ABSTRACT

Title of Thesis: SEMANTIC FOUNDATIONS FOR FORMALIZING BRAIN CANCER PROFILES

Joel Abraham
Master of Science, 2019

Thesis directed by: Professor Mark Austin
Department of Civil and Environmental Engineering
and Institute for Systems Research

With the advent of whole-genome DNA sequencing technologies, tailoring of medical treatment to individual patients based on their genetic makeup has become the vanguard of modern medicine. One such area that can benefit from individualized medicine is that of brain and other Central Nervous System (CNS) cancers. The prognosis of malignant brain cancers is among the worst due to the heterogeneity and complexity of these tumors and their micro-environment. We present a framework that combines data mining and machine learning techniques with semantic approaches for building a clinically-relevant knowledge base of brain cancer profiles. We construct clusters of patients based on the similarity of their profiles using the *k-means* clustering algorithm and extract relevant molecular attributes of these clusters to classify instances of the clusters. We create a semantic model with ontologies, rule checking and reasoning, to enable rational therapeutic regimen selection. Finally, we lay the foundation to incorporate this framework into a digital twin architecture of a patient.

SEMANTIC FOUNDATIONS FOR FORMALIZING BRAIN CANCER PROFILES

by

Joel Abraham

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2019

Advisory Committee:
Associate Professor Mark Austin, Chair/Advisor,
Professor Bilal Ayyub,
Dr. Orieta Celiku.

Acknowledgments

First and foremost I would like to express my gratitude to my thesis advisors Dr. Mark Austin of the Department of Civil and Environmental Engineering and Dr. Orieta Celiku of the Neuro-Oncology Branch at the National Institutes of Health. They both have been instrumental with their counsel and insight on this research. I would like to thank Dr. Austin for providing me an opportunity to be a part of his research and for all his support and positivity throughout the research. I would like to thank Dr. Celiku for initiating this joint venture and for providing guidance in all aspects of this thesis, especially with respect to the biology topics.

I would also like to thank Dr. Bilal Ayyub of the Department of Civil and Environmental Engineering for serving on my thesis committee.

I would also like to express my gratitude to the Neuro-Oncology Branch at the National Institutes of Health for the summer internship opportunity and for being on the vanguard of the fight against brain cancer.

I would like thank my family for their unyielding support, love, and patience. Finally, I would like to thank God for his grace and his wisdom.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

CNS	Central Nervous System
CCLE	Cancer Cell Line Encyclopedia
ML	Machine Learning
NCI	National Cancer Institute
OWL	Web Ontology Language
TCGA	The Cancer Genome Atlas
RDF	Resource Description Framework
SQL	Structured Query Language
SysML	The Systems Modeling Language
Weka	Waikato Environment for Knowledge Analysis
XML	eXtended Markup Language

Glossary of Terms

This glossary provides definitions of key terms employed in this work:

DataType Property: DataType Property defines the relation between instances of classes and literal values, i.e., String using the Protg tool.

Description logic: (DL) is a family of logic-based knowledge representation languages that can be used to represent the terminological knowledge of an application domain in a structured way.

DNA: Deoxyribonucleic acid. A double helix of genetic information vital to the development, growth and function of living organisms.

Extended Markup Language (XML): The extensible Markup Language provides the fundamental layer for representation and management of data on the Web.

Individual: Is a semantic web terminology that represents an instance of a class in the ontology.

Jena: Jena is an open source Java framework for building Semantic Web and linked data applications.

Jena Rules: Jena Rules is an inference (reasoning) engine that plugs into Jena.

Model-Based Systems Engineering: Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases (INCOSE-TP-2004-004-02, Version 2.03, September 2007).

mRNA: Messenger RNA. A subset of the family of RNA molecules which carries genetic information from DNA to the ribosome for protein production.

NCI: National Cancer Institute. The US government's principal agency for cancer research.

Neo4J: Graph Database management system developed by Neo4j, Inc.

No-SQL: Not Only SQL databases. A new paradigm of database management deviating from relational databases. Ideal for large distributed data.

Ontology: A model that describes what entities exist in a design domain, and how such entities are related.

Ontology Class: A placeholder for an entity in the system design. An ontology class may have some dataType or objectType properties.

Ontology Instance: An ontology instance is a specific realization of any ontology class object. An object may be varied in a number of ways. Each realized variation of that object is an instance. The creation of a realized instance is called instantiation.

ObjectType Property: ObjectType Property defines the relation between instances (individuals) of two classes in semantic web terminology.

Ontology Web Language: The Web Ontology Language (OWL) is a knowledge representation languages for defining ontologies.

Reasoner (Rule Engine): A semantic reasoner, reasoning engine, rules engine, or simply a reasoner, is a piece of software able to infer logical consequences from a set of asserted facts or axioms.

Reasoning: To infer new statements based on set of asserted facts in the ontology.

Resource Description Framework (RDF): a model for encoding semantic relationships between items of data so that these relationships can be interpreted computationally.

Rule Checking: A mechanism that ensures existing data in the ontology is consistent with rules defined over the ontology. A rule engine performs this task.

Semantic Web: Refers to W3Cs vision of the Web of linked data.

SQL: Structured Query Language. Standard language used to communicate with relational databases.

SysML: The Systems Modeling Language (SysML) is a graphical modeling language used to define models of systems structure and system behavior.

TCGA: The Cancer Genome Atlas. A database of cancer genomics data.

Weka: Waikato Environment for Knowledge Analysis. An open-source suite of data mining and machine learning tools developed by the University of Waikato.

Chapter 1: **Complexities of Cancer**

1.1 Problem Statement

1.1.1 Cancer: Aberrant Genes

Cancer is a class of diseases characterized by uncontrollable division of abnormal cells. Cancers can develop in nearly all organs and tissues in the body and can proliferate rapidly forming metastases in secondary sites in the body. Cancer is the second leading cause of death behind only heart disease; in 2018, an estimated $1,735,350$ new cases of cancer were diagnosed in the United States [36]. With advances in medicine, cancer death rates have slowly decreased at an average rate of 1.6% per year over the past fifteen years [36], however much work remains in tackling the complexities of cancer.

Cancer is characterized as a genetic disease: it is caused by (inborn or acquired) changes to genes that control the way cells in our bodies grow and divide. Genes are the basic unit of heredity and contain information necessary for making proteins – the building blocks and work force of the cells in the body. Genes are parts of long molecules called Deoxyribonucleic Acid (DNA) which are composed of nucleotides and are packaged into structures called chromatin and chromosomes. The process of reading genes to form proteins involves several steps and is illustrated in Figure 1.1: DNA undergoes transcription to messenger Riboncleic Acid (mRNA), which is then translated to amino acid chains which fold to make a multiplicity of complex three-dimensional proteins. Proteins are involved in most cellular functions in the body ranging from cell signaling to combating viruses and bacteria.

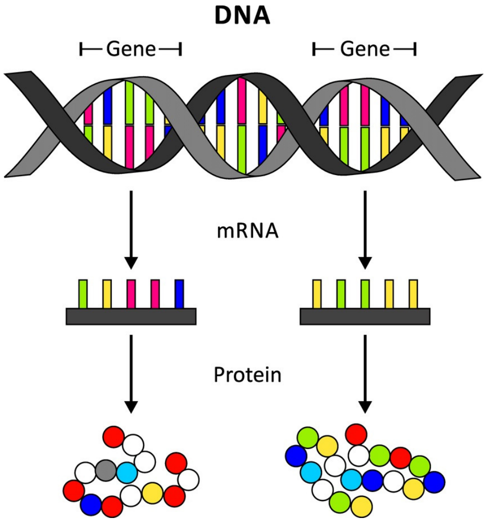


Figure 1.1: Schematic of DNA to protein transformation in cells.

Alterations or mutations in DNA can affect the structure, function and amount of proteins in cells, which in turn can lead to transformation of normal cells to cancerous ones. Normal cells are programmed to develop, divide, and die at tissue-specific intervals. Cancer cells on the other hand develop the ability to resist cell-death signals, or proliferate at faster rates than normal cells of their tissue of origin. While the body has several mechanisms for guarding against such growths, if a sufficient number of such cells escape these guards the result leads to formation of cancerous growths and metastases, as shown in in Figure 1.2.

Not all genetic alternations are detrimental: in fact, genetic variation is what

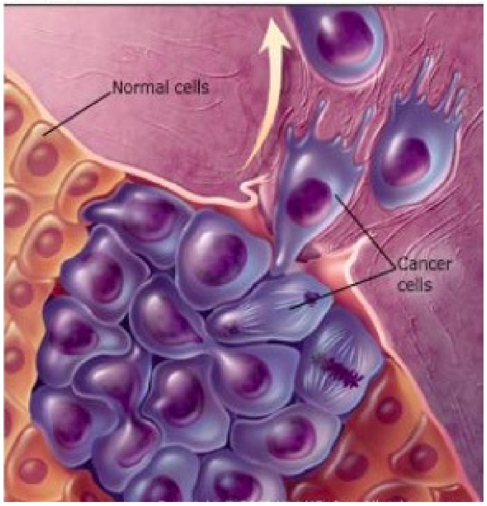


Figure 1.2: Differentiation and metastasis of cancer cells.

enables evolution of species and makes us individuals. Two main classes of genetic alterations can lead to cancerous growths: changes that affect so-called *tumor-suppressor* genes which produce proteins that guard against abnormal growth, and changes that create so-called *oncogenes* which lead to creation of proteins with abnormal structure that may, for example, lead to abnormal turning-on of the signal for the cells to grow and divide. Once aberrant cells develop, they may accumulate more genetic changes which further lead to faster growth and aggressive behavior.

Each person's cancer is also unique in its set of genetic aberrations; even within the same tumor, there can be signficant differences in the genetic alterations between the cells [11]. With over $20,000$ protein-coding genes within the human genome, and various levels of regulation of the process of replication, transcription, and translation, grappling with and understanding the complexity and heterogeneity of cancers is a monumental task for today's researchers. Next-generation sequencing technologies are however making it possible to extract information on genetic aberrations at the single nucleotide level and heralding the age of tackling the problem of cancer at the individual level where each patient's genome is sequenced and analyzed. This new era of biotechnology and big data provides opportunities to tailor treatment to the individual and not just the disease.

1.1.2 Heterogeneity and Complexity of Brain Tumors

Cancer can be classified into different types based on their location in the body and their genetic makeup. This thesis will focus primarily on *Gliomas* which are the largest subset of brain and Central Nervous System (CNS) cancers and originate in the glial, neuron-supporting cells in the brain. Primary brain tumors or neoplasms originate in the brain (as opposed to spreading from other parts of the body). According to the World Health Organization, there are over 130 types of primary brain neoplasms [39]. Of these, nearly 32% are malignant, meaning they divide uncontrollably and 68% are non-malignant. Although not as common as breast and lung cancers, incidences of malignant brain and CNS tumors range from 5.71 to 10.25 per $100,000$ population in the United States. Brain cancers are also the most common cancers diagnosed among children aged $0 - 14$ years [39].

The heterogeneity and location present a significant barrier to diagnosis and treatment of brain and CNS tumors. For example, a study of $1,122$ patient samples of adult diffuse glioma produced seven different subtypes based on distinct genetic signatures [11]. Not only do these tumors differ between patients, but also tumors within individual patients display characteristics from multiple sub-types. As illus-

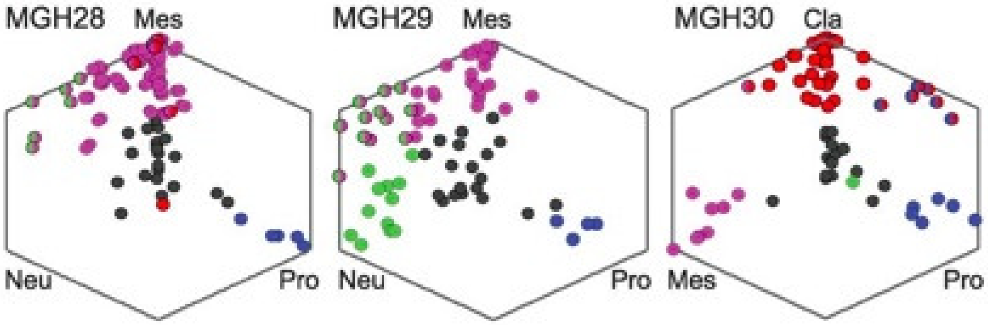


Figure 1.3: Profiling of primary Glioblastoma shows cells with different sub-type characteristics. Each dot represents a cell. Mes, Neu and Pro represent different Glioma subtypes. The location of the cells are based on a classifier score based on all cells in the tumor [41].

trated in Figure 1.3, profiling of primary Glioblastomas (Grade 4 Gliomas) showed that within a single tumor itself, cells can have characteristics of different subtypes of Glioma [41].

Treatment of brain tumors typically may include surgery, radiation therapy and/or chemotherapy. Many patients undergo surgery to remove the tumor, which in most benign cases, is enough to mitigate patient symptoms. Aggressive, malignant tumors such as Glioblastomas, have a high rate of recurrence, which necessitates aggressive treatment that includes radiation therapy or chemotherapy. Radiation therapy is the targeted treatment of cancerous cells using high energy beams such as X-ray or protons. Chemotherapy, on the other hand, administers drugs that work to destroy or damage cancerous cells. These drugs most often target DNA and RNA sequences which halt or slow down cell division. However, many chemotherapy drugs cannot differentiate between healthy and cancerous cells; this leads to severe side effects for the patients such as nausea, fatigue, neurological impairment. Advances in treatment primarily focus on finding gene and protein targets within tumor cells that would enable focusing the effect of treatment on cancer cells. More recent approaches include attempts to recruit the patient's immune system in the fight against cancer cells.

1.1.3 Gliomas

The scope of this thesis covers a subset of CNS tumors called Gliomas. Gliomas originate in glial cells of the brain or the spine. Glial cells support and provide protection to neurons. Nearly 80% of all malignant brain tumors and 30% of all brain tumors are diagnosed as some form of Glioma [22].

Types of Glioma. There are four primary types of Gliomas [24]:

Ependymomas Tumors that arise from a tissue called ependyma.

Astrocytoma Tumors originating in a particular glial cell called astrocytes. Astrocytoma can be organized into two classes: 1. Narrow Infiltration which are mostly noninvasive tumors 2. Diffuse Intfiltration which are high grade tumors with poor prognosis.

Oligodendrogliomas Tumors that originate from oligodendrocytes.

Brain Stem Glioma Tumors that originate in the brain stem.

Gliomas are categorized into grades for diagnosis according to the WHO Classification of Tumors of the Central Nervous System. The grades are determined by pathological and molecular evaluation of the biopsy of the tumor. Gliomas can be characterized as either Low-grade or High-grade:

Low-Grade Gliomas (WHO Grade 2) Well defined and differentiated tumors that are more often benign with better prognosis for the patient. The median survival rate of patients diagnosed is approximate 5.6 to 11.6 years [38]. However, they can progress into high-grade Gliomas.

High-Grade Gliomas (WHO Grade 3-4) Ill defined and undifferentiated malignant tumors that have a worse prognosis. The median survival rate is approximately 3 years. Glioblastoma multiforme is the most malignant of the high grade tumors with a median survival rate of around 8.8 months [38].

The exact causes for the formation of gliomas are still unknown but, as with most other cancers, Gliomas form due to genetic abnormalities that a patient develops or is predisposed to. Access to recent molecular characterization studies of Glioma diagnosed patients have shown that certain molecular characteristics define cohorts within the disease with varying prognosis, that is, likely course of disease [11]. In particular, mutations in the Isocitrate Dehydrogenase I and II gene (IDH1/2) define a subset of Glioma, which along with a specific hypermethylation phenotype shows favorable prognostic outcomes. On the other hand, patients with unaffected, or "wild type" IDH1/2 have poor prognosis. As depicted in Figure 1.4, the study of 1,112 Lower Grade Glioma and Glioblastoma patients can be clustered based on different molecular characterizations. These clusters are shown to have varying survival rates, age of diagnosis, grade and histology. It is evident that a molecular characterization effort to understand and define such clusters can aid a physician in determining the likely prognosis. In this thesis, we use a similar approach to cluster patients based on molecular data and derive functional attributes that enable creation of tumor models to select patients and preclinical models.

1.2 Formalizing Brain Tumor Profiles

1.2.1 Precision Medicine: Whole Genome Sequencing

Diagnosis of common illnesses today is based on symptoms of the patient and other non-molecular clinical factors. With illnesses such as the common cold, this is adequate, however, when dealing with complex diseases such as neurological ailments, there are more nuances that a clinician will have to account for. With the advent of new medical technologies, medicine is shifting from treating the disease to treating the individual. The realization that each of us are biologically unique has led to "omic" assessments that integrate an individual's DNA, RNA and protein data to aid in the diagnosis of the disease [54]. This section provides a brief introduction to the sate of current medical technologies and methods that can be leveraged for precision medicine.

The sequencing of the human genome and the rapid advancement in the speed and reduction of costs in using such technology has been one of the most important achievements of the past two decades. DNA sequencing works by first segmenting the DNA into smaller pieces and copying the DNA multiple times over using bacterial cells. The DNA then undergoes a cycle of copying where a nucleotide base attached with a fluorescent tag is added to the final base of the DNA fragment. At the end of this process, a machine is able to read the bases based on the wavelength of

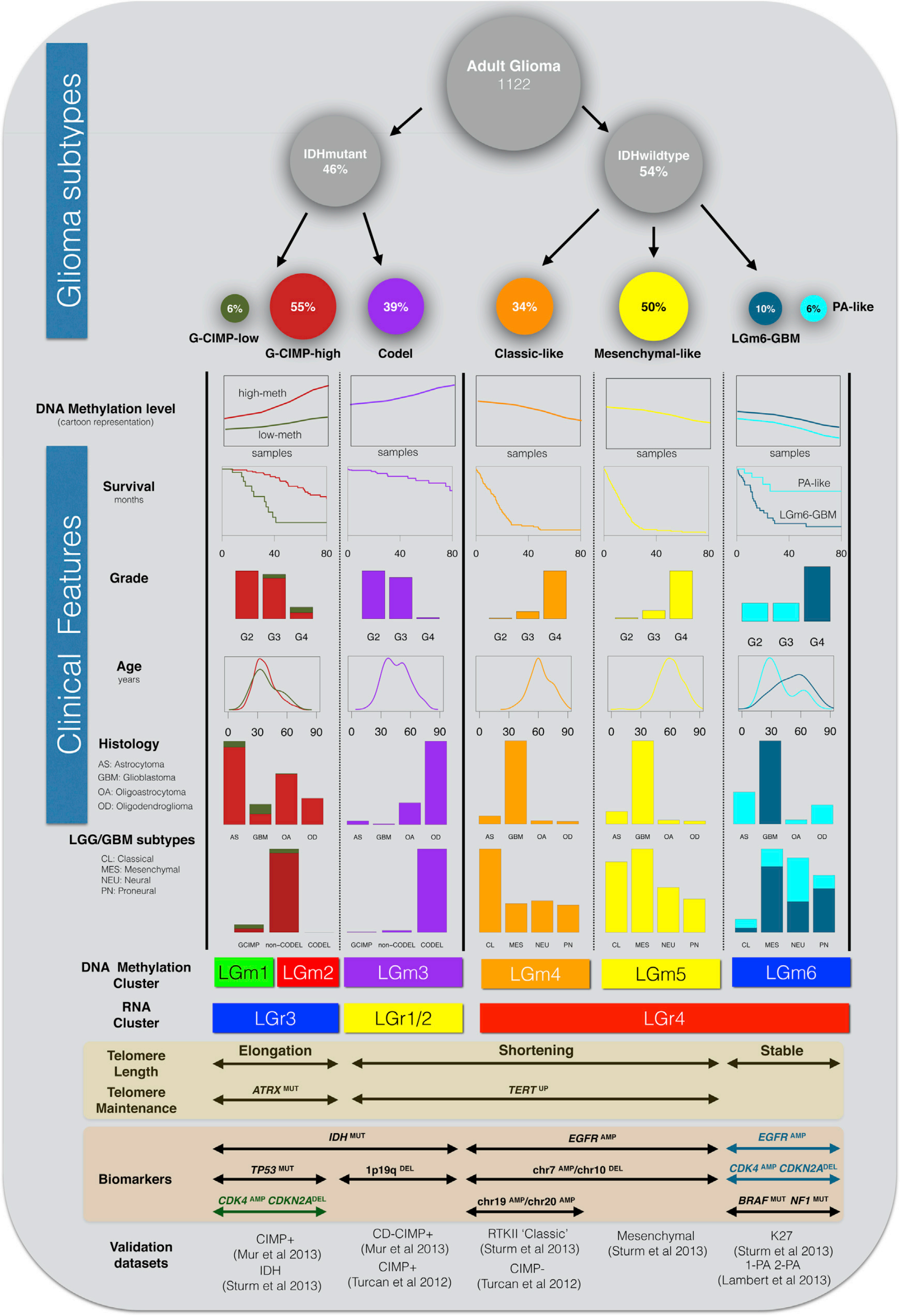


Figure 1.4: Molecular characterization of Adult Diffuse Glioma produces seven different cohorts with varying survival rates, age of diagnosis, grade and histological characteristics.(Source: Caccarelli, et al. [11])

the fluorescent tag. High-throughput sequencing of the genome can be done within 3 to 4 days [54], allowing access to the genetic information of the patient. With this sequencing, the molecular basis of rare or low-frequency variants of diseases have been uncovered [54]. Single-Cell sequencing, the sequencing of the genomic material of individual cells, has also been a major factor in providing insights into disease. DNA sequences obtained from multiple cells within a tumor have shown that there is even heterogeneity amongst the cells of the same tissue [11]. DNA sequences allow physicians to ascertain if there are any mutations or other structural changes (for example, additional copies, deleted copies, fused genes, or insertions of viral DNA) in a person's DNA. As the technology improves and the costs fall, more people will have access to these diagnostic tools, creating vast data sets that can be assessed in detail to be used functionally.

