types, and the `CellPhenotype` nodes describe the cell's appearance or physical attributes [48, p. 312]. The `CellPhenotypeCategory` nodes categorize the `CellPhenotype` nodes. Finally, the `ObjectOfInterest` nodes define the object the experiment observed, which can be any neural structure from generic cells to more specific cell types or cell regions.

Appendix B

Sitemap of web interface

Figure B.1 presents the sitemap of the web application developed as part of this thesis.

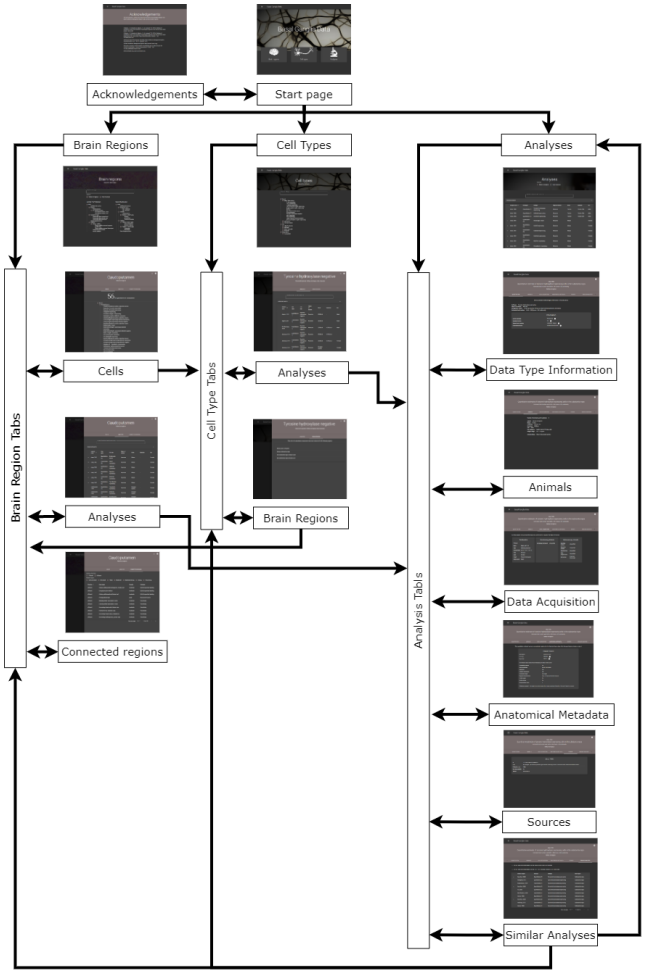

Figure B.1: Sitemap of the basal ganglia web application.

Appendix C

Survey on usage of neuroscience data

This appendix presents the survey presented by this thesis to obtain a deeper understanding of the usage of publicly available neuroscience data. We sent the survey to multiple research institutes, and manged to get fourteen responses. We created the survey using Google Forms. The first section presents the results of the survey questions, while the second section presents the Google form.

C.1 Survey results

Chart B.1 presenting the results of the first survey question asking about the participants' research background, displayed in a pie chart.

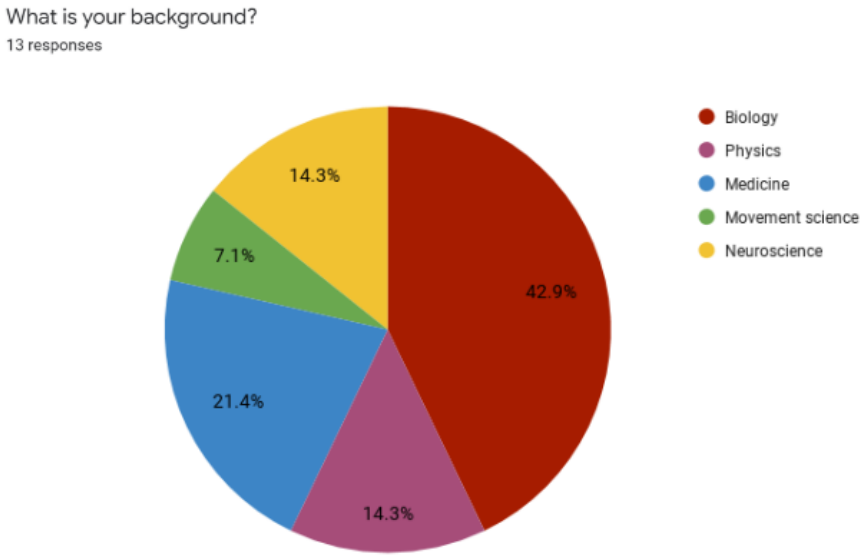

Chart B.1: Results of the first survey question asking about the participants' background.