DNA provides a rather static view of the cell; RNA, on the other hand provides a dynamic view of the state of the cell. RNA is the precursor to protein in the cell; a RNA based assay shows what is being produced in the cell at the time of the assay. Although not as streamlined as genome sequencing, RNA sequencing technologies are a relatively new set of tools that provide a dynamic view of the patient. These tools capture data on gene fusions, spliced transcripts and the whole gamut of RNAs including microRNA, and ribosomalRNA [54]. With RNA sequencing, physicians are able to quantify cellular changes between a healthy and a diagnosed patient. Theses changes, usually measured as mRNA expression, enable physicians to understand which genes are active and inactive for a certain disease and enable targeting of pathways associated with the translation of such genes.

Epigenetics is the study of the modification in gene expression through regulatory molecules that affect the genetic code. The epigenome consists of chemical compounds that instruct the genome what to do without altering the code itself. These compounds usually attach themselves to DNA and can turn on or off DNA to direct protein expression. Of the myriad of compounds, methyl groups – three hydrogen molecules attached to a carbon molecule – have been mapped extensively within 200 different cell types of the body [54]. Unusual epigenetic markers often indicate altered states within cells and this can be leveraged to differentiate diseases and find potential pathways for drug-targeting.

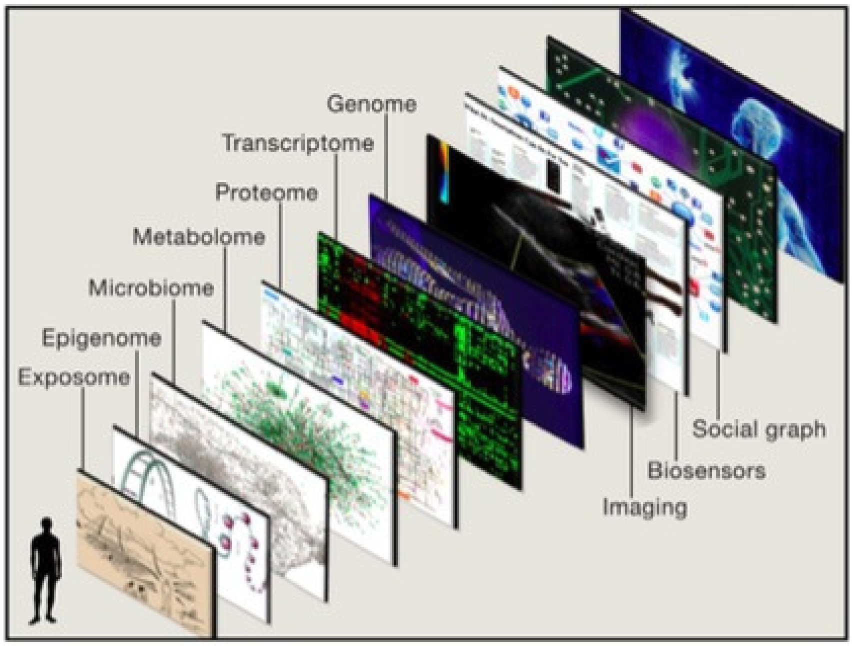


Figure 1.5: The integration of patient data from the genome to the patient's environment is paving the way for new treatment and diagnostic paradigms (Source: Topol [54])

It is evident from Figure 1.5 that the paradigm of treating the disease is shifting to that of treating the patient. With the availability of data from different domains, it is now possible to digitize the patient. The key going forward will be the development of new frameworks and algorithms used in conjunction with the integrated data set. In this thesis, we integrate data from the genome in the form of mutations and copy number changes, from the epigenome in the form of methylation sites and from the transcriptome in the form of mRNA expression. We provide a framework that integrates information from these dimensions of the patient genome and utilize the multi-dimensional data to generate models and rules that can guide clinical decisions for patients diagnosed with Gliomas.

1.2.2 Preclinical Models

Before a new cancer drug can undergo clinical trials, which test the therapy on human patients, they must undergo extensive preclinical studies. Preclinical studies are performed on (experimental) model systems. The two main categories of experimental models are: *In Vitro*-cell cultures, where cancer cells (originating from humans or animals) are grown in Petri dishes under controlled conditions and subjected to experimental treatments; and *In Vivo*-animal models, where cancer cells are implanted in a live animal, or cancer is induced to form in the animal and the treatment is administered to the animal. These cell and animal models are an integral part of the extensive preclinical studies performed to understand the efficacy and potential toxicity of new cancer drugs. The drug development life cycle depicted in Figure 1.6, is composed of drug discovery, preclinical trials, clinical trials and FDA reviews and approvals. The development of a drug is often a decade long process with the majority of the time spent in development and preclinical testing.

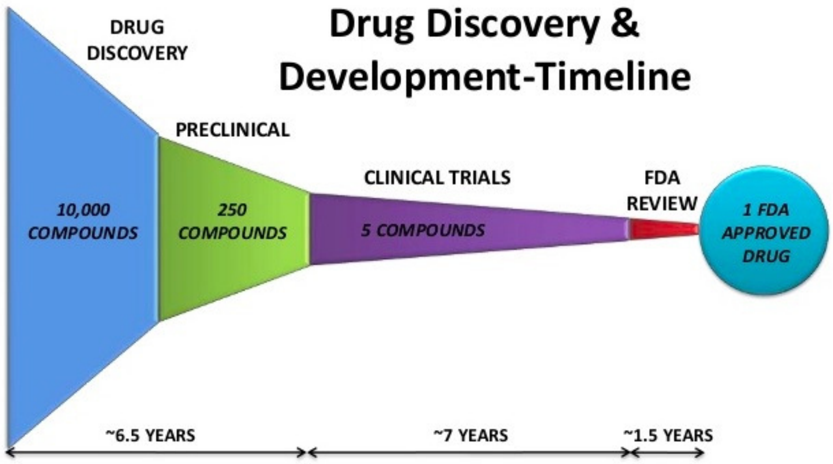


Figure 1.6: Timeline for drug discovery and development.

Animal and cell models have been a mainstay of preclinical testing. However, the complexity of aggressive cancers, and their interplay with other systems of the organism (immune, organ, cellular) make it impossible for the cell line and animal models to faithfully represent the human disease [29]; therefore, even therapies that are proved successful in preclinical studies often fail to show efficacy in clinical trials. In fact, only about a third of highly cited preclinical trials enter clinical trials, and out of those only about 8% of these drugs pass phase 1 of clinical trials [29].

Recent efforts such as the Cancer Cell Line Encyclopedia [7], have profiled and compiled genetic characteristics of cancer cell lines and their response to a collection of drugs, in search of genetic predictors of drug sensitivity [7]. In this thesis we develop a framework that will enable characterization of patient cancer profiles alongside cell line profiles, and reasoning about their similarity. We will identify genetic markers that characterize molecular subtypes within Gliomas and use these markers to identify preclinical models that share similar genetic characteristics. With a data driven semantic model, we hope to bridge the translational barrier between preclinical and clinical trials.

1.3 Project Vision: Systems Support for Personalized Medicine

The long-term objectives of this work are to develop and validate methodologies and tools for the personalized treatment of brain cancer patients. The proposed approach employs ideas in systems development with design platforms and digital twins, and is supported by semantic modeling/machine learning techniques for reasoning with medical domain knowledge and various forms of patient-specific data.

1.3.1 Design Platforms

Design is a transformational process that takes a specification and turns it into a product. The way in which this process is organized is called a methodology. As systems become progressively more complex, and time-to-market constraints progressively more stringent, the relative cost of systematically exploring design spaces to find good designs, and then verifying and testing behavior will steadily increase unless new approaches are developed.

We define platform-based design [48] as the creation of a stable architecture that can be rapidly extended, customized for a range of applications (instead of a single product), and delivered to customers for quick deployment. Platform-based design methodologies improve the efficiency, correctness and economics of design by: (1) restricting the space of design options to pre-defined components, connectors, and rules for assembly (all contained in a library), and (2) providing designers with

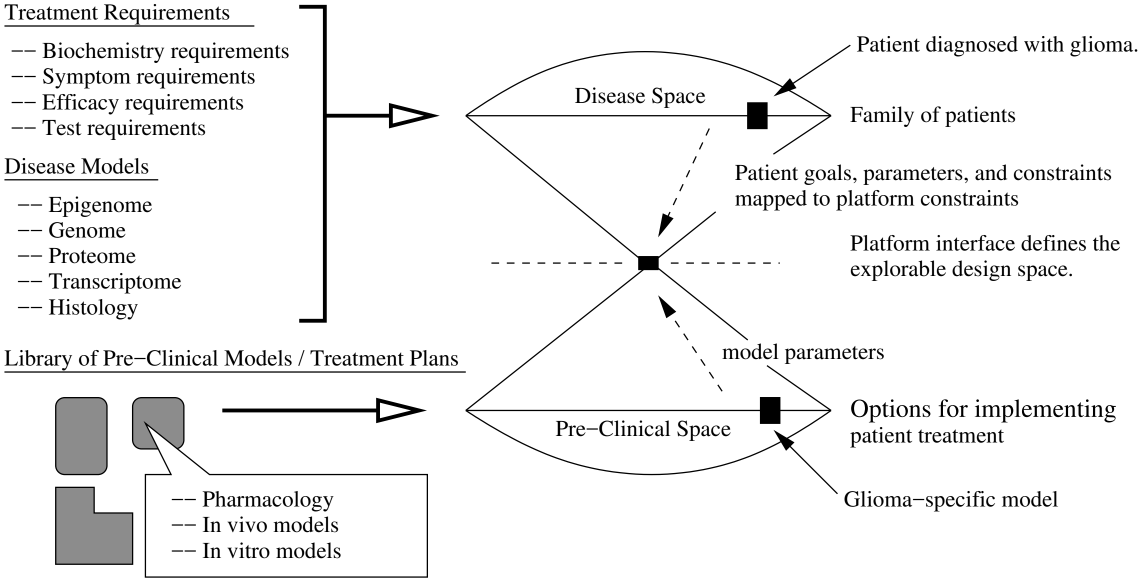


Figure 1.7: Platform-based design for diagnosis and treatment of cancer patients.

the ability to look ahead and reason with libraries of available options. Design is more efficient because an engineer working on abstraction level n can improve the quality of decision making by looking ahead to information at lower-level abstraction levels (n+1, n+2, ...). The latter reduces both the number of required iterations of development and large loop corrections. Design is more correct because systems can only be assembled from components and connectors that have already been developed and are known to work.

While these techniques were initially developed in the late 1980s and 90s for the design of electronic, automotive and aircraft systems [26,47,50], it is now evident that the same approaches add value to the design of experiments needed for accurate development of biomedical devices [34]. A second emerging opportunity is synthesis of patient treatment plans in the medical domain. As a case in point, Figure 1.7 shows how the design of patient treatment plans can be viewed as a meet-me-in-the-middle process where a top-down refinement of patient's medical condition and constraints on medical treatment (i.e., patient/medical application space), meets with an architecture (glioma models and treatment options) space of potentially good implementations for treatment of patients, plus constraints and measures of effectiveness for evaluating success.In order to capitalize upon the added capabilities of such a technique, two key tenets of this work are that: (1) methods to succinctly model a breadth of biological systems must be developed, and (2) these models must be able to integrate with system-level models capable of describing the performance of the entire doctor-patient-healthcare system. Current methods and techniques for cancer patient treatment are simply are not capable of such full-system modeling.

1.3.2 Digital Twin Architectures

While platform approaches to design focus on efficiencies at the the front-end of system development, digital twins are expected to provide decision making support throughout the system life cycle. For the application domain at hand, this corresponds to the complete period in which a cancer patient is provided medical treatment.

A digital twin is a cyber representation of a system that mirrors its implementation in the physical world; this is achieved through modeling of system structure and behavior plus real-time monitoring and synchronization of data associated with events. The latter are made possible by remarkable advances in sensing, communication, and AI technologies that have occured over the past few decades. From a

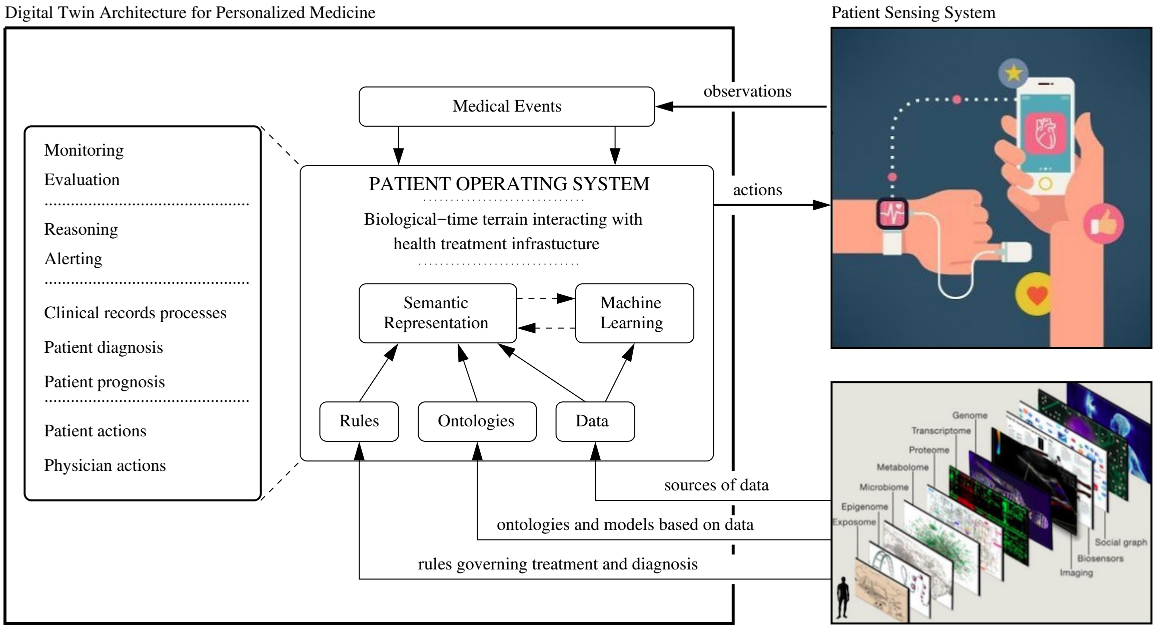


Figure 1.8: Digital twin architecture for personalized medicine.

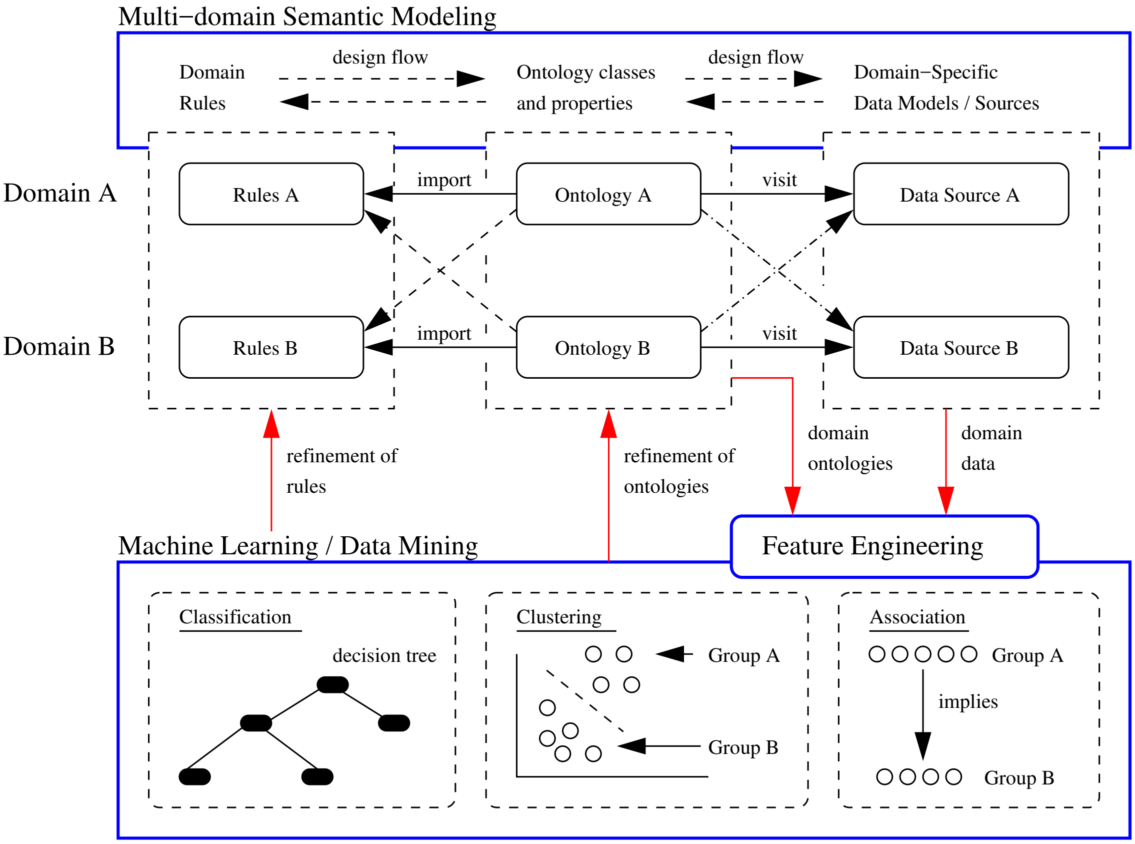


Figure 1.9: Semantic modeling integrating multiple domains using rules and inferences gained from machine learning and data mining techniques (Adapted from: [4, 12]).

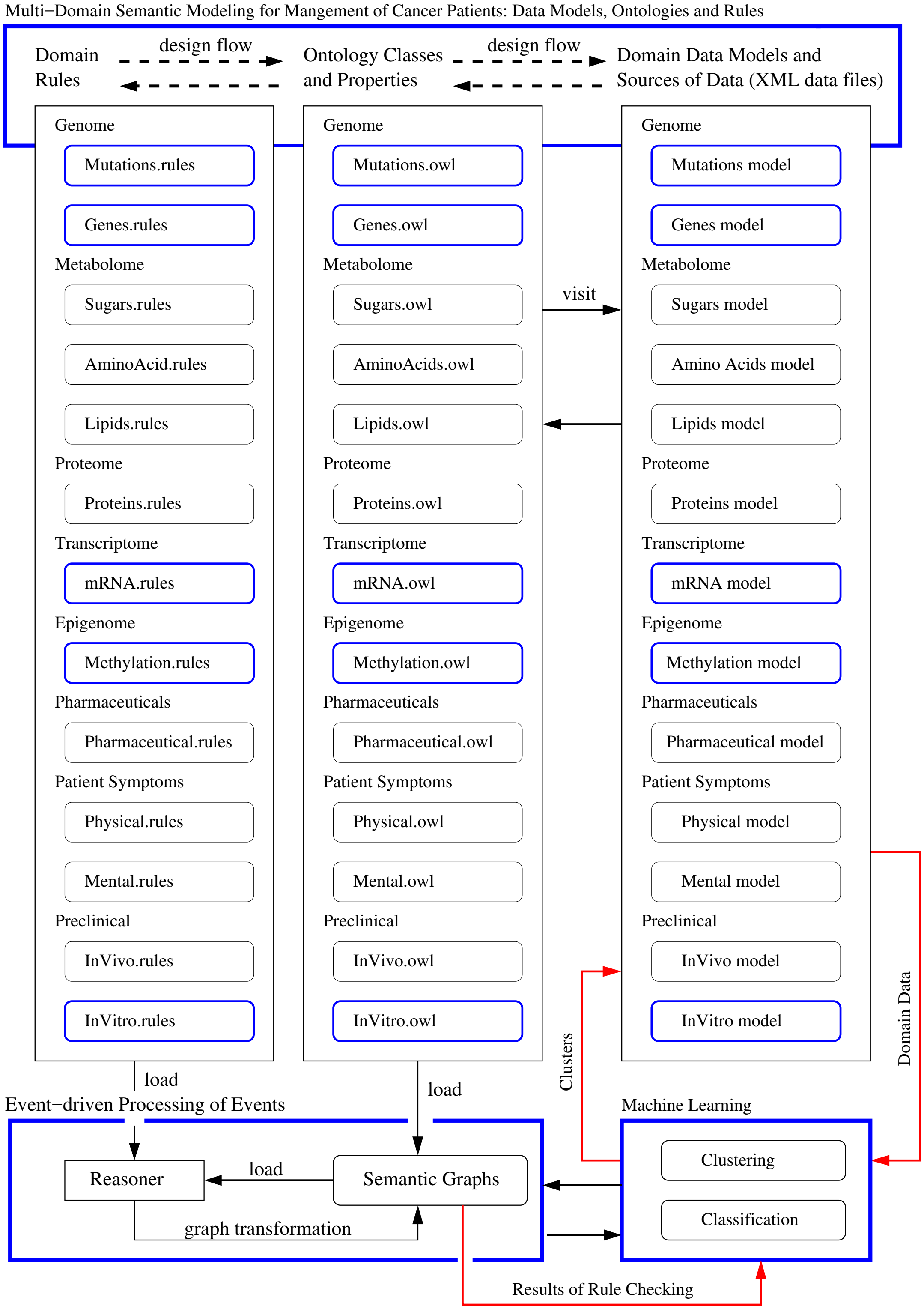


Figure 1.10: Abbreviated digital twin architecture for a combined semantic and machine learning approach to real-time monitoring and treatment of cancer patients. Focus areas for this thesis are highlighted in blue.

temporal standpoint, the associated software and algorithms work to provide simulation and optimization support for forecasting of near-term system performance and long-term planning. The digital twin concept [21] was initially proposed in the 2000-2010 era as a way to support the design and operation of air vehicles for NASA. Since then the range of potential applications has expanded to include automotive components, manufacturing processes, power plants, and smart cities, among others [21,27,33]. Within the systems engineering community, Siemens now sees digital twins being the successor to procedures for model-based systems engineering [10]. The associated view within the healthcare community is that digital twin technologies that embrace open ecosystems and services can open the door to improved clinical services and improved economics [18].

Figure 1.8 is a high-level schematic for digital twin architecture for personalized medicine. In this setup, streams of patient data will be collected by wearable devices, integrated with a patient's clinical data and transmitted to a "patient" digital twin that works as an operating system to identify medical events and then match details of the biological-patient terrain to feasible plans for health treatment.

1.3.3 Combined Semantic and Machine Learning Approach

This work explores a combined approach to formalizing brain cancer profiles, where semantic models and machine learning techniques work collaboratively as a team to represent and reason with various types of patient data and medical domain knowledge to determine recommendations for doctor action and patient treatment.

The interaction of semantic modeling and machine learning techniques can be succinctly represented by the architectural template shown in Figure 1.9. Semantic models are ideal for the development of ontologies and inference rules of the Glioma domain. In contrast, basic machine model are ideal for the identification of classification, clustering and association relationship in data, and for the identification of anomalies in streams of patient data. The architectural template employs feature engineering [57] that allows to find or define features that enable ML algorithms to work. Feature engineering begins with raw data from which relevant or useful features are extracted and formatted as inputs of the ML algorithm. Clustering identifies groups of objects in the domain that are related and decision tree classifiers identify rules that maximize the likelihood of prediction for a target. Association algorithms look for rules that strongly correlate different features of data that enable the creation of rules that can span multiple domains. Finally, this template allows dynamic changes to the knowledge base where new data can be easily ingested and rules likewise updated.

Figure 1.10 takes the architectural template and customizes it to cover the range of concerns one might see in a full implementation of the project vision. The individual rows – data, ontologies and rules – represent the various domains of interest, including those already introduced in Figures 1.5, 1.7 and 1.8. The lower portion of Figure 1.10 depicts how ontologies are imported into a semantic graph, rules are executed via a reasoner, and how the semantic graphs responds to any incoming data or events triggering a graph transformation.

1.4 Thesis Contributions and Organization

Figure 1.11 is a flowchart of the research activities covered by this thesis. On the semantic side of the problem, the scope of investigation covers development of ontologies and rules for domains highlighted in blue in Figure 1.10. These activities are supported on the machine learning side with procedures for clustering and classification of patient data. The clustering and classification analyses are handled by MATLAB and Neo4J, and Weka, respectively.

Sources of Data. Lower Grade Glioma and Glioblastoma patient data was obtained from The Cancer Genome Atlas (TCGA). TCGA is a program initiated by the National Cancer Institute to molecularly profile over 20,000 patients samples covering 33 different cancer types [53]. A comprehensive overview of the data is provided in Section 3.1.1.

Contributions. The contributions of this thesis are as follows:

1. We propose that a semantic approach to the CNS cancer domain will lower the translational barrier of laboratory therapies not effectively working in clinical trials by:

   - Enabling the rational selection of preclinical models.
   - Enabling selection of patients for clinical trials based on similarity to such models.

   In order to accomplish this, we provide the initial steps and framework to

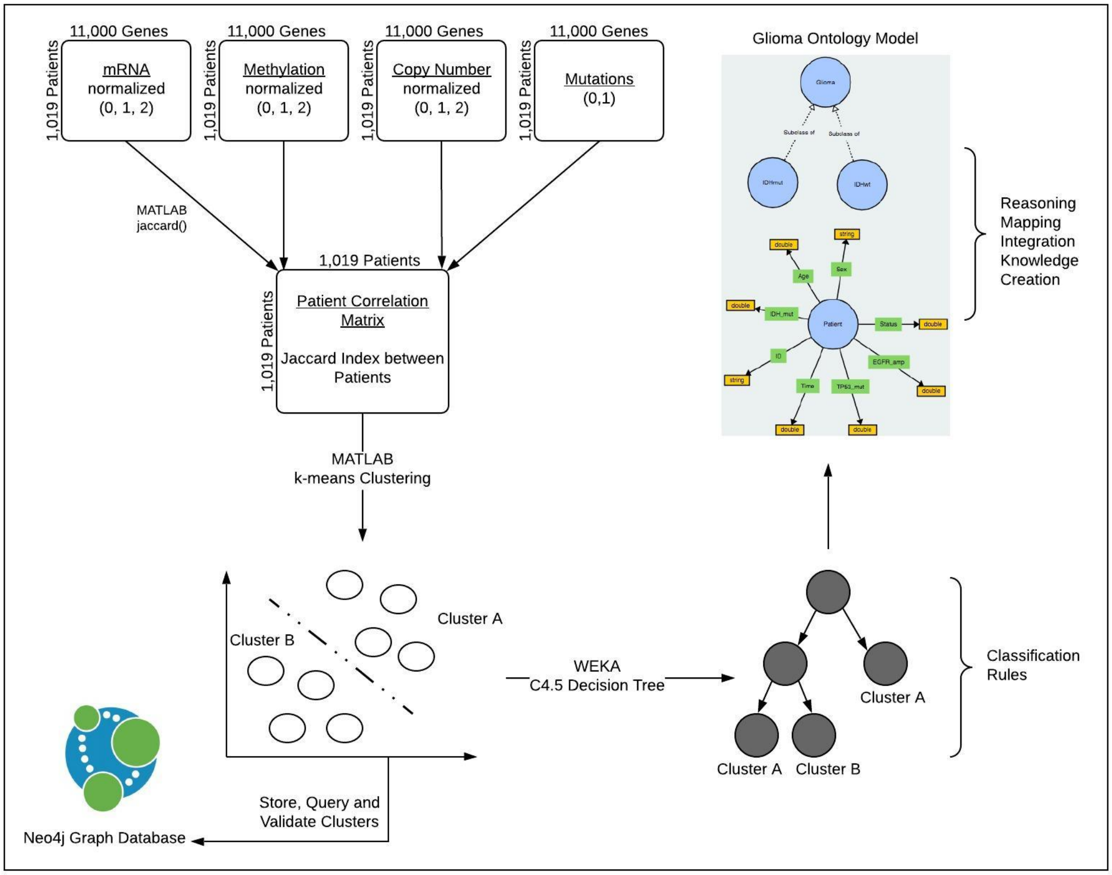


Figure 1.11: Flowchart of research activities covered by this thesis.

create a semantic model that incorporates patient genome data to be used for model selection. We utilize the Semantic Model and Machine Learning Architectural template [4,12] to create a semantic model that encompasses the Glioma domain with rules that allow mapping of patients to relevant preclinical models.

2. We develop prototype ontologies and rules for:

Glioma Ontology A simplified knowledge graph of terms associated with Glioma, including sub-types with their hierarchical relationships.

Patient Ontology A knowledge base of terms and data types associated with features relevant to a patient.

Cell Line Ontology A simplified knowledge graph of attributes and terms associated with cancer cell line models.

Mapping Rules Rules derived from ML algorithms that enable mapping of patients and cell line models to prognostically relevant clusters.

Individuals Instances of patients and cell line models ingested into the ontology from different data sources.

3. We provide a framework for tackling high-dimensional whole genome sequenced data by employing ML algorithms to cluster data into similar clusters. We then utilize classifier algorithms to extract features that are relevant to these clusters to be used in the mapping of the patients to the models.

4. We provide a graph database of the data and the results. We employ the Neo4j graph database platform to store, query and validate the data. The database allows for the querying of relevant clinical and molecular attributes of each cluster and individual patients and allows for visualization of the clusters.

Organization. The thesis is organized as follows: Chapter 2 covers related work in semantic modeling and semantic web technologies, and basic capabilities of machine learning. Chapter 3 provides the framework to use unsupervised ML to create clusters from high-dimensional patient genome data and the creation of a graph database to query and visualize clusters. We provide an overview of the data used for analysis and the formal description of the *k-means clustering* algorithm used to cluster the data. Chapter 4 provides a formal description of the classifier algorithm and its results extracting features from the clusters. Chapter 5 describles the steps and tools used to create the Glioma specific Semantic Model. We also provide an application of the semantic model in mapping preclinical models to prognostically relevant clusters. Finally, Chapter 6 provides the conclusion and future work

Chapter 2: **Related Work**

2.1 The Semantic Web

2.1.1 Semantic Web Technologies

The World Wide Web is a network of machines that allow linking of documents through hyperlinks. It was created with the initial purpose for the sharing of information among members of the scientific community. Early versions of the World Wide Web only allowed for the retrieval of documents and interpretation of these documents by the end-user. The Semantic Web is an extension of the World Wide Web that aims to imbue semantic data into the network allowing machines to access, share and automatically discover new knowledge [23, 49].

Applications that access data from many sources and from large databases will benefit from the automatic machine aided assistance in the creation of knowledge. To this end, the Semantic Web utilizes markup languages to introduce, coordinate and share semantic data and offers the ability to reason and draw inferences via ontologies. The Semantic Web provides an ideal framework to create models that integrate different domains, react to new data and allow for automatic reasoning; all of which will be crucial for precision medicine.

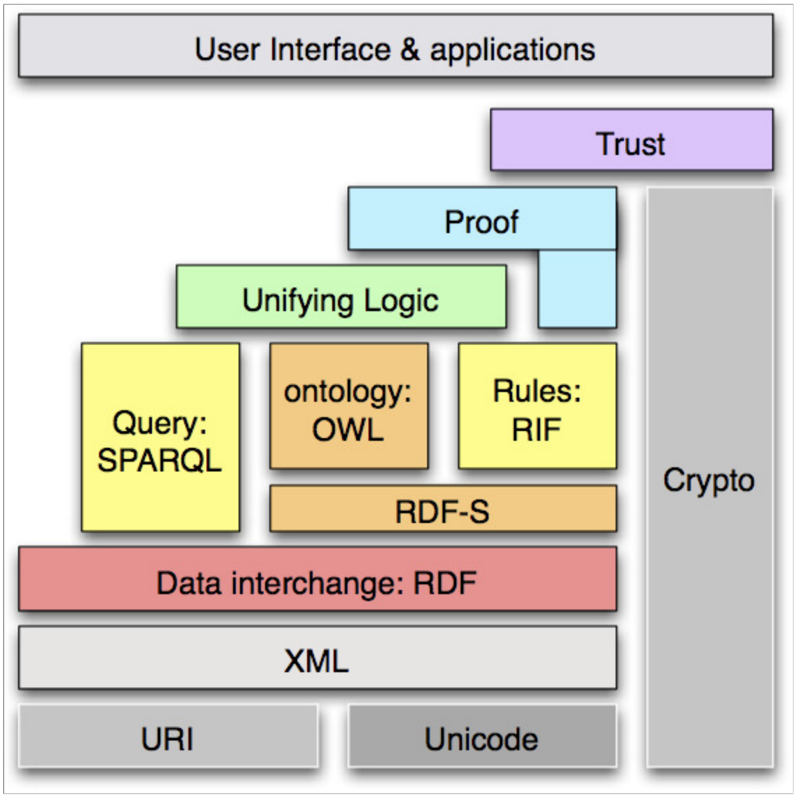


Figure 2.1: The Semantic Web Layer Technologies (Source: Feigenbaum, L. [17]).

Figure 2.1 presents the technical infrastructure supporting the Semantic Web and the framework to construct and employ a semantic model. Each layer is built upon the capabilities of the lower layer with the top-most layer providing interfaces for applications with the intent of knowledge discovery and reasoning. URI and Unicode allow for identifying resources on the web and linking documents. The extensible Markup Language (XML) provides the layer for representation and management of data. XML allows semantic web applications to gather information from various sources on the web. Resource description framework (RDF) allows representation of the data from web sources in a graph model. The graph representation of data allows for the hierarchical representation and querying of data. Finally, the web ontology language (OWL) provides semantic meaning to the model and the data. These technologies in conjunction provide a framework for reasoning on multi-domain data, which for precision medicine applications, is a crucial element for knowledge codification and creation.

2.1.2 Web Ontology Language (OWL)

Description Logic (DL) are a family of formal knowledge representation languages used in artificial intelligence to represent and reason about concepts of a domain. DL is used in the biomedical domain to codify and reason over biomedical knowledge. In information science, ontologies capture a domain's definitions, properties and/or attributes of data, classes, relations and individuals (or instances). Ontologies are analogous to a class hierarchy and datatypes found in object-oriented design (OOP). Unlike OOP, ontologies capture domain structure that assert relationship of domain entities (e.g.: subClassOf) and enable reasoning over multiple domains.

The Web Ontology Language (OWL) is a DL-based knowledge representation language used for the construction of ontologies [1]. OWL is built upon the RDF concept and adds structure and vocabulary for describing properties and classes. OWL allows property definitions, class restrictions and hierarchies and provides an infrastructure to use first order logic to reason and infer new knowledge. Figure 2.2 presents an example of how a domain entity, a car, is captured in OWL.

In Figure 2.2, Company, Factory and Car are classes which have their own definition and attributes already defined in their respective ontologies. The relationships

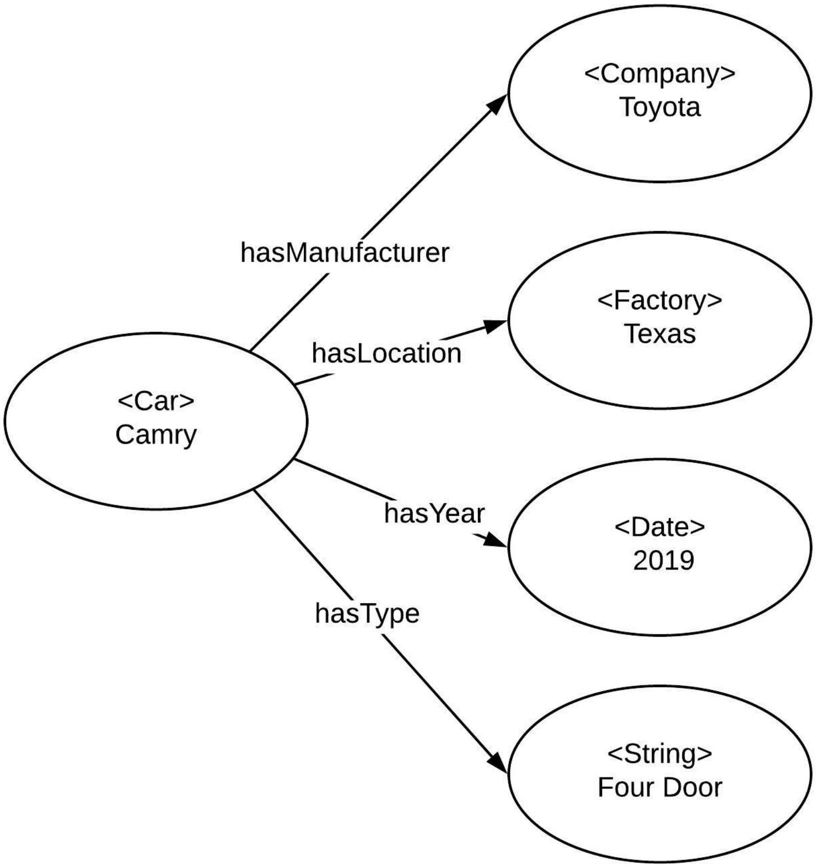


Figure 2.2: An OWL graph of attributes and relationships describing a car.

between different domain entities (hasManufacturer and hasLocation) are object properties that specify a relationship between a pair of resources or nodes. Date and String are connected via datatype property (hasCompletionDate and hasType) and are akin to datatype in OOP. Figure 2.3 shows how OWL represents these nodes and relationships formally. OWL is powerful in that it provides a framework to order multi-domain data and the infrastructure to deploy a reasoner on the data.

```
// Classes...

<owl:Class rdf:about="http://www.example.org/carOntology#Car"/>
</owl:Class>

<owl:Class rdf:about="http://www.example.org/carOntology#Company"/>
</owl:Class>

<owl:Class rdf:about="http://www.example.org/carOntology#Factory"/>
</owl:Class>

// Datatype Properties...

<owl:DatatypeProperty rdf:about="http://www.example.org/carOntology#hasType">
    <rdfs:domain rdf:resource="http://www.example.org/carOntology#Car"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="http://www.example.org/carOntology#hasYear">
    <rdfs:domain rdf:resource="http://www.example.org/carOntology#Car"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#dateTime"/>
</owl:DatatypeProperty>

// Object Properties...

<owl:ObjectProperty rdf:about="http://www.example.org/carOntology#hasLocation">
    <rdfs:domain rdf:resource="http://www.example.org/carOntology#Car"/>
    <rdfs:range rdf:resource="http://www.example.org/carOntology#Factory"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="http://www.example.org/carOntology#hasManufacturer">
    <rdfs:domain rdf:resource="http://www.example.org/carOntology#Car"/>
    <rdfs:range rdf:resource="http://www.example.org/carOntology#Company"/>
</owl:ObjectProperty>
```

Figure 2.3: OWL definition of a car.