Chart B.2 presents the results of the second survey question asking how often the participants work with a range of publicly available neuroscience data repositories, shown in a column chart.

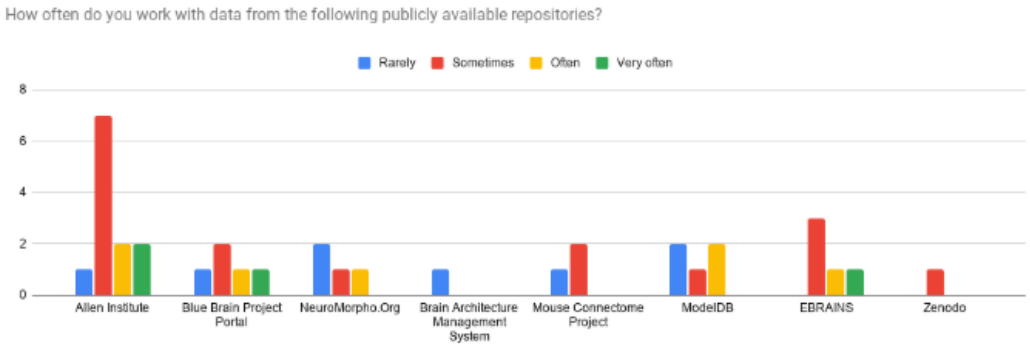

Chart B.2: Results of the second survey question asking how often the participants work with a range of publicly available neuroscience data repositories.

Chart B.3 presents the results of the survey question that asks which data repositories the survey participants use, displayed in a bar chart.

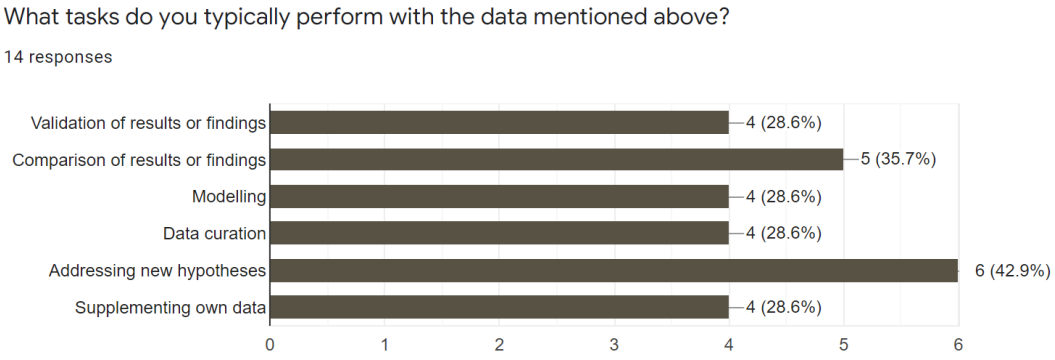

Chart B.3: Results of the survey question regarding which data repositories the survey participants use.

Chart B.4 presents the results of the survey question asking for what tasks the participants use the publicly available data, shown in a column chart.