2.1.3 Jena and Jena Rules

Apache Jena [2] is an open source Java framework that allows creation of Semantic Web data applications. Jena is primarily used to create and manipulate RDF (resource description framework) graphs and provides APIs that enable developers to utilize OWL (web ontology language) and SPARQL (RDF graph query support) frameworks. Once a semantic model is created, Jena supports the querying, transformation and reasoning of the model. Jena provides standard but limited querying capabilities which span from listing all the statements in the model to selecting statements based on attributes and/or subjects. It also provides three operations; union, intersection and difference to merge and manipulate data from disparate sources. Finally, Jena provides a reasoning platform to dynamically alter the semantic model and to infer knowledge.

Jena utilizes a rule-based reasoning approach; the knowledge-based system is developed by deduction, induction and abduction methods from a starting set of data and rules. Jena provides inference engines or reasoners to utilize and transform the semantic model. Reasoners provide means to derive additional RDF statements from a base RDF graph with ingestion of new data and the axioms and rules associated with the reasoner. Jena Rules reasoner engine is used as part of this thesis. The RDF knowledge graph along with the reasoner makes the semantic model a dynamic and responsive model capable of integrating multi-domain data.

2.2 Semantic Modeling: Ontologies and Rules supported by Data

Figure 2.4 presents a framework for the implementation of semantic models using ontologies, rules, and reasoning mechanisms. From a data science community perspective, an ontology is a set of knowledge terms that includes vocabulary, semantic interconnections, and some simple rules of inference and logic for some particular topic [23]. To provide a formal conceptualization within a particular domain, and thereby facilitate communication and reasoning among domains, ontologies need to accomplish three things: (1) Provide a semantic representation of each entity and its relationships to other entities, (2) Provide constraints and rules that permit reasoning within the ontology, and (3) Describes behavior associated with stated or inferred facts.

System data models contain the data and relationships among data needed to build models of system structure and system behavior. For medical domain models, this information will consist of catalogues of knowledge defining the disease space, positioned alongside data collected from specific patients. The semantic counterpart of medical domain models is ontologies (class hierarchies), individuals (graphs), and rules. Data contained within the medical domain models will be ingested into the semantic model as data property values. Relationships (including dependencies) among the various classes will be represented as object properties. For the semantic modeling of complex multi-domain applications it is common practice to organize ontologies into hierarchies of knowledge. The middle and lower sections of Figure 2.4

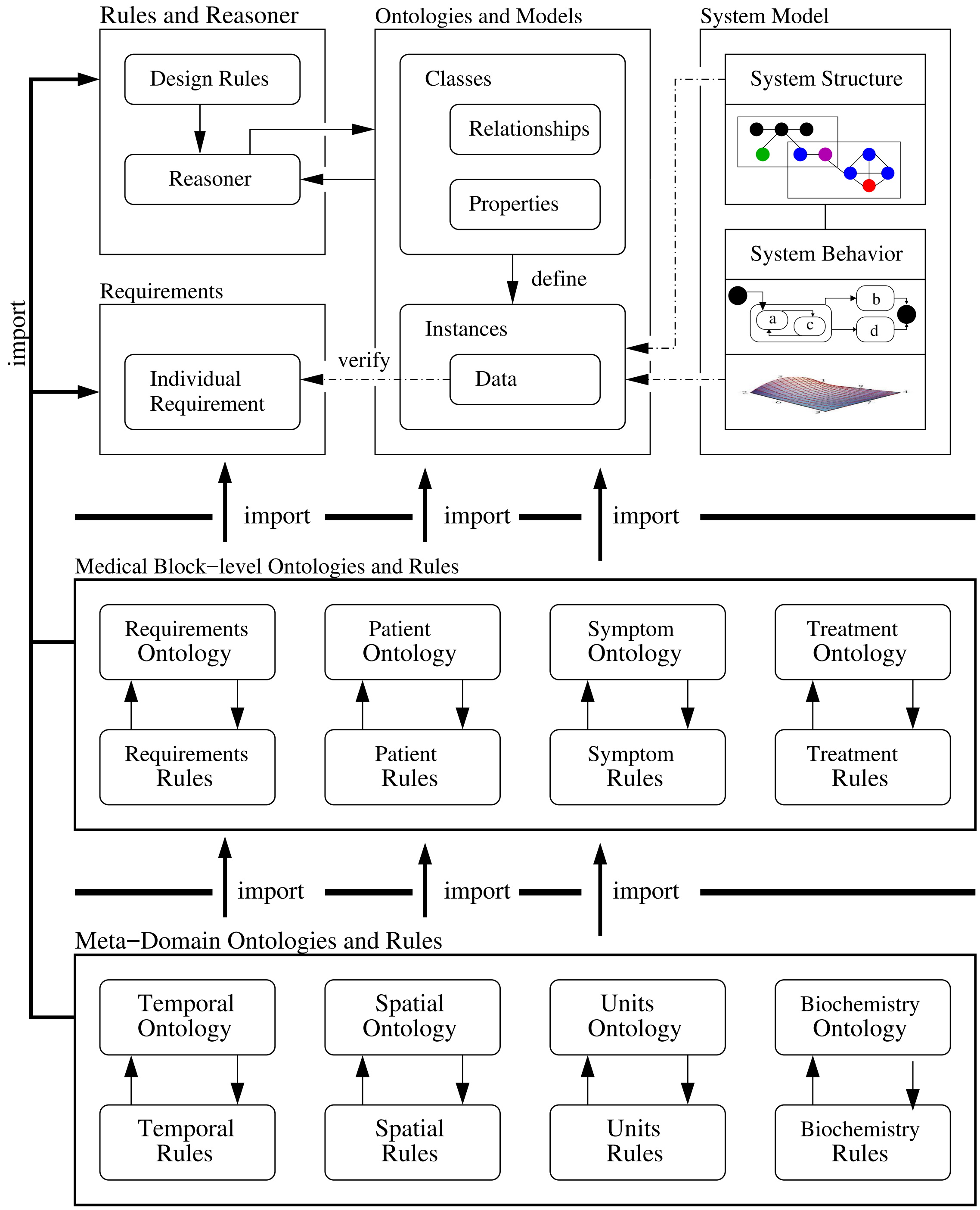


Figure 2.4: Framework for semantic modeling in medical domain applications.

illustrate, for example, one such organization for the brain cancer domain. Patient, symptom, and patient treatment ontologies are directly applicable to our domain of interest, but also apply to the solution of medical problems outside brain cancer. They can import and use top-level ontologies representing general concepts such as time, space, and physical units that apply to and cut across many domains. One important difference between many engineering systems and the medical domain is that the latter is concerned with systems that are living. The basic formal ontology (BF0) [3] is an effort to provide medical practitioners (among others) with sets of carefully designed ontologies for the description of general concepts. BFO has found considerable success in the medical and biomedical domains [51].

Rule-based approaches to problem solving provide several advantages: (1) rules that represent policies are easily communicated and understood, (2) rules retain a higher level of independence than logic embedded in systems, (3) rules separate knowledge from its implementation logic, and (4) rules can be changed without changing source code or underlying model [46]. They are particularly beneficial when the application logic of a problem domain is dynamic, and where rules are imposed on the system by external entities. These conditions apply to a wide range of problems in systems engineering and analysis (e.g., semantic modeling for cyber-physical systems [14, 15, 42], traceability of requirements to component-level behaviors [13], component-based modeling, design and trade-off analysis with RDF graphs [35], validation of connectivity relationships in component-based systems [6] and behavior modeling of distributed systems [5]).

2.3 Machine Learning: Uncovering Patterns in Data

Modern-day machine learning (ML) techniques provide insights - sets of patterns and behavior - to large amounts of data. These techniques and tools are used ubiquitously in domains ranging from smart cities and buildings [4, 16] to bioinformatics. Raw data, especially in the case of whole genome patient data, is often, to the beholder, a monolith with no discernable dependencies or patterns that can be easily modeled from first principles. However, today's ML and data mining tools leverage statistical methods to extract functional data from large data sets to be used for diagnostic and prognostic needs. This section provides a brief overview of the two primary flavors of ML and their application to this thesis.

ML techniques can be divided into two broad categories, unsupervised and supervised learning.

1. Unsupervised learning tries to find the underlying structure and pattern to a set of data where no label or 'right answer' is specified. Unsupervised learning attempts to decipher what features of the data can be used to find partitions or labels in the data that can be modeled. Common unsupervised algorithms include *k-means clustering* and *convolutional neural networks.* Labeling of patient data is often at the diagnostic level based on disease histology and identification of a set of key biomarkers. For prognostic purposes, it is required to go deeper into the data to find what specific attributes contribute to different variations of the same disease. Adult Glioma, for example, can be divided into

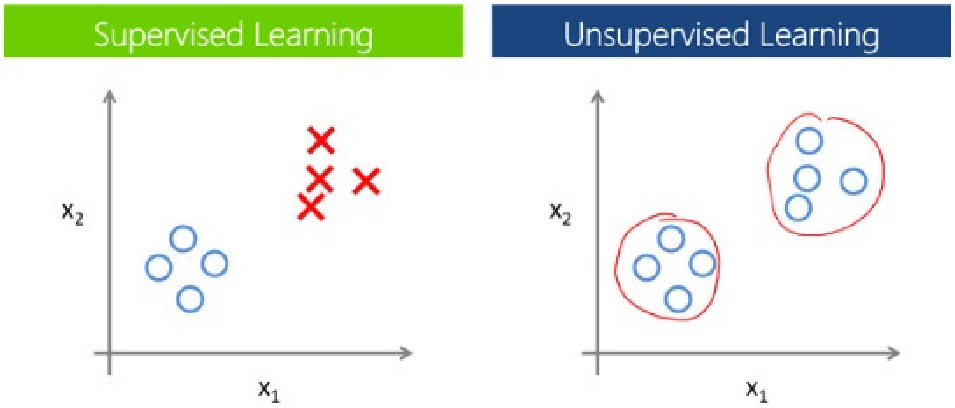


Figure 2.5: The figure on the left illustrates supervised learning where natural groupings of data is sought and the figure on the right illustrates unsupervised learning where these natural grouping are mapped.

distinct subsets that have variable survival rates, patient age and other clinical attributes based on each patient's molecular features [11]. These kinds of problems can be categorized as *clustering* problems where groups of instances or examples that belong together are sought. We use the *k-means clustering* algorithm to ascertain similar clusters based on each patient's molecular profile and identify those clusters which are relevant for prognosis. Once the clusters are deemed clinically relevant, we are then able to label patients based on the clustering and run supervised ML algorithms to ascertains features that distinguish the clusters.

2. Supervised learning differs from unsupervised learning in that it finds patterns and mappings of a data set based on user-specified or predetermined labeling of a training data set. Supervised learning is a form of *classification* learning where the learning scheme is presented with a set of classified examples or a training set from which it is expected to learn a way of classifying unclassified data. Supervised ML has two steps: (i) Training (ii) Prediction. The training step uses probabilistic models to create decision models or functions that best mirror the mappings specified by the training set. The prediction step uses the derived model and applies it to the dataset to calculate the effectiveness of the model.

The k-means algorithm provides labels to each instance of the data based on molecular similarity. This labeled set of data will then become the training set for the classification investigation. We utilize the *J48* data mining algorithm, a Weka based java implementation of the *C4.5* algorithm [43, 56], to create decision trees that allow classification of patients based on their molecular profile. The decision tree provides rules that enable us to map data, without any labels, to a prognostically relevant cluster.

Chapter 3: **Leveraging Patient Similarity for Clustering**

3.1 Finding Patient Clusters Based on Profile-Similarity

3.1.1 Patient Data

There has been a concerted effort to profile cancers based on their molecular makeup. The Cancer Genome Atlas (TCGA) was one of the largest-scale efforts that aimed to generate, analyze, and interpret molecular profiles at the DNA, RNA, protein, and epigenetic levels [53]. This effort has led to the creation of a data set that allows for the comparison and contrast of multiple tumor types. The data set includes molecular and clinical data from more than twelve different tumor types; Glioblastoma, first, and later Lower Grade Glioma and Glioblastoma, were also profiled as part of TCGA.

Previous work using the Cancer Genome Atlas data have incorporated multiple dimensions of the TCGA omic characterizations [9]. In our approach we work with four of the seven TCGA omic characterizations in Figure 3.1: Mutations, Copy Number, DNA methylation and mRNA expression. Clinical data is also added to the database but not used for clustering analysis. By combining these different characterizations or patient views, we hope to get a more comprehensive definition

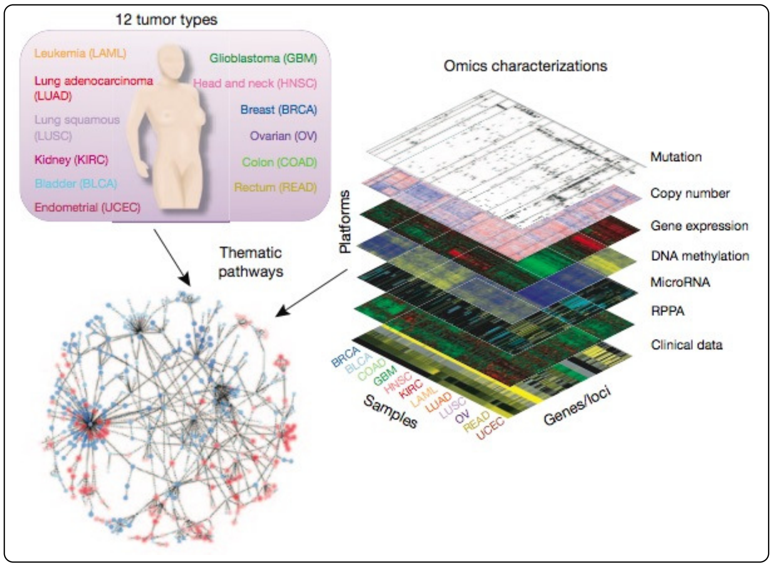


Figure 3.1: The Cancer Genome Atlas Integrated Data set. (Source: Weinstein, et al. [53])

of patient similarity for clustering analysis.

The sample set derived from TCGA are all patients diagnosed with brain tumors; specifically Glioblastoma and Lower Grade Glioma. The total sample set consists of 1,019 patients, each with four different patient views: Mutation, Copy number, mRNA expression and DNA methylation. Each patient view consists of approximately 11,000 genes. So in total, there are approximately 44,000 units of data for each patient sample. The values for each gene in each patient view are normalized to 0 (low), 1 (intermediate) or 2 (high) for Copy Number, DNA Methylation and mRNA Expression and simply 0 (no mutation) or 1 (mutated) for mutations.

Patient Views. The four different patient views are defined as follows:

Mutations Nucleotide alterations in a gene where a single or multiple nucleotide base pair(s) are altered due to DNA copying errors. Mutations may cause changes in protein structure and expression. Values are binary with 0 meaning no mutation and 1 meaning mutation.

Copy Number Repetition or deletion of long sequences of nucleotides. These often arise from incorrect repair of DNA damage and may result in aberrant protein expression. For each gene, values are normalized across the cohort to 0: low copy number, 1: intermediate copy number, and 2: high copy number.

DNA Methylation Process by which methyl groups, composed of carbon and hydrogen molecules, are added to the DNA molecule. This process affects how much of the DNA is active without changing the sequence. For each gene, values are normalized across the cohort to 0: low degree of methylation (also known as hypomethylation), 1: intermediate degree methylation, and 2: high degree of methylation (also known as hypermethylation).

mRNA Expression Measurement of how much mRNA is produced from particular genes. For each gene, values are normalized across the cohort to 0: low level of expression, 1: intermediate level of expression, and 2: high level of expression.

TCGA clinical data for each of the samples was used to assess prognostic significance of the identified clusters. Once clustering analysis using the molecular patient views was conducted, overall survival information was used to calculate Kaplan-Meier survival. The Kaplan-Meier survival assesses whether the clusters identified using their molecular characteristics, contained prognostic value.

Patient Clinical Data. The patient clinical attributes are defined as follows:

Sample ID A TCGA ID unique to each sample.

Sex The sex of the patient.

Race The race of the patient.

age The age of the patient.

AgeQ2 The normalized age of the sample to the whole sample set.

Time The survival time of the patient measured in days.

TimeQ2 The normalized survival time of the patient to the whole sample set.

Status The status of the patient, with 1 being alive at last follow up or study end and 0 being deceased.

It is to be noted that for methylation and mutation studies, a significant portion of the GBM patient molecular data is unavailable. This is due to the fact that the TCGA program spanned over a decade with GBM being one of the pilot studies. With advances in profiling technologies, studies done after GBM, such as LGG, were conducted using newer technologies and streamlined methods leading to better data availability. More sequencing data is available under protected availability but not used for this thesis.

3.1.2 Calculating Inter-Patient Similarity

With high dimensional patient data that contains thousands, if not, millions of data points, brute-force statistical and machine learning methods are inefficient to find patient groups. Hence, the Jaccard index was calculated to discover patient groupings based on inter-patient similarity. The Jaccard index measures similarity between any number of non-empty finite sets and is defined as the intersection over the union. Let A and B be two non-empty finite sets, then the Jaccard index is defined as:

$$J(A,B) = \frac{|A \wedge B|}{|A \cup B|} = \frac{|A \wedge B|}{|A| + |B| - |A \cup B|} \tag{3.1}$$

MATLAB was used to calculate the Jaccard index between each sample. Each patient view was downloaded as a matrix from a text file. Data for each patient view contained integers for genes with available data and the string 'NA' for genes with no data. The following algorithm was used to generate the intersection and the union for each patient view:

```
for Number of Patient Samples do
    for Number of Genes do
        if Both Patients have Integer data then
            increment the count of the union of the two patients by one;
        end
        if Both Patients have identical Integer values then
            Increment the count of the intersection of the two patients by one;
        end
    end
end
Result: Two 1019x1019 matrix that contain the Union and the Intersection.
```

Algorithm 1: Algorithm to calculate the Jaccard Index for each Patient View

A total of eight 1019x1019 matrices were generated; four for the intersection and four for the union of each patient view. For each of the patient views, the respective intersection matrix was divided by the union matrix to calculate the jaccard index matrix. The jaccard index matrix for each of the four patient views were then averaged together to create a correlation matrix that contained the jaccard index of how similar a sample is to another. The final result is a 1019x1019 matrix which contains the Jaccard index for each patient sample to the rest of the samples. Each value in the matrix corresponds to the Jaccard index between the patient represented by the row number and the patient represented by the column number.

3.1.3 k-means Clustering Algorithm

Clustering techniques are applied when there is no class to be predicted but the instances can be assigned or partitioned into natural groups or clusters. A cluster is defined as an aggregation of data points or vectors based on similarity. The *k-means* algorithm is an unsupervised clustering algorithm that partitions data into k mutually exclusive clusters. The algorithm iteratively tries to assign each data vector to a cluster based on the features provided. The goal of the algorithm is to create the specified k centroids and reduce the sum of squares of all the data points assigned to the centroid. Formally, the algorithm aims to minimize the following objective function [28]:

$$K = \sum_{j=1}^{k} \sum_{i=1}^{m} ||x_i^j - c_j||^2 \tag{3.2}$$

where $||x_i^j - c_j||^2$ is the euclidean distance between a data vector x_i^j and a centroid, c_j. k is the number of clusters specified and m is the total number of data points. Note that x_i^m can be an n-dimensional vector.

The algorithm iteratively minimizes the sum of squares for each of the clusters. It begins with assigning k centroids and iteratively moves the centroids until the best minimization is found. Figure 3.2 is a graphical representation of each iteration of the algorithm in 2-D space. It is composed of the following steps:

1. Place k random points in the n-dimensional space of all the data points as far away from other points. These are the initial centroids.

2. Assign each data point to the closest centroid. Closeness is calculated as the euclidean distance between the data point and the centroid.

3. When all data points have been assigned, take the average of the points in the cluster and recalculate the centroid position.

   $$c_j^{t+1} = \frac{1}{n_{cj}} \sum x_j \tag{3.3}$$

   c_j is the new centroid position, n_{cj} is the number of data points in the cluster and x_j are the data points in the cluster.

4. Repeat steps 2 and 3 iteratively until the centroids do not move.

K-means is easy to implement and is one of the best clustering algorithms to run on large data sets. Compared to other clusters, the computational cost of

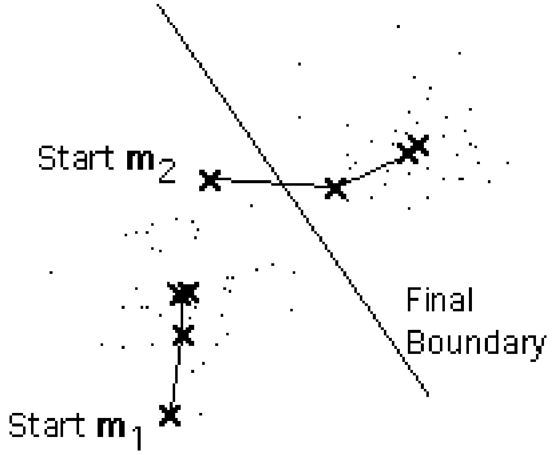


Figure 3.2: K-means clustering in 2-D space.

k-means is low; with a complexity of *O(K*m*n)*, where k is the number of clusters, m is the number of data points and n is the dimensions of the vectors. However, k-means can only be used on numeric data sets and perform better where the clusters are more spherical. Another caveat of the algorithm is specifying the number of clusters to partition the data into. It is not always evident what the optimal k value should be, however methods such as the elbow method and silhouette method can be utilized to narrow down to a range of k values. The algorithm is a built-in function in MATLAB, and requires as inputs the data matrix and the number of clusters k, and outputs the cluster ID of each sample and the centroid locations.

The elbow method was used to determine the minimum k value for the data set. The method works by running the k-means algorithm from 1 to k times iteratively on the data set, and measuring the total sum of squares of all the data from each cluster. The initial few clusters will have a high sum of squares value but, as the number of clusters increase the total sum of squares will drop precipitously. As the

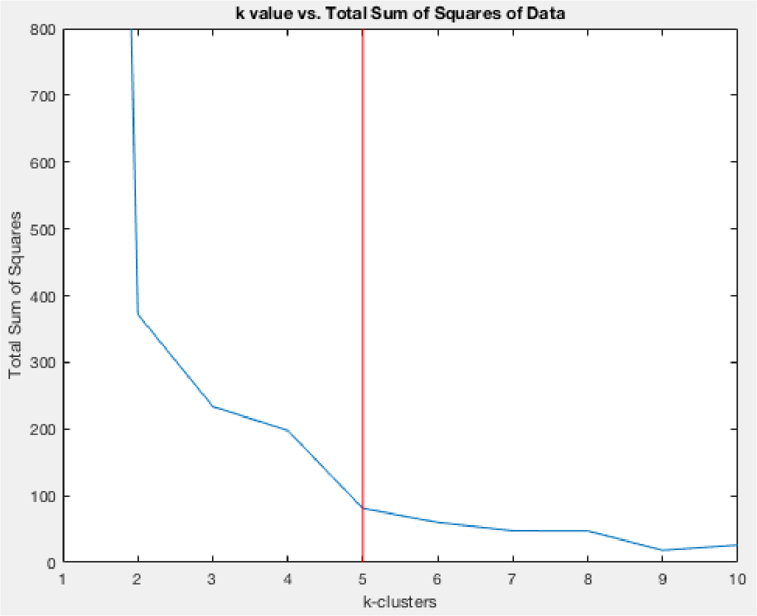


Figure 3.3: The Total sum of squares vs. the number of clusters identify a k value of 5 as the minimum value when evaluating k values from 1 to 10.

process continues, there will come a point at which adding a new cluster will only decrease the total sum of squares marginally, at which point, an angle will develop in the plot of the number of clusters versus the total sum of squares. This point denotes the minimum k value for the data set.

Previous work using the TCGA data sets selected 7 clusters of Glioblastoma and Lower Grade Glioma diagnosed patients [11]. For our data set, we explored k values ranging from 2 to 10 clusters. As depicted in Figure 3.3, the elbow is created at a k value of 5. The result of the elbow method suggests that for the patient correlation matrix, a minimum k value of 5 is adequate to decrease the total sum of squares of the data. Any k value higher than 5 will give us clusters with relatively low sum of squares, whereas any value lower than 5 will have a high total sum of squares. The results of the clustering is presented in Section 3.2.

3.2 Clustering Results

3.2.1 Validating Results

Based on the results of the elbow method detailed in section 3.1.3, we ran the k-means method with k values from 5 to 9. This produced five different data sets with the patient samples clustered into 5, 6, 7, 8 and 9 clusters respectively. For each data set, we performed the Kaplan-Meier overall survival analysis [25] and the Log Rank test [30] to determine if overall survival of patients from different clusters were statistically different. That is, we wanted to identify if the partitioned clusters validly captured a cohort of the population with a different survival rate. A statistical significance between the clusters would indicate a difference in the prognosis of the patients in the cluster compared to other patients.

The Kaplan-Meier Test estimates the probability of survival at each point in time [25]. It is used to compute a population survival function from the survival time and status (alive or dead) for patients. Survival time is measured as the unit of time a patient is alive after diagnosis and is updated each time a patient comes in for a follow-up. Kaplan-Meier also takes into account censored data where the patient data is missing after a period due to the patient withdrawing from the study or from not having a follow-up. The probability of surviving at any time t_i is calculated as the product of the probability of surviving at each of the times before t_i, i.e. from t_1 to t_{i-1}. The Kaplan-Meier function is defined below:

$$S(t) = \prod_{i=t_1}^{t_n} \frac{a_i - d_i}{a_i} \tag{3.4}$$

where $S(t)$ is the probability of surviving longer than time t, t_i is the time interval, t_n is the maximum survival time in the data, a_i is the number of patients known to have survived at time t_i and d_i is the number of patients who have died at t_i.

The Log Rank test is a hypothesis test that compares the survival curves of two independent groups or clusters [30]. The test can be approximately distributed as a *Chi-Squared* statistic. The test is used to identify if two clusters are statistically different from one another. Essentially the test compares the observed and the expected survival rates for two clusters. The **null hypothesis** for the test states that two independent groups or clusters have equal survival rates. The degree of freedom is calculated as the number of outcomes (dead or alive) minus one which in this case is 1. The Chi-Squared value is computed as follows:

$$X^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{3.5}$$

Where n is the total number of clusters being compared, O_i is the observed number of deaths in each cluster and E_i is the expected number of deaths in each cluster. The Log Rank test can be done pair-wise for each cluster in the group to identify if each cluster is statistically significant from the other clusters. When performing a pair-wise cluster comparison, the expected number of deaths in each cluster can be calculated as follows:

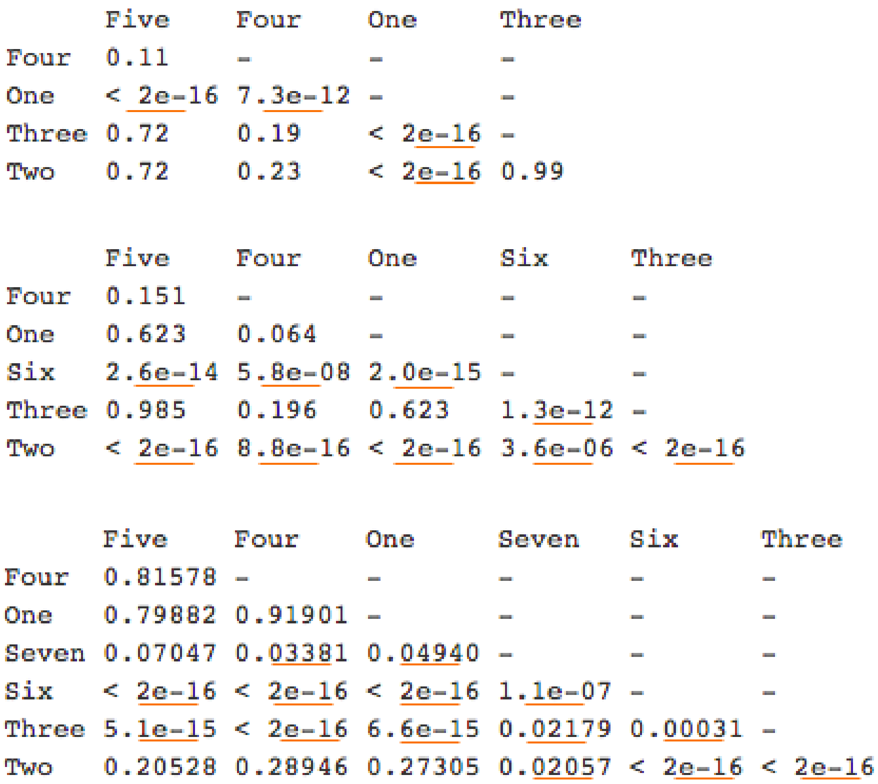
$$E_1 = \sum_{i=t_1}^{t_n} \frac{N_1 \cdot O_{1,2}}{N_{1,2}}, \quad E_2 = \sum_{i=t_1}^{t_n} \frac{N_2 \cdot O_{1,2}}{N_{1,2}} \tag{3.6}$$

where, at time t, N_1 and N_2 are the number of survivors in cluster 1 and 2 respectively, $O_{1,2}$ is the total number of deaths observed in both clusters and $N_{1,2}$ is the total number of survivors in both clusters. The Chi-Squared value can then be used to determine the *p-value*. A p-value less than 0.05 corresponds to rejecting the null hypothesis; meaning there is statistical significance between the survival rates of the two clusters. The Kaplan-Meier and Log Rank tests were jointly conducted using the *survdiff* function in the R statistical software [55].

3.2.2 Clustering Results for Multiple k Values

Pair-wise Log Rank tests were conducted for k values ranging from 5 to 9. The results of each clustering are presented below.

Figure 3.4 presents the Log Rank test output for k values from 5 to 7. For $k = 5$, only cluster *One* is statistically different from other clusters with a p-value less than 0.05. The remaining clusters have higher p-values, indicating that the overall survival of the clusters are not statistically different. Of the total 9 pair-wise comparisons, only 4 were less than 0.05. For $k = 6$, clusters *Two* and *Six* are statistically different from the rest. Even with 60% of pair-wise comparisons, 9 out of 15, having a p-value higher than 0.05, only two clusters out of the 6 are distinct from others. Finally, for $k = 7$, clusters *Three*, *Six* and *Seven* have majority of pair-wise p-values less than 0.05 in the pair-wise comparisons. However, the majority

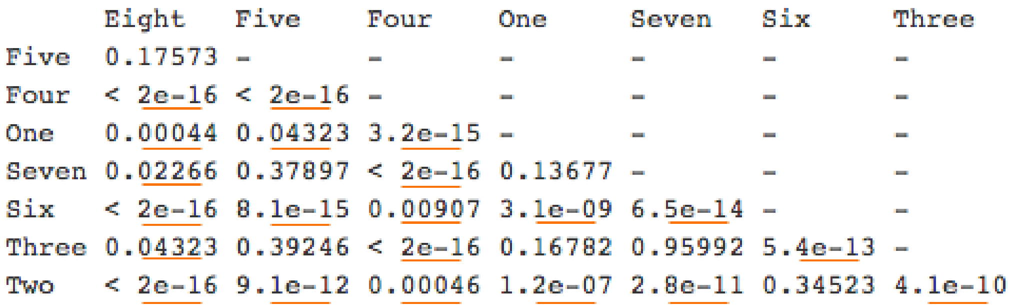


```
      Five     Four     One      Three
Four  0.11     -        -        -
One   < 2e-16 7.3e-12 -        -
Three 0.72     0.19     < 2e-16 -
Two   0.72     0.23     < 2e-16 0.99

      Five     Four     One      Six      Three
Four  0.151    -        -        -        -
One   0.623    0.064    -        -        -
Six   2.6e-14 5.8e-08 2.0e-15 -        -
Three 0.985    0.196    0.623    1.3e-12 -
Two   < 2e-16 8.8e-16 < 2e-16 3.6e-06 < 2e-16

       Five    Four    One     Seven   Six     Three
Four   0.81578 -       -       -       -       -
One    0.79882 0.91901 -       -       -       -
Seven  0.07047 0.03381 0.04940 -       -       -
Six    < 2e-16 < 2e-16 < 2e-16 1.1e-07 -       -
Three  5.1e-15 < 2e-16 6.6e-15 0.02179 0.00031 -
Two    0.20528 0.28946 0.27305 0.02057 < 2e-16 < 2e-16
```

Figure 3.4: Log Rank pair-wise p-values for k values of 5 (top), 6 (middle) and 7 (bottom). P-values less than 0.05 are underlined in red.

of pair-wise comparisons for clusters *Five*, *Four*, *One* and *Two* have high p-values indicating that these clusters are not well differentiated from the rest. The results suggests that a clustering of the patient correlation data into 5, 6 or 7 clusters is not adequate to produce distinct clusters with differing prognosis.

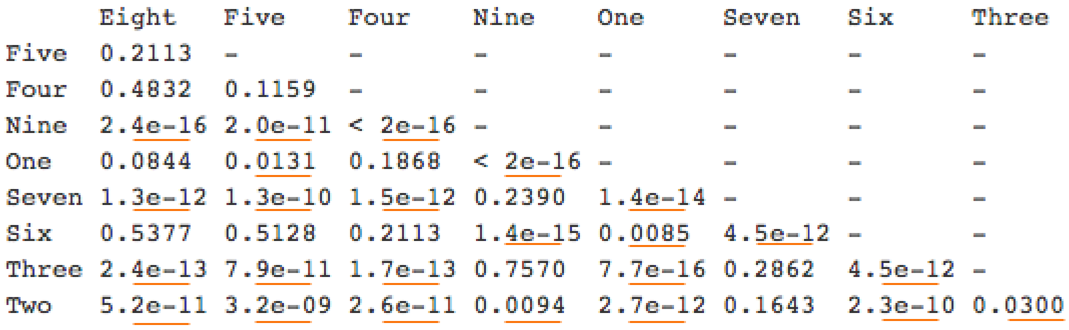
Clustering of patient data into 8 clusters produced clusters where 75% of pair-wise p-values, 21 out of 28, were lower than 0.05. Figure 3.5, shows the results of the Log Rank pair-wise test for 8 clusters. The test indicates that cluster *Four* is statistically different from all other clusters, clusters *Two* and *Six* are statistically different from 6 out of the 7 clusters, clusters *One* and *Eight* from 5 out of 7 clusters and finally clusters *Three*, *Five* and *Seven* from 4 out of 7 clusters. The



```
      Eight    Five     Four     One      Seven    Six      Three
Five  0.17573 -        -        -        -        -        -
Four  < 2e-16 < 2e-16 -        -        -        -        -
One   0.00044 0.04323 3.2e-15 -        -        -        -
Seven 0.02266 0.37897 < 2e-16 0.13677 -        -        -
Six   < 2e-16 8.1e-15 0.00907 3.1e-09 6.5e-14 -        -
Three 0.04323 0.39246 < 2e-16 0.16782 0.95992 5.4e-13 -
Two   < 2e-16 9.1e-12 0.00046 1.2e-07 2.8e-11 0.34523 4.1e-10
```

Figure 3.5: Log Rank pair-wise p-values for patient correlation data clustered into 8 clusters. p-values less than 0.05 are underlined in red.

results indicate that all clusters are statistically different to at least the majority of clusters.



```
      Eight   Five    Four    Nine    One     Seven   Six     Three
Five  0.2113  -       -       -       -       -       -       -
Four  0.4832  0.1159  -       -       -       -       -       -
Nine  2.4e-16 2.0e-11 < 2e-16 -       -       -       -       -
One   0.0844  0.0131  0.1868  < 2e-16 -       -       -       -
Seven 1.3e-12 1.3e-10 1.5e-12 0.2390  1.4e-14 -       -       -
Six   0.5377  0.5128  0.2113  1.4e-15 0.0085  4.5e-12 -       -
Three 2.4e-13 7.9e-11 1.7e-13 0.7570  7.7e-16 0.2862  4.5e-12 -
Two   5.2e-11 3.2e-09 2.6e-11 0.0094  2.7e-12 0.1643  2.3e-10 0.0300
```

Figure 3.6: Log Rank pair-wise p-values for patient correlation data clustered into 9 clusters. p-values less than 0.05 are underlined in red.

Clustering of patient data into 9 clusters produced only 66% of pair-wise results, 24 out of 36, to have p-values less than 0.05. This indicates that the clusters are less statistically significant when compared to the partitioning of data into 8 clusters. Cluster *Two* is statistically significant from 7 out of 8 clusters, clusters *One*, *Three* and *Nine* from 6 out of 8, clusters *Five*, *Six* and *Seven* from 5 out of 8 and cluster *Four* from 4 out of 8. It is interesting to note that although the total sum of squares decreased, the clustering of patients into 9 clusters did not produce more significant results when compared to 8 clusters.

Clustering of patient correlation data into 8 clusters captures enough patient cohorts with variable survival times when compared to the clustering of data into 5, 6, 7 or 9 clusters. As mentioned above, the 8 clusters had the highest percent of pair-wise p-values less than 0.05. Previous studies using the TCGA data set produced 7 clusters [11]; of which 3 out of 7 were majority IDH mutants with lower grade gliomas and 4 out of 7 were primarily IDH wild type with majority glioblastomas. Similarly, in our clustering results, 3 out of the 8 clusters, clusters *Two*, *Four* and *Six*, were primarily IDH mutant with majority lower grade gliomas. Likewise, 4 out of 8 clusters, clusters *Three*, *Five*, *Seven* and *Eight* were IDH wild type with majority glioblastomas. Only cluster *One*, which is composed of almost an equal ratio of IDH wild type and mutants, was not identified in literature. Also of note, cluster *Four* from the 8 cluster result was split into two clusters, cluster *Two* and *Seven*, for the 9 cluster result. A comparison of the pair-wise p-value in Figure 3.6, show that clusters *Two* and *Seven* are not statistically different in the 9 cluster result whereas cluster *Four* can be differentiated from all clusters in the 8 cluster results. In conclusion, clustering of data into 8 clusters was chosen as the best clustered result for the patient correlation data.

Now that a representative cluster is identified, the 8 clusters will be the input of the data mining investigation conducted in Section 4.3. The data mining procedure will allow us to identify molecular attributes that map each sample into its respective cluster. This in turn allows us to map new patient data into the 8 clusters without performing the k-means clustering procedure again. A discussion of each of the clusters from the 8 cluster results is found in Section 3.2.3.

3.2.3 Eight Clusters Result

Cluster IDs for each sample was outputted by the native k-means function in MATLAB. The cluster IDs along with clinical data and relevant molecular attributes detailed in previous TCGA studies [11] were ingested into Neo4j. Neo4j is a graph database platform used for the querying and visualization of the cluster data [32]. For a more detailed discussion on graph databases and Neo4j, refer to Section 3.3. The following figure, Figure 3.7, provides the block definition diagram of the TCGA patient Neo4j node with its property names and data types. Once ingested, relevant properties of a cluster can be queried using Neo4j's native Cypher query language [19].

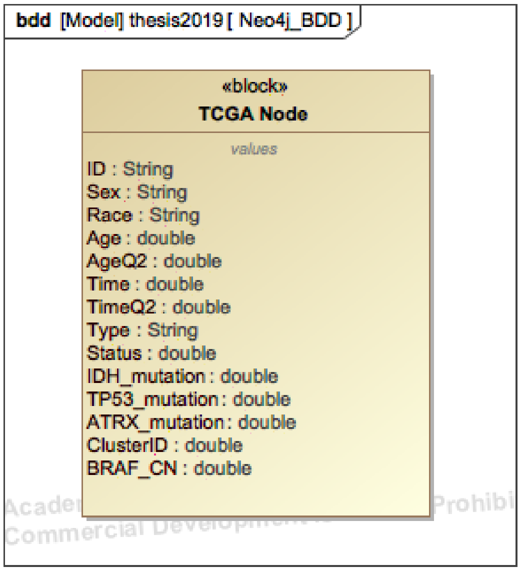


Figure 3.7: SysML Block Definition Diagram of the TCGA patient Node in Neo4j.