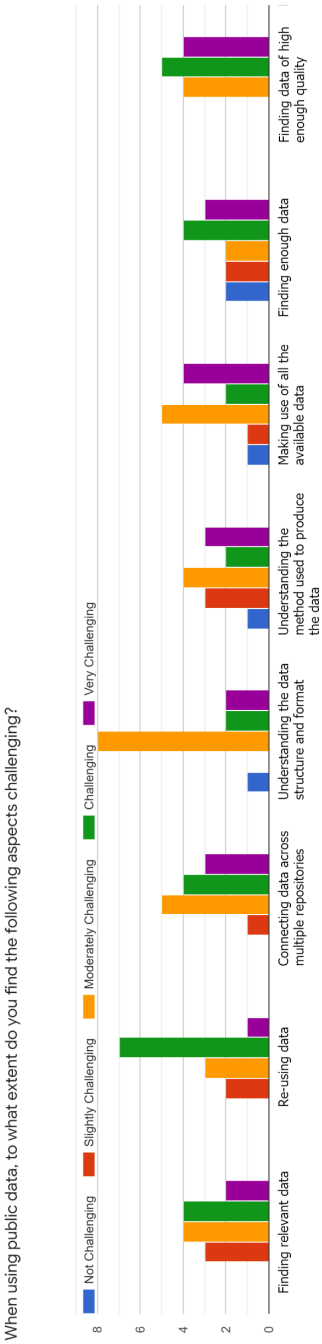

Chart B.4: Results of the survey question asking for what tasks the participants use the data.

The finding that the respondent researchers predominantly use publicly available data to *address new hypotheses* and *compare results and findings* is visible from Chart B.3, where six and five respondents respectively stated this.

The result that *Finding data of high enough quality*, *connecting data from multiple sources*, and *understanding the data structure and format* were found the most challenging was calculated by summing the number of respondents that selected *moderately challenging*, *challenging*, and *very challenging*.

C.2 Survey questions

Usage of publicly available neuroscience data

The goal of this survey is to understand how you work with publicly available neuroscience data in your research.
The results will be used in a master's thesis to better understand how neuroscience data can be made more accessible for researchers.

We collect your email address to make sure the form is only answered once and to be able to get in contact with you. It will be stored for six months on a Google Drive.

* Required

Email address *

Your email

What is your background?

- ◯ Medicine
- ◯ Biology
- ◯ Psychology
- ◯ Informatics
- ◯ Other:

How often do you work with data from the following publicly available repositories?

	Never	Rarely	Sometimes	Often	Very often
Allen Institute	◯	◯	◯	◯	◯
Blue Brain Project Portal	◯	◯	◯	◯	◯

Neuromorpho.org	◯	◯	◯	◯	◯
Brain Architecture Management System	◯	◯	◯	◯	◯
Mouse Connectome Project	◯	◯	◯	◯	◯
Mouse Brain Architecture	◯	◯	◯	◯	◯
CocoMaq	◯	◯	◯	◯	◯
Cell-centered database	◯	◯	◯	◯	◯
hippocampome.org	◯	◯	◯	◯	◯
ModelDB	◯	◯	◯	◯	◯
Neuroelectro	◯	◯	◯	◯	◯
Other sources	◯	◯	◯	◯	◯

If you selected "other", please specify which public repository or repositories:

Your answer

What tasks do you typically perform with the data mentioned above?

- [ ] Validation of results or findings
- [ ] Comparison of results or findings
- [ ] Modelling
- [ ] Data curation
- [ ] Addressing new hypotheses
- [ ] Supplementing own data
- [ ] Other:

When using public data, to what extent do you find the following aspects challenging?

	Not Challenging	Slightly Challenging	Moderately Challenging	Challenging	Very Challenging
Finding relevant data	○	○	○	○	○
Re-using data	○	○	○	○	○
Connecting data across multiple repositories	○	○	○	○	○
Understanding the data structure and format	○	○	○	○	○
Understanding the method used to produce the data	○	○	○	○	○
Making use of all the available data	○	○	○	○	○
Finding enough data	○	○	○	○	○
Finding data of high enough quality	○	○	○	○	○

Are there any other challenges, not mentioned above, you would like to note?

Your answer

Send me a copy of my responses.

Submit

Appendix D

Usability study set-up

We performed a usability study to validate the usability of the web-based user interface of the murine basal ganglia graph data. For each test in the study, there was one observer and one participant. The observer refers to the person facilitating the test, and the participant refers to the researcher testing the user interface. We performed all the usability tests over Zoom. During the test, the participant shared their screen with the observer. This appendix describes the general set-up of a usability test.

The test started with the observer introducing the background and purpose of the study to the participant. The following text was the basis for the introduction:

> We are currently working on the master's thesis; *Graph-based representation,integration, and analysis of neuroscience data — The case of the murine basal ganglia.*
> In the thesis, we research how a graph data model can provide a better understanding of the existing data, how easy it will be to combine with other data sets, and how web-based access can improve the usability of the data. To evaluate these statements, we have built a web application where researchers can interact with the data set. This web application's interface is what we are here to test today.