Table 3.1 provides the total number of samples, the average age of patient, the average survival time measured in days, the IDH mutation count and the type of Gliomas diagnosed for each cluster. Glioma Type has two attributes for the diagnosis

Cluster ID	**Num. of Samples**	**Avg. Age**	**Avg. Time**	**Glioma Type**
1	130	52.7	585.5	LGG: 35 GBM: 95
2	119	40.6	840.0	LGG: 94 GBM: 25
3	134	53.1	557.5	LGG: 46 GBM: 88
4	122	43.7	781.2	LGG: 122 GBM: 0
5	139	55.7	437.8	LGG: 36 GBM: 103
6	125	40.4	717.2	LGG: 116 GBM: 9
7	129	57.3	472.5	LGG: 55 GBM: 74
8	121	56.8	474.2	LGG: 7 GBM: 114

Table 3.1: Relevant clinical attributes of each of the 8 clusters.

Cluster ID	**IDH Mut.**	**TP53 Mut.**	**ATRX Mut.**	**Avg. EGFR Amp.**
1	1: 48 0: 61	1: 24 0: 45	1: 9 0: 60	-0.430
2	1: 90 0: 19	1: 81 0: 17	1: 63 0: 35	0.166
3	1: 29 0: 85	1: 45 0: 29	1: 22 0: 52	-0.027
4	1: 122 0: 0	1: 5 0 : 116	1: 2 0: 119	-0.589
5	1: 28 0: 94	1: 6 0: 59	1: 3 0: 62	0.147
6	1: 112 0: 10	1: 99 0: 16	1: 70 0: 44	0.122
7	1: 4 0: 106	1: 6 0: 68	1: 2 0: 72	1.13
8	1: 15 0: 81	1: 7 0: 27	1: 2 0: 32	-0.115

Table 3.2: Relevant molecular attributes of each of the 8 clusters.

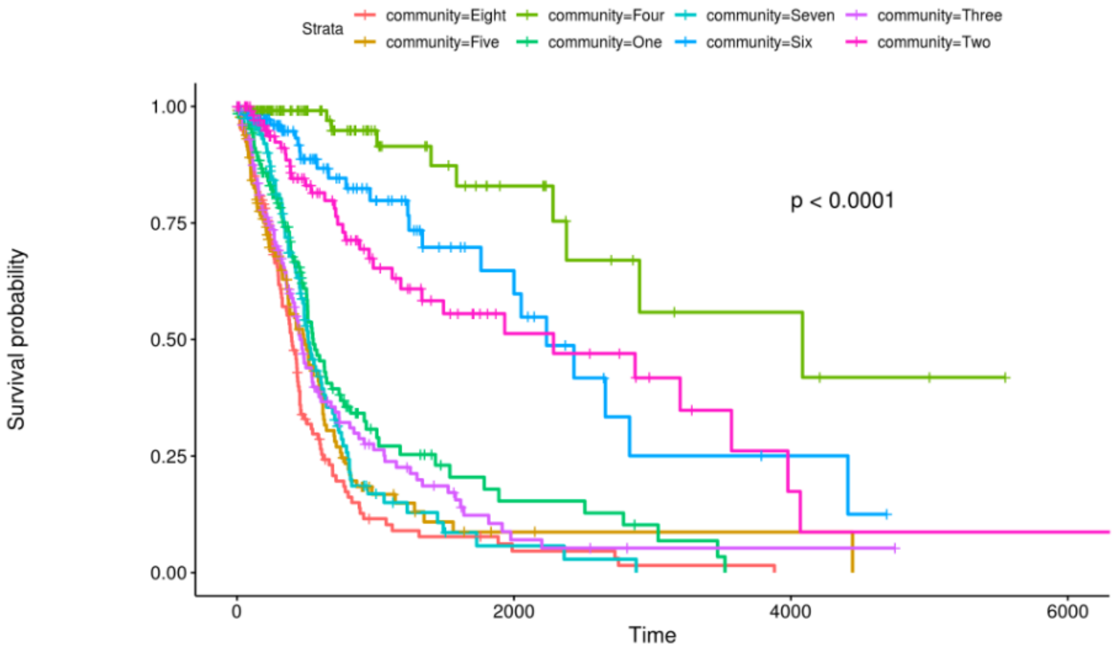


Figure 3.8: Kaplan-Meier survival curves for each cluster.

of the disease; LGG is *Lower Grade Glioma* and GBM is *Glioblastoma*. Table 3.2 provides some of the relevant molecular attributes identified in literature [11] and in Figure 1.4 for each of the clusters. All mutation data has two attributes: 1 signifies a mutated gene and 0 a wild-type gene. The average of the unnormalized EGFR copy number data is also provided to identify any amplification in the gene. Figure 3.8, provides the Kaplan-Meier survival curves for the eight clusters.

Although the clusters do not partition exactly compared to previous studies [9, 11], there are identifiable molecular attributes in the clusters which validate the respective clinical findings. For example, IDH mutations in gliomas are characterized as having an early age of diagnosis with longer survival rates than IDH wild-type gliomas [11]. Most brain tumors with IDH mutations are also diagnosed as Lower Grade Gliomas. These findings are reflected in clusters Two, *Four* and *Six* which all have majority IDH mutations, and the patients have relatively younger age at diagnosis and relatively long survival time and probability (slowly sloping survival curves). The majority of the patients in these clusters are diagnosed as Lower Grade Gliomas as well. On the contrary, patients harboring IDH wild-type tumors have older age at diagnosis and poor prognosis with short survival times (steep survival curves). These tumors are often diagnosed as Glioblastomas. Clusters *Three*, *Five*, *Seven* and *Eight* reflect the clinical attributes found in IDH wild type tumors.

A closer look at the clustering also reveals clusters that potentially coincide with clusters identified by Caccarelli, et al. [11]. Clusters *Two* and *Six* are both characterized with a majority of IDH, TP53 and ATRX mutations. These molecular attributes are also reflected in the G-CIMP-low and G-CIMP-high clusters in Figure 1.4. All IDH wild type clusters also have relatively high EGFR amplification values, indicating that these clusters form the Classic-like, Mesenchymal-like, GBM and PA-like partitions represented in Figure 1.4. With clinical findings validated by select molecular attributes, the 8 cluster result provides a usable training set for the data mining investigation described in Chapter 4.

3.3 Querying Results using Graph Databases

3.3.1 Graph Databases

Traditional databases follow the relational paradigm where data is grouped into tuples and relations. This model organizes data into tables or relations where each tuple or row is a data object or entity with columns defining the attributes of the entity. Graph databases deviate from this traditional model; instead of rows and columns, graph databases use nodes and edges to represent and store properties and relations of data. Graph databases provide flexibility in defining data entities and stress the relation between entities of data.

Graph databases are a subset of the NoSQL database paradigm which aims to address limitations of relational databases (RDBMS). NoSQL stands for "Not Only SQL," which emphasizes performance and scalability than rigidity of data. This paradigm is a result of rapid growth in web services such as social networks which have flexible and non-rigid data entities that are highly connected. This allows the insertion of data entities that may not have the same attributes as others into the database without losing application functionality. This flexibility means that the database designer does not have to excessively design and plan the database before ingesting data.

Graph databases as the name implies, are based on graph theory and are comprised of a set of nodes, edges and properties.

Nodes Represent entities or data objects in the database such as a person or a social network account. Nodes are analogous to rows in RDBMS.

Edges Represent relationship between nodes. Edges may be directed or undirected in graph databases. Edges differentiate graph databases from RDBMS, where relationships are not explicitly represented.

Properties Represent the attributes of each data entity. These are analogous to columns in RDBMS. Edges may also contain properties in a graph database.

The true advantage of graph networks is in the scalability and flexibility of the data that it can contain. They are exceptional in handling search and queries when working with large volumes of data and in areas where data topology or connectivity is important. Examples of graph database employed successfully include Google's Knowledge graph [52], Twitter's FlockDB, and many more. Applications in biology, chemistry, and the semantic web are examples of fields that can more naturally be represented by graph databases.

Graph databases are slowly becoming more common in bioninformatics solutions. They are helpful in storing genome, protein and other views of the human body while enabling creation of the web of connections between these different views. When working with human genome data, RDBMS are adept at representing the genome attributes of single individuals. However, graph databases allow the storage, search and queries of data as well as the relationship between individual entities in an efficient and speedy manner.

3.3.2 Neo4j Graph Platform

Neo4J is the leading open-source graph database platform available at the moment. It is touted to be able to handle billions of records including nodes and edges while maintaining search and query support. Neo4j has been used as the primary graph database solution to large-scale data projects. For example, Neo4j powers Bio4j which is an aggregated knowledge base for protein related information accrued from disparate sources such as Gene Ontology, NCBI Taxonomy etc [40].

Data storage in Neo4j is dissimilar to relational database management systems (RDBMS). Whereas RDBMS use pre-defined tables with often disjointed relationship between entities, Neo4j leverages a graph storage structure. Data modeling is much more flexible and easier as different data entities can be added or changed at any time. RDBMS on the other hand require upfront development of the logical model and data types sources need to be known ahead of time. Querying in Neo4j is much faster than in RDBMS where relationship between entities are not explicitly defined. The graph structure of Neo4j allows natural querying regardless of the number of relationships. Neo4j is a highly reliable, available and fault-tolerant platform that complies with modern RDBMS database standards. Much like oracle databases, Neo4j is fully ACID compliant [32]; this means all transaction with the database are processed reliably and validly even in the event of failures or user errors.

ACID Compliance. Acid compliance is defined as having the following attributes:

Atomicity If a transaction fails, the database will be unaffected.

Consistency Ensures any and all transactions can change data only in predefined or valid ways.

Isolation Any data being modified during a transaction cannot be accessed until the transaction is complete.

Durability All complete transactions will be reflected in the database.

Neo4j is now widely accepted as the standard in graph database enterprise solution.

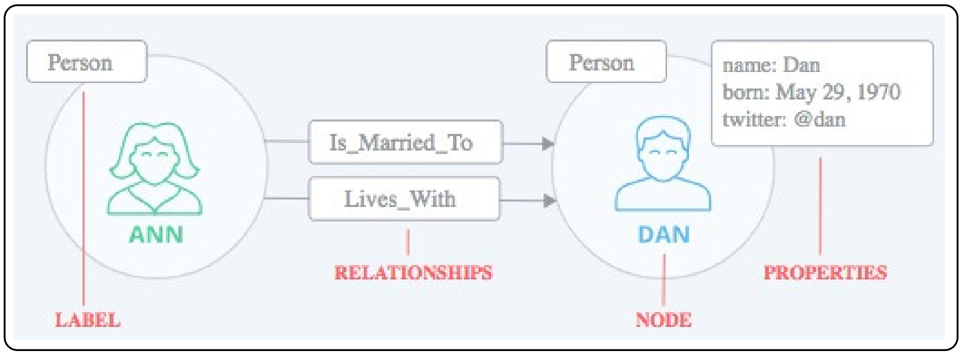


Figure 3.9: The label property graph model of Neo4j.

As mentioned in Section 3.3, Neo4j employs Nodes to store data attributes and properties, and edges to store relationships. Nodes are connected by edges and may have one or more labels-groupings of nodes into sets. Labels and properties are indexed for optimized querying. Figure 3.9, shows a simple schematic of the structure of Neo4j. The two nodes represent a person, which is identified by the node label. The two nodes are connected by two relationships which can be directed or undirected and finally, the nodes contain attributes specific to each node.

Neo4j can be embedded in Java applications. For the purposes of this thesis, Neo4j Community Edition 3.4.5 was used with Neo4j dependency added via Apache Maven. Apache Maven is a build automation tool that dynamically downloads Java libraries from an online central repository. Java version 8 and 11 are supported by current versions of the Neo4j java driver. Neo4j also supports the Cypher Query Language which allows for efficient querying and updating of graphs [19]. Cypher offers the full range of features expected from a RDBMS Query Language optimized for graph databases.

From figure 3.9, it is easy to see how Neo4j's structure can be leveraged to not only store and query data but employ graph algorithms to garner more insights into the data. Neo4j offers a suite of algorithms used to compute metrics of graphs of nodes and relationships. Neo4j offers centrality algorithms which help in determining the importance of nodes in a network and path finding algorithms that allow to evaluate the availability and length of routes. Neo4j also offers community detection algorithms that evaluate if there are clusters or partitions of nodes. With the integration of data from multiple domains, Neo4j is an ideal tool to visualize and query networks of biological data from a database storage and query perspective. The results of the clustering performed in this thesis can be downloaded as a Neo4j database upon request.

3.4 Discussion

This chapter provided an overview of the patient data, the clustering analysis and the results of the clustering analysis. With high-dimensional data, we employed the Jaccard index to create a correlation matrix capturing how similar a patient is to another based on four patient views (Mutations, Copy Number, DNA methylation and mRNA expression). A clustering analysis was conducted using the k-means algorithm to then partition the patients based on molecular similarity. The idea is that molecular similarity will lend itself to similar prognosis. The clustering results were then validated using the Kaplan-Meier and Log Rank tests and corroborated with literature. The results are then uploaded to the Neo4j graph database for querying and visualization.

As sequencing technologies advance, patient data will become more precise and abundant. However, transforming the raw data for functional use requires new frameworks and algorithms. We provide a straightforward and effective approach by using the k-means algorithm to transform independent numerical patient data into cohorts with similar characteristics. This is in no way a novel approach, however, this approach combined with classification and semantic modeling, discussed in the subsequent chapters, create a framework for effectively transforming the patient data into functional models.

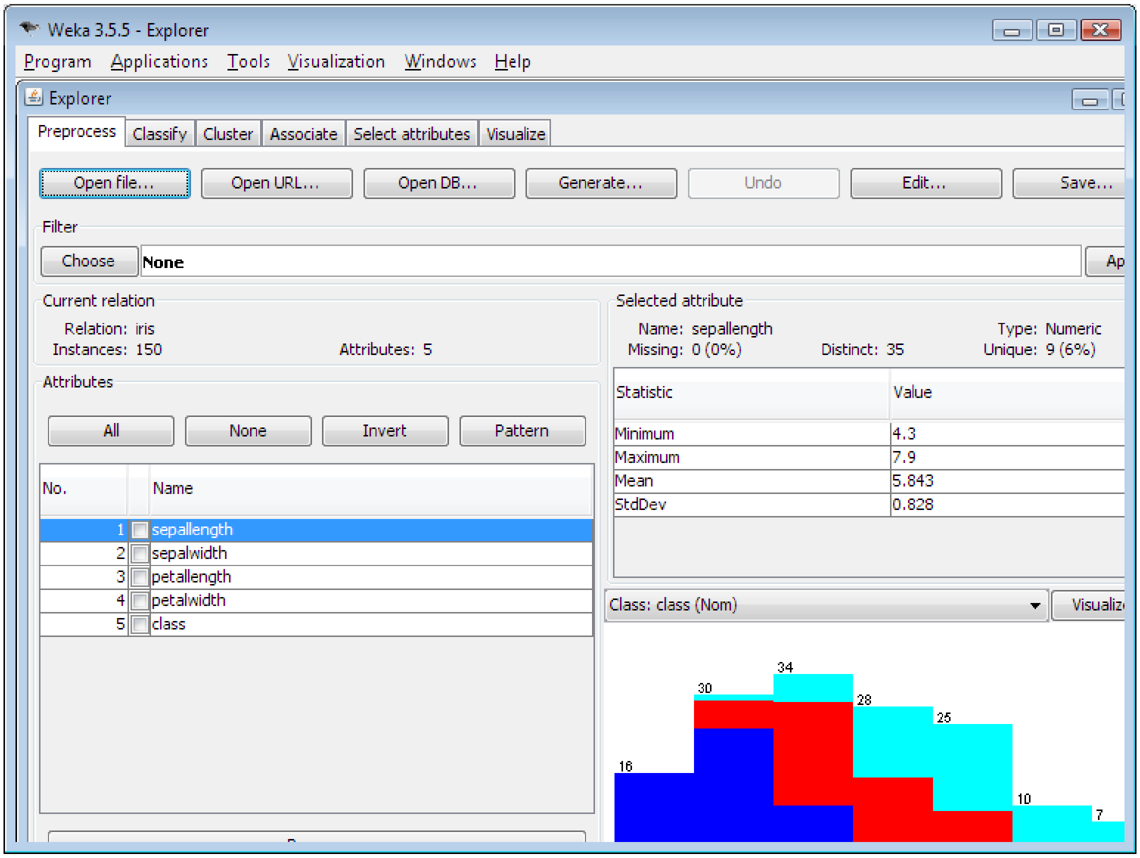
Chapter 4: **Decisions Supported by Data**

4.1 Weka: Open-Source Data Mining Application

4.1.1 Weka Explorer

Data mining helps in elucidating the hidden patterns and intelligently analyzing the vast sources of data that is being generated and stored nowadays. Data mining is defined as the process of discovering patterns in data in an automatic or semiautomatic way [56]. In data mining, the data will take the form of a set of examples – examples of sepal length of flowers. The output of data mining is the predictions on new examples based on the previous examples or data – the species to which a flower belongs to based on its sepal length. The Waikato Environment for Knowledge Analysis (Weka) is an open-source software developed at the University of Waikato, New Zealand that provides multiple algorithms and tools to analyze large data sets and provide functional insights into the data [56].

Weka provides implementations of data analysis and predictive modeling software that can be used via an interactive graphical user interface or via Java APIs (Application Programming Interfaces). Weka provides methods for most of the important data mining problems ranging from regression, classification, clustering,


Weka 3.5.5 - Explorer
Program Applications Tools Visualization Windows Help
Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Open file... Open URL... Open DB... Generate... Undo Edit... Save...
Filter
Choose
Ap
Current relation
Attributes
All None Invert Pattern
Selected attribute
Visualiz
34
30
28
25
16
10
7

Filter: None

Relation: iris
Instances: 150
Attributes: 5

No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class

Name: sepallength
Type: Numeric
Missing: 0 (0%)
Distinct: 35
Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

Class: class (Nom)

Figure 4.1: The Weka Explorer GUI Menu.

association rule mining and attribute selection. It also provides a workbench which allows for data pre-processing and data visualization. The easiest way to use Weka's algorithms and tools is via the Weka Explorer GUI, depicted in Figure 4.1. The Weka Explorer has six primary tabs which are summarized below.

- *Preprocess* Choose and modify dataset for investigation.
- *Classify* Train learning schemes and evaluate the schemes through classification or regression.
- *Cluster* Learn clusters for the dataset.
- *Associate* Learn and evaluate association rules for the dataset.

- *Select attributes* Select attributes of interest from dataset for learning scheme.

- *Visualize* View 2-D plots of data and other diagrams like decision trees.

Weka is written in Java and can be run on Linux, Windows and Macintosh operating systems. Weka version 3.8.3 was used as part of this thesis.

4.1.2 Instances, Attributes and ARFF format

The input to a Machine learning function is a set of instances or examples. The input data set is expressed as a set of independent instances each with its own set of predetermined numeric and/or non-numeric attributes. Weka broadly classifies all attributes as either *numeric* or *nominal* [56]. Numeric attributes measures numbers that are either real or integer valued. Nominal attributes are values that are non-numeric symbols and serve as labels or names. *Boolean* attributes are a special case of nominal attributes-often designated as *true* or *false*, however they can also be characterized by the standard integer convention.

The bulk of the time in a data mining investigation is consumed by input preparation and processing. Weka uses the attribute-relation file format (ARFF) for all input of data. The ARFF file is an ASCII text file that lists a set of instances that share a common list of attributes. Figure 4.2, depicts the contents of an ARFF file relating to weather data. The file is composed of two sections. The header section contains the name of the relation and a list of attribute definitions. Each attribute is defined by its name and its data type. The final attribute definition is the attribute to be classified. For example, in Figure 4.2, a learning scheme will be created to determine if one should play or not based on the weather data instances provided. The second section is the data section which contains instances of data with predetermined or observed values for the attributes.

```
%ARFF file for weather data with some numeric features

@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature numeric
@attribute humidity numeric
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
%14 instances

sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

Figure 4.2: Example of an ARFF file content.

Weka provides the ARFF-Viewer tool to view, edit and create ARFF files. The patient data was first saved as a comma separated file (csv) containing the relevant molecular information and the Cluster ID for each patient. It is to be noted that the first row of the csv files must contain the name of the attributes with each subsequent row being the instances, otherwise, Weka will not be able convert the file into an ARFF file. The csv file was then uploaded into ARFF-Viewer from which attributes can be selected or deselected as well as the attribute to be classified can be selected. The csv file can then be saved as an ARFF file by simply going to File and Save As.

4.2 Decisions made from Data

4.2.1 Decision Trees

Decision trees are decision support tools that enable deduction of a set of conclusions from a set of observations. Decision trees are used for visual and explicit representation in decision analysis. In Machine Learning, each node in a decision tree represents an attribute and each branch is the outcome of the conditional statement on the attribute. Decision trees can be viewed as the output of the mathematical and computational algorithms used to categorize and generalize a dataset. Decision trees are used ubiquitously throughout the field of medicine, especially in the classification of diseases based on the structure of affected tissue and molecular attributes. Although not perfect, decision trees provide an easy to understand representation and heuristic to complex problems and questions.

Rules in data mining can be characterized as if-then patterns found within the data. Classification rules enable the mapping of an instance or an example to a class or a category. The *precondition* of a rule is the series of branches that is traversed to reach the *consequent*, which is the conclusion or the class that apply to the instances covered by the rule. It is easy to generate a set of classification rules

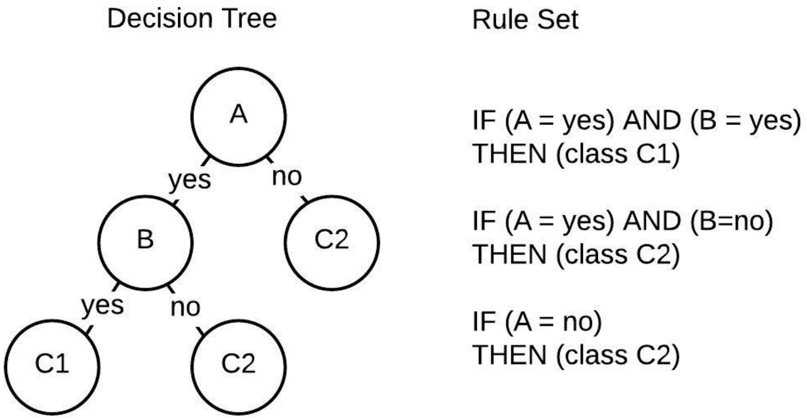


Figure 4.3: Creation of rules from a decision tree (Source: Freitas, et al. [20]).

directly off a decision tree. A simple pass through the branches of the decision tree from the root to the leaf is enough to generate a rule. This procedure is illustrated in Figure 4.3.

The C4.5 algorithm summarized in Section 4.2.2, produces a decision tree based on the molecular attributes of each cluster. The consequent of the tree is the cluster the instance in question belongs to. Each of the node in the tree is a molecular attribute and each of the branches from the node is the path of a conditional statement on the node. A traversal from the root to each of the leaves of the tree will produce unique rules that allow for the classification of instances into the clusters. The classification rules are extracted form the C4.5 generated tree using the PART rule generator in Weka [56].

4.2.2 *C4.5*: Divide and Conquer

The C4.5 algorithm is a supervised learning algorithm that allows for the classification of new instances from a training set. The algorithm uses a univariate decision tree approach where splitting of the decision tree is based on a single attribute at each node. The algorithm utilizes the concept of 'entropy,' which is the measure of disorder of the data [8]. Weka implements a java version of C4.5 called *J48*.

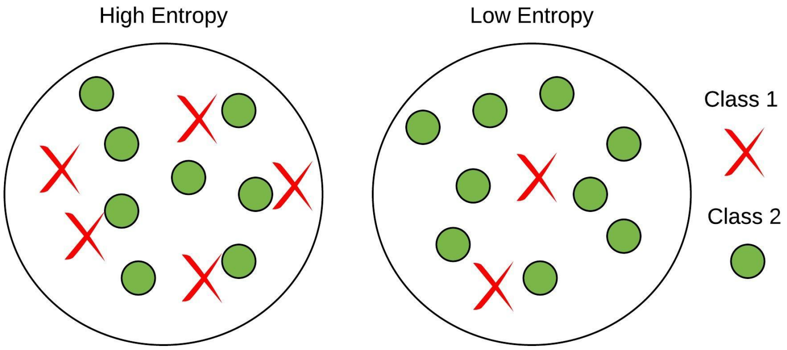


Figure 4.4: Entropy is a measure of how 'impure' the data is.

Entropy in data mining is the measure of how much disorder or uncertainty is in the data. As depicted in Figure 4.4, a low entropy is characteristic of homogeneous data where most of the instances belong to a particular class. High entropy, on the other hand, has uncertainty and variance among the classification of the instances. Entropy is defined as

$$E(X) = -\sum_{i=1}^{j} \frac{|n_i|}{|n|} log_2 \frac{|n_i|}{|n|} \tag{4.1}$$

where j is the total number of classes, $|n_i|$ is the number of instances of class i and $|n|$ is the total number of instances in the training set in a particular node.

Information gain is described as the amount of information that is gained by knowing the value of an attribute. That is, information gain determines which attribute(s) in a set of training attributes is most useful for distinguishing between the classes to be classified. A decision tree is constructed on those attributes which return the highest information gain. Information gain is a function of the entropy of all instances of the parent node and the entropy of the instances split on a specific attribute. Information gain is defined as

$$Gain(p, x) = E(p) - E(l|x) - E(r|x) \tag{4.2}$$

where p is the training set before the split i.e. parent node, l and r are the subset of the training set that is split based on the value of the attribute x i.e. child nodes.

The C4.5 algorithm uses entropy and information gain to divide the training set into more homogeneous subsets to create decision tress [8]. A step-by-step summary of the algorithm is given below.

1. Check if all instances belong to the same class, then the tree is simply a leaf labeled with the class.

2. Otherwise, for each attribute, calculate the information gain.

3. The attribute with the highest information gain will be the best splitting attribute and the parent node.

4. Repeat steps 2 and 3 at each parent node recursively.

5. Stop splitting when all instances are classified.

6. Prune to generalize the decision tree.

Decision trees created from a training set often contains unnecessary structure and bias. Pruning is done to simplify and generalize the decision tree. C4.5 adopts the strategy of *subtree replacement* where a subtree is replaced by a leaf node if overall information gain is only marginally decreased. Although the accuracy of the tree on the training set is decreased, pruning may increase the accuracy on an independently chosen test set.

Select molecular information of all the patient instances from the clustering data were uploaded, along with the cluster label, into Weka. The C4.5 algorithm was employed to create a decision tree based on the input data. The PART rule generator was used to traverse the tree and generate rules. The results of the decision tree analysis is discussed in the following section.

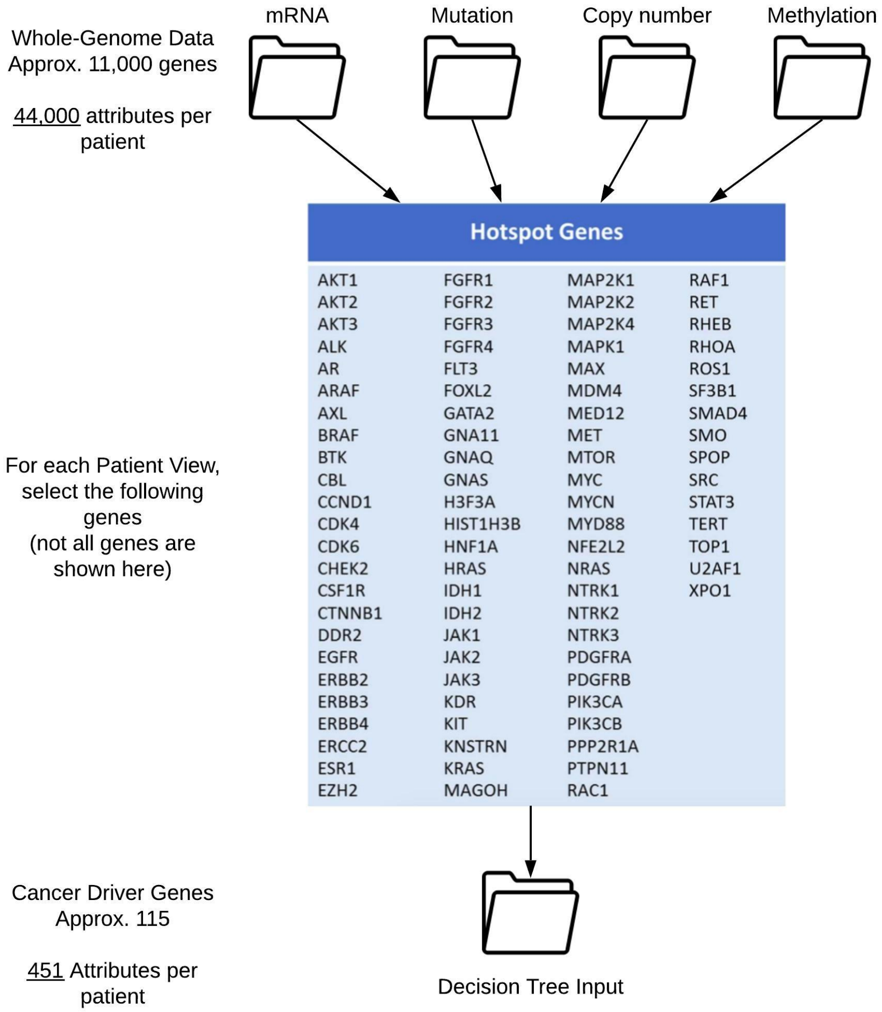
4.3 Decision Tree Results

4.3.1 Decision Tree Input

Clinical trials are experiments conducted to understand the efficacy of a new drug or treatment on human subjects. The NCI-MATCH program is the primary cancer clinical trial at the National Cancer Institute [31]. The program chooses patients based on the genetic makeup of their tumors. Genomic sequencing and other tests are used to determine the genetic makeup of the cancer cells in the patients and patients with identifiable genetic changes that match with treatments in the trial then receive that treatment. The clinical trial assays a subset of the genome identified to be cancer driver genes. This subset of genes will be used as the input to the decision tree.

A whole-genome sequencing of patients in a clinical trial is time consuming and expensive. As such, certain biomarker assays were created that target specific genes that play an important role in cancer development. Oncomine assays are multi-biomarker genomic assays designed for cancer research and used as part of NCI-MATCH [44]. The assay detects mutations, insertions and deletions and copy number changes in almost 120 unique cancer driver genes.

Rather than using the whole genome, the oncomine cancer driver genes were used; significantly reducing the genes per patient from approximately 11,000 to 115 genes. This allows for the targeting of cancer driver genes when creating the decision tree and is a better representation of the data a physician will have when choosing treatment options. For each patient view (mRNA, mutation, copy number and methylation), the cancer driver genes were selected and stored in a csv file. This essentially pruned the data from having approximately 44,000 attributes to 451 attributes for each patient. A schematic of the selection of genes is presented in Figure 4.5. Each instance in the training set now contained the select cancer driver gene data for each patient view and the cluster label used for classification. The csv file was then saved as an ARFF file using the ARFF-viewer tool in Weka and was inputted into the C4.5 algorithm.


mRNA
Mutation
Copy number
Methylation
Whole-Genome Data
Approx. 11,000 genes
44,000 attributes per patient
For each Patient View, select the following genes
(not all genes are shown here)
Cancer Driver Genes
Approx. 115
451 Attributes per patient
Decision Tree Input

Hotspot Genes

AKT1
AKT2
AKT3
ALK
AR
ARAF
AXL
BRAF
BTK
CBL
CCND1
CDK4
CDK6
CHEK2
CSF1R
CTNNB1
DDR2
EGFR
ERBB2
ERBB3
ERBB4
ERCC2
ESR1
EZH2
FGFR1
FGFR2
FGFR3
FGFR4
FLT3
FOXL2
GATA2
GNA11
GNAQ
GNAS
H3F3A
HIST1H3B
HNF1A
HRAS
IDH1
IDH2
JAK1
JAK2
JAK3
KDR
KIT
KNSTRN
KRAS
MAGOH
MAP2K1
MAP2K2
MAP2K4
MAPK1
MAX
MDM4
MED12
MET
MTOR
MYC
MYCN
MYD88
NFE2L2
NRAS
NTRK1
NTRK2
NTRK3
PDGFRA
PDGFRB
PIK3CA
PIK3CB
PPP2R1A
PTPN11
RAC1
RAF1
RET
RHEB
RHOA
ROS1
SF3B1
SMAD4
SMO
SPOP
SRC
STAT3
TERT
TOP1
U2AF1
XPO1

Figure 4.5: Schematic of selecting attributes for decision tree input.

4.3.2 Decision Tree Rules

The PART rule generator extracted rules from the decision tree created by the C4.5 algorithm. A total of 75 rules were extracted from the training set. Figure 4.6 shows the first three rules outputted from the C4.5 algorithm. The precondition of the rules is a series of molecular attributes ANDed together and the consequent is the cluster to which a sample would belong to based on the rules. The number to the left in the parenthesis is the number of instances classified correctly and the number to the right is the number of instances incorrectly classified based on the rule. For example in Rule 1 in Figure 4.6, if all the molecular attributes and respective values are satisfied in a sample, then the sample would be classified into cluster 4. A total of 110 instances were classified using this rule. The remainder of the rules can be found in the Appendix.

```
=== Classifier model (full training set) ===

PART decision list
------------------

STAT3_scnaq > 1 AND
MAGOH_scnaq <= 0 AND
PPP2R1A_scnaq <= 1 AND
DDR2_scnaq > 0 AND
ATRX_scnaq <= 1 AND
FOXL2_scnaq <= 1 AND
NFE2L2_scnaq <= 1: Four (110.0)

DDR2_scnaq > 1 AND
JAK1_scnaq > 0 AND
ERBB2_scnaq <= 0 AND
NRAS_scnaq <= 1 AND
GNA11_scnaq <= 0: Six (6.0/1.0)

DDR2_scnaq > 1 AND
JAK1_scnaq > 0 AND
ERBB2_scnaq <= 0 AND
MAP2K2_scnaq <= 1 AND
JAK1_scnaq > 1 AND
EZH2_mRNA > 0 AND
NTRK3_mRNA <= 1: Three (27.0)
```

Figure 4.6: The first three rules out of seventy five outputted by the C4.5 algorithm.

As with any model, the decision tree model does not encapsulate the training set perfectly. The model had an accuracy of 59.32%. The decision tree was able to classify 595 out of 1003 instances into the correct clusters using the rules. 408 out of 1003 instances were classified into the wrong cluster based on the rules. Multiple attempts of running the algorithm on the data set produced the same result suggesting that this is the best output of the algorithm.

Given these results, the decision tree however was able to give new insights into the molecular data. Out of the four patient views (mRNA, mutation, copy number and methylation), the rules were primarily composed of copy number and mRNA attributes. This potentially suggests that mRNA and copy number data provide more information to partition the instances into clusters than mutation and methylation data. However, more than likely, this could be a consequence of mutation and methylation patient views lacking data as discussed in Section 3.1.1. Of the 451 different attributes, only 143 attributes were deemed relevant to the partition of the clusters. An analysis of these attributes and their function can provide more insights into pathways of cancer development in brain cancers. Also of note, out of the mutation data only the IDH mutation was represented in the rules, suggesting that indeed IDH mutation plays an important role in brain cancer development.

A confusion matrix is also outputted as part of the results. A confusion matrix is a table that is created to describe the performance of a classification model. The confusion matrix in Figure 4.7 suggests that cluster *Four*, which had the highest number of p-values less than 0.05, was the best classified cluster with 90% of the instances classified correctly. Clusters *Five*, *Six* and *Seven* had the next best classification results with nearly 60% of the instances classified correctly. The remaining

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   <-- classified as
  71   5  10  12  10   6   5  11 |   a = One
   6  53   6   0  17  18   9  10 |   b = Two
  14  11  68   0   4  17   4  16 |   c = Three
  10   1   0 110   0   0   0   1 |   d = Four
   8  12   3   0  81   2  19  14 |   e = Five
   9  19   6   0   3  81   3   4 |   f = Six
   4   7  10   2  13   7  83   3 |   g = Seven
  13   6  11   3  16   3   5  48 |   h = Eight
```

Figure 4.7: Confusion matrix from C4.5 output which depicts where each instance from each cluster was classified into based on the rules.

clusters fared worse with less than 55% accurate classification. Clusters *Two* and *Eight* had the worst accuracy with only 44% and 40% correct classification respectively. Considering clusters *Two* and *Eight* were statistically distinct from 6 out of 7 of the other clusters, the classification results are surprising. This implies that the attributes that contribute to the different prognosis of these clusters are not being captured properly in the decision tree model or that the subset of the genome used in the assay is not sufficient to capture the partitions.

Even with accuracy lacking for clusters *Two* and *Eight*, the decision tree model provides a good framework to classify the majority of samples into a majority of the clusters. Unlike other high accuracy Machine Learning (ML) models which use neural nets and other learning schemes, decision tree models output the actual function for classification via the rules. Most ML algorithms are black box algorithms with hard to understand internal behavior; they provide the classification result but not the mapping function. The availability of the rules from a decision tree model allows clinicians to validate and test the rules and allows for a straightforward implementation of a semantic model, which is discussed in the next chapter.

4.4 Discussion

This chapter provided the classification method used to extract a decision tree from the clustering results. Classification determined select molecular attributes of each cluster allowing for any new data to be classified immediately rather than using the k-mean clustering again. The C4.5 algorithm, implemented as J48 in Weka, employed the concept of data entropy to partition the instances of patient gene data. The classification, although not perfect, provides a set of mapping rules to classify the majority of instances into the clusters.

The clustering and the classification framework provides a viable method to handle high-dimensional patient data and to extract usable attributes based on patient similarity. As more data is made available, this application and the results of the method will become more robust to be used for clinical purposes. A semantic model can now be created containing the infrastructure to model the Glioma domain and data. A semantic model, as discussed in the following chapter, allows for the representation of the Glioma domain along with the clustering and classification results in a dynamic model.

Chapter 5: **Ontologies Supported by Data**

5.1 Semantic Model Software Architecture

The software organization for the development and generation of the semantic model is given in Figure 5.1. The semantic model consists of glioma-specific ontologies, instances of data and rules derived from the classification results. The software model is implemented in Java with the ontology created using the RDF/OWL framework and the data imported from text files.

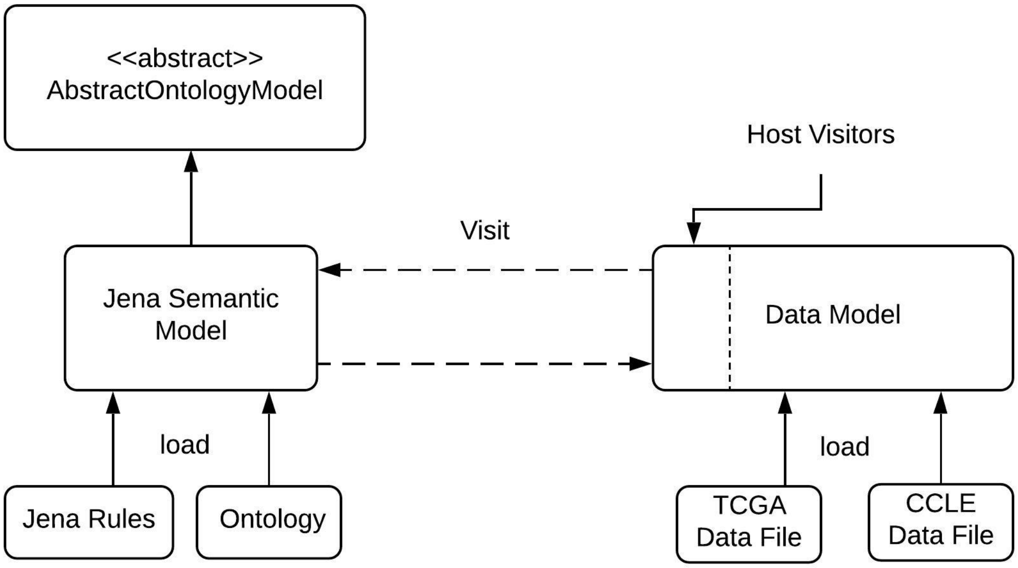


Figure 5.1: Software Architecture of generation of semantic models.