After introducing the purpose of the test, the observer introduced the participant to the concept of thinking-out-loud. The following text was the basis for this explanation:

> Before we start with the tasks, we ask you to think out loud during the test. This means that you verbalize your thoughts as you move through the website. There is nothing that is wrong to say, and we would rather you say too much than too little.

Before the participant performs the tasks, the observer presented that if the participant cannot complete a task, the participant can say so, and the

observer would provide the next task. The observer also stated that she could not answer questions during the test and provides the next task when the participant states they have completed the task. At this point, the observer asked if the participant had any questions and answered those. Before the tasks commenced, the observer provided the URL for the website and verified that the participant navigated to it correctly.

Table D.1 presents the tasks in the usability study. For each task, the observer presented the task, both orally and through the Zoom chat. A task is either a task or a sub-task, and the observer presented the sub-task(s) only if the participant managed to complete the tasks before. We grouped the tasks into three categories. The first set of tasks, 1-6, are aimed at the analyses; the second set, 7-12, is intended to evaluate the cell type pages; tasks 13-15 are focused on the brain regions pages. The participant was not informed of this grouping and could freely start the navigation on any page.

	Task	**Type**
1a	Can you find how many analyses have been performed on Rattus norvegicus?	Task
1b	Can you find how many analyses have been performed on Mus musculus?	Task
2	How many morphology analyses have been performed on the species Mus musculus?	Task
3	How many analyses, performed on rat, have used the antibody with unique id RRID:AB_476894	Task
4	How many analyses are performed on a juvenile rat (19-28 days)?	Task
5	Can among the analyses find the study by Fujiyama (2016)?	Task
5a	In the study by Fujiyama (2016), how many axonal varicosities in total were observed in the substantia nigra?	Sub-task
5b	In this study, what was the weight range of the specimens used?	Sub-task
5c	In which journal was this study published?	Sub-task
6	See if you can find the study by Echeverry (2004) on NADPHD expressing neurons?.	Task

6a	In this study, what part of the Caudoputamen was covered?	Sub-task
7	How many analyses that have been performed on the rat substantia nigra?	Task
8	Can you find the number of regions that are connected to the rat Caudoputamen?	Task
8a	Which of the connected regions does the Caudoputamen have a very strong, afferent relationship to?	Task
8b	Can you find how these relationships were derived?	Sub-task
9	In the mouse Caudoputamen, how many mixed class neuron cell types are observed?	Task
10	For this region, can you find how many analyses are performed on dopamine 1 receptor expressing cells?	Task
11	Staying on this page, can you get back all the analyses performed on mus musculus?	Task
12	See if you can find a morphology analysis of medium spiny neuron cells.	Task
12a	Select one of these morphologies.	Sub-task
12b	From which repository was the morphology illustration collected?	Sub-task
13	How many cells are returned when searching for "dopamine receptor"?	Task
14	Can you find a description of the cell type "Glia"?	Task
14a	Where was this description collected from?	Sub-task
15	Calretinin expressing interneuron is the cell type investigated in a number of analyses, can you find how many?	Task
15a	Can you find how many brain regions Calretinin expressing interneuron are observed in?	Sub-task
15b	Can you find the number of analyses concerning Calretinin expressing interneuron in the substantia nigra?	Sub-task
16	Can you find the sources and repositories that have contributed to the website data	Task

Table D.1: Usability study tasks

After the participant was done with the tasks, the observer performed an interview to get the participant's overall impression and get feedback exceeding the tasks. The following list presents the questions used as a basis for the interview. As a general notion for all the questions, we aimed not to guide the participants into desired answers. In addition to the presented questions, the observer asked questions specific for each participant based on their task performance, such as asking about a task the participant struggled with or had difficulty solving.

1. What is your overall impression of navigating this data?
2. In this graph database we have integrated data from
   - Morphologies from NeuromMorpho.Org
   - Cell descriptions from InterLex
   - Brain region connectivity from BAMS

   Do you have a comment on that?
3. What properties of the analysis are most valuable for you to search on?
4. Would it be an interesting feature to find similarities between analyses? If so, what properties that you have seen here would be interesting to compare the analyses on?
5. Do you have any other feedback or questions?