Generation of the semantic model begins with the creation of domain-specific ontologies in OWL. In this case, a glioma, TCGA patient and CCLE cell line ontolgoies are created to capture the domain space. Each ontology is given ontological descriptions, hierarchy of classes and data and object properties. Classification rules from the data mining investigation and domain-specific constraints are then transformed into Jena rules to be read by the model. Finally, data is ingested into the model from TCGA and CCLE text files. The Data model reads and imports the data. A visitor design pattern is implemented to transfer the data model to the Jena Semantic model. Once all the components of the model are created, the semantic rules can be applied to the model and applications can visit, query and reason with the model.

5.2 Glioma Ontology Models

Protege is an open source suite of tools that facilitate the building of knowledge bases through ontologies [37]. It is used to create domain ontologies, specify domain relationships, datatype properties, object properties and specify constraints on said domains and properties. Protege was used in the creation of the glioma-specific ontology. A brief and simplified description of each of the ontologies is provided below.

Figure 5.2 provides a simplified representation of the Glioma ontology. The Glioma class is further broken down into the IDH wild-type and IDH mutation subclasses consistent with literature [11] and classification results. The IDH wild-type and mutation classes are then stratified further based on the classification results from Section 3.2. This ontology captures the domain relationship via sub-class properties and allows reasoning of instances assigned from the leaf classes up to the root Glioma class.

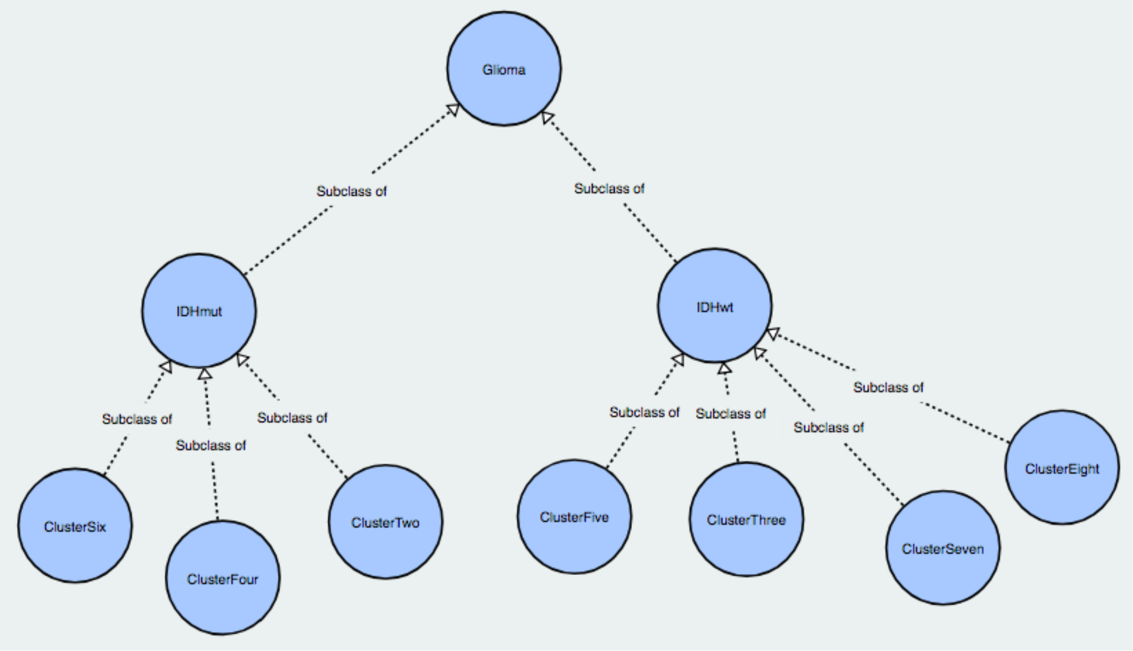


Figure 5.2: Glioma Ontology with clusters from classification results.

Figure 5.3 presents the simplified representation of the TCGA patient ontology. The ontology incorporates patient clinical data and the molecular attributes deemed important by the data mining investigation. Note that only a select few molecular attributes are shown in the figure. This ontology enables the ingestion of patient instances from the TCGA patient data into the semantic model and allows mapping of patients to the glioma clusters based on the datatype properties of each instance. The mapping is defined via the hasCluster object property.

Figure 5.4 presents the simplified representation of the CCLE cell line ontology.

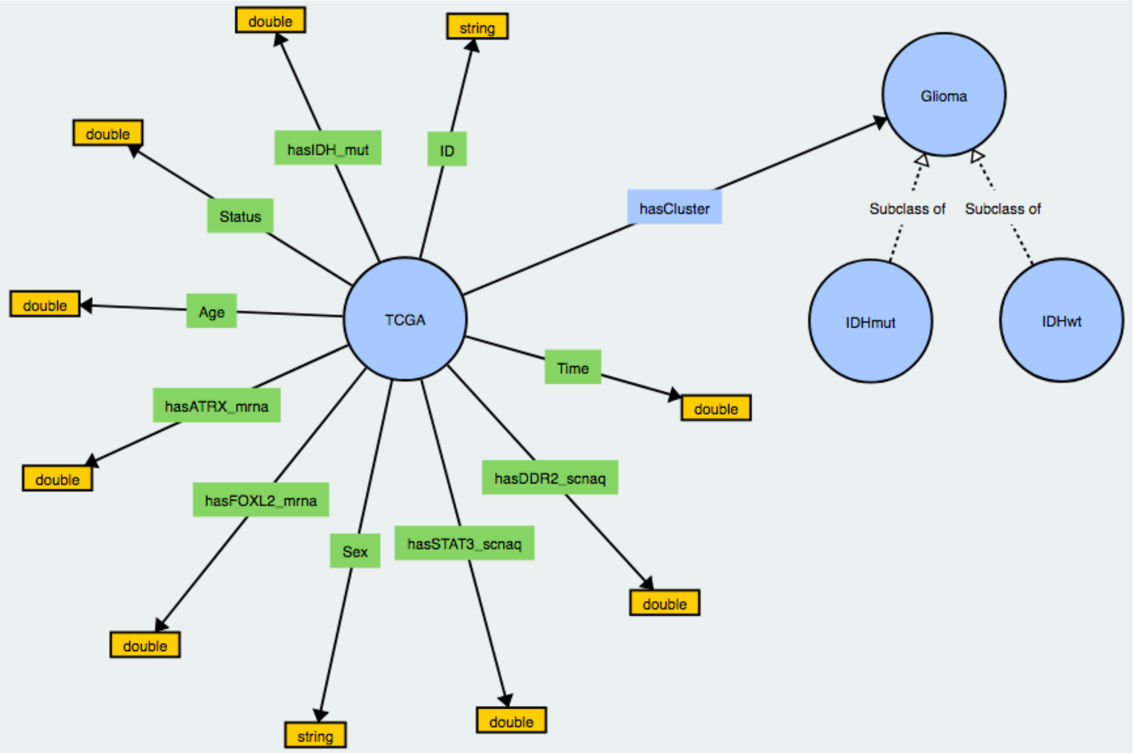


Figure 5.3: TCGA patient ontology.

The ontology incorporates all 143 molecular attributes considered relevant by the data mining investigation. Note that only a select few molecular attributes are shown in the figure. The cell line ontology enables the creation of cell line instances which are ingested from the CCLE data model. Once all cell line instances are created, the rules will map each cell line to a Glioma cluster via the hasCluster object property.

The ontologies provide a simplified but adequate representation of the Glioma domain and allow data to be created as instances of the ontology. New patient data can now be ingested into the model and with the help of the rules can be mapped immediately to the relevant clusters. Likewise, as more cell line data is produced, it can also be incorporated into the model seamlessly. Although the

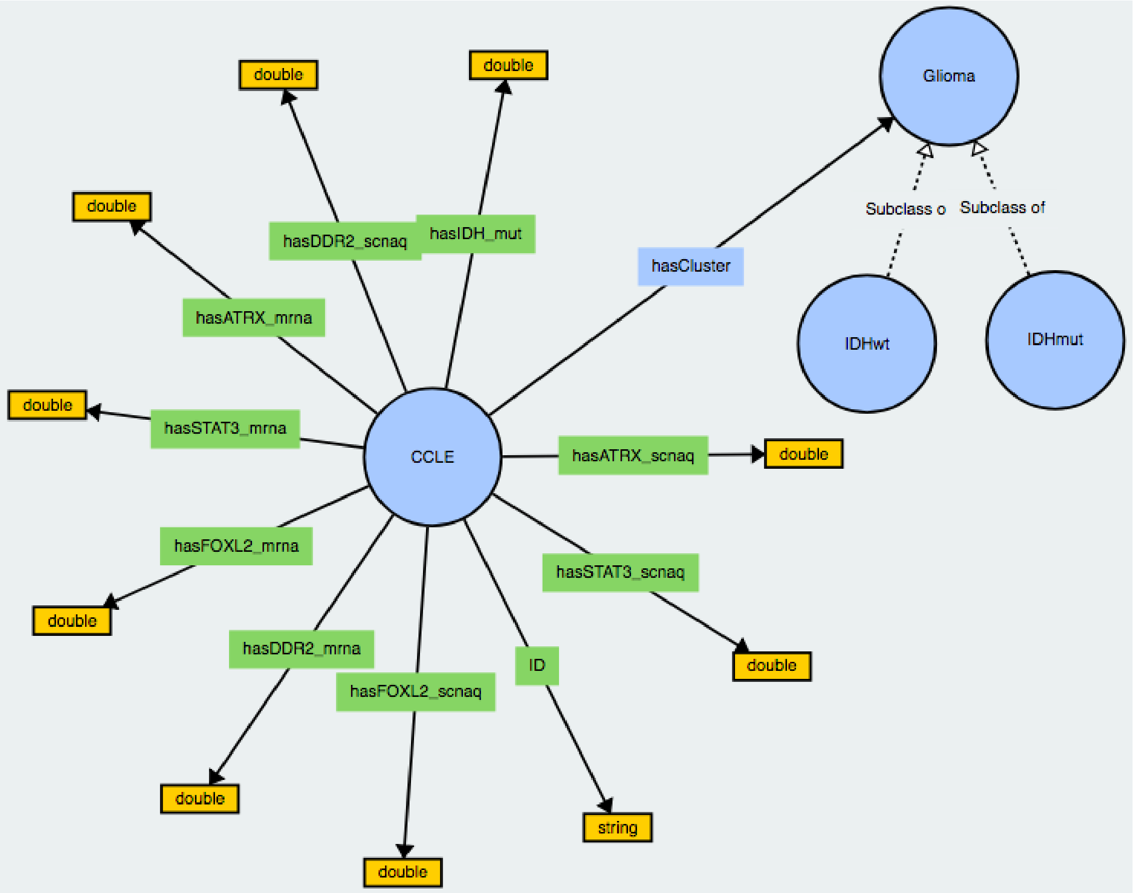


Figure 5.4: CCLE cancer cell line ontology.

current ontologies are created for the purposes of matching cell lines to clusters, more semantic description, domains and data can be added to the model in the future. As the model is refined and given more functionality, it can be used for more complex tasks and narrower reasoning.

5.3 Mapping of Preclinical Models

We proposed that a semantic approach will enable the rational selection of preclinical models and patients for clinical trials. The Glioma semantic model provides a framework to achieve these goals. The Glioma ontologies along with the rules allow for the ingestion of patient and cell line model data and allow for the mapping and selection of models based on molecular similarity. This section provides a schematic of the mapping and the results of the selection of preclinical models for each cluster.

```
prefix af: <http://www.nih.gov/abraham/GliomaOntology.owl#>.

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

//Propagation Rule 01: Propogate Class hierarchy...
[rdfs01: (?x rdfs:subClassOf ?y), notEqual(?x, ?y) ->
[(?a rdf:type ?y) <- (?a rdf:type ?x)] ]

//Rule 01:
[ Rule01: (?x rdf:type af:Ccle) (?x af:hasSTAT3_scnaq ?stat)
(?x af:hasMAGOH_scnaq ?ma) (?x af:hasPPP2R1A_scnaq ?pp)
(?x af:hasDDR2_scnaq ?dd) (?x af:hasATRX_scnaq ?atr)
(?x af:hasFOXL2_scnaq ?fox) (?x af:hasNFE2L2_scnaq ?nef)
greaterThan(?stat, 1.0) lessThan(?ma, 1.0)
greaterThan(?ma, -1.0) lessThan(?pp, 2.0)
greaterThan(?pp, -1.0) greaterThan(?dd, 0.0)
lessThan(?atr, 2.0) greaterThan(?atr, -1.0)
lessThan(?fox, 2.0) greaterThan(?fox, -1.0) lessThan(?nef, 2.0)
greaterThan(?nef, -1.0) -> (?x af:hasCluster af:ClusterFour) ]

//Rule 03:
[ Rule03: (?x rdf:type af:Ccle) (?x af:hasDDR2_scnaq ?dd)
(?x af:hasJAK1_scnaq ?jak) (?x af:hasERBB2_scnaq ?erb)
(?x af:hasMAP2K2_scnaq ?map) (?x af:hasEZH2_mRNA ?ez)
(?x af:hasNTRK3_mRNA ?nt) greaterThan(?dd, 1.0)
greaterThan(?jak, 0.0) equal(?erb, 0.0) le(?map, 1.0)
greaterThan(?map, -1.0) greaterThan(?ez, 0.0) le(?nt, 1.0)
greaterThan(?nt, -1.0) -> (?x af:hasCluster af:ClusterThree)]
```

Figure 5.5: Classification Rules transformed to Jena Rules.

Classification rules from the data mining investigation was first transformed into the standard format of Jena Rules. Figure 5.5 depicts the class propagation and classification rules in the Jena format. The propagation rule propagates class hierarchy; placing constraints on the type an instance can be. The remaining rules are classifier rules. For example, in Rule 01, the reasoner traverses the RDF knowledge graph of CCLE instances and checks if the rule constraints are satisfied for each instance. If the rules are satisfied then the CCLE instance is mapped to the ClusterFour class in the Glioma Ontology via the hasCluster object property. The rules can be executed at any time creating a dynamic and responsive model.

Figure 5.6 depicts the graph transformation that occurs when a rule is satisfied. In the figure, an instance of the CCLE class with its particular molecular attribute satisfies the rule which creates an object property from the instance to the ClusterTwo class instance. From a semantic point of view, the CCLE instance now has access to all class descriptions and properties of ClusterTwo and its parent classes. Once all the rules from the classification results are transformed into Jena rules, they can be executed sequentially and all instances in the Glioma ontology satisfying the rules will be mapped to the relevant cluster.

Tables 5.1 and 5.2, provides the results of the mapping of preclinical models to each cluster. The table provides the IDs of CCLE models which responded to each cluster based on the classification rules. Once all the rules were executed, those CCLE instances that satisfied the rules were then assigned an object property to the relevant cluster. An iterator was then used to iterate through the CCLE models for each cluster. These set of models for each cluster have the closest molecular similarity to the patients in each of the clusters. The hope is that these subset of models will lead to better understanding of the efficacy of treatments for a patient diagnosed into the respective cluster. These models could potentially provide better prognostic insights into how well the patient might fare to experimental treatment.

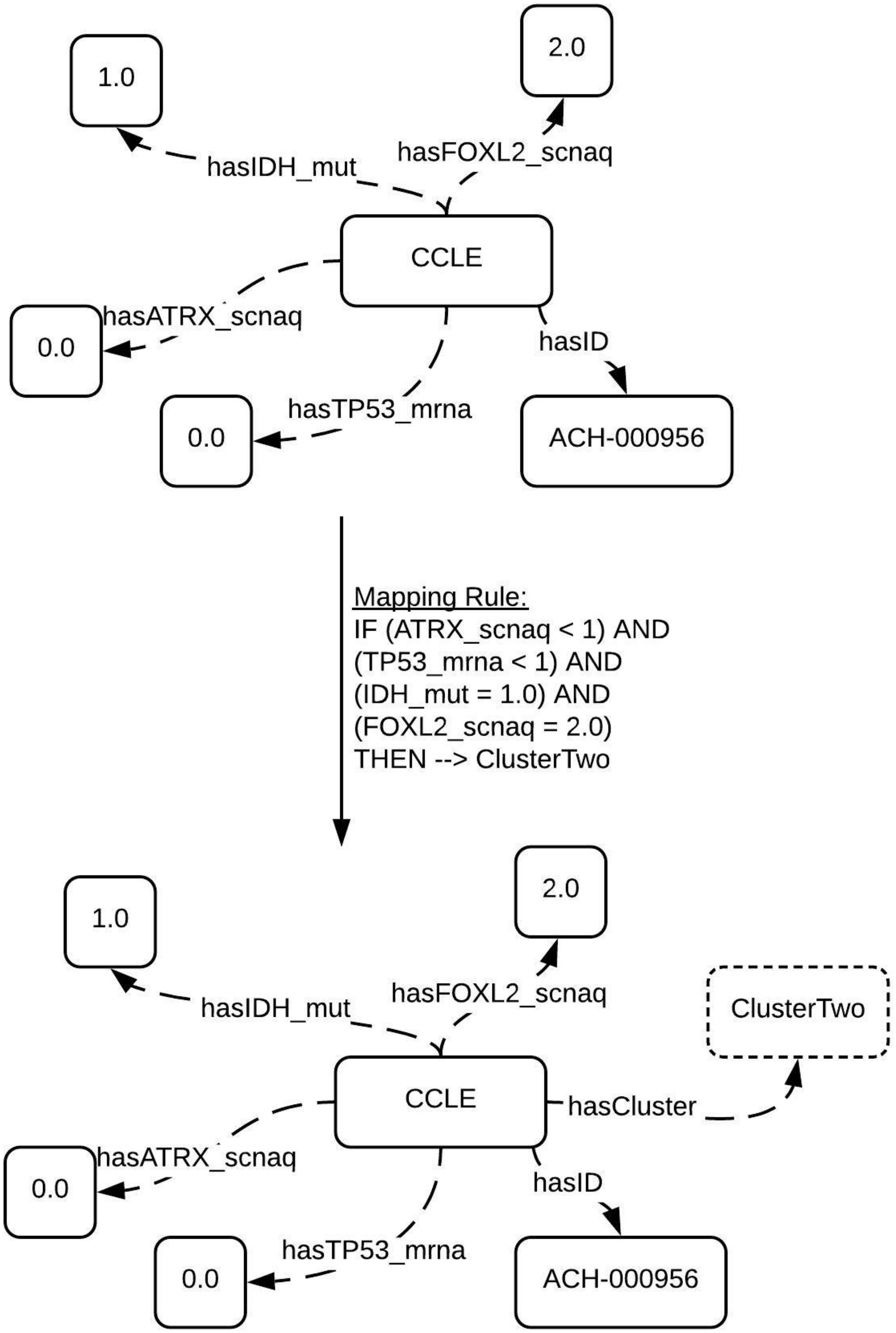


Figure 5.6: Mapping of CCLE instance to Cluster.

Considering each cluster has different overall survivability, preclinical trials can also be tailored to each cluster of patients narrowing the cohort for the trial. With the selection of preclinical models based on the molecular basis of patients with different prognosis, the translational barrier of preclinical to clinical trials can be shortened.

Cluster ID	Num. of CCLE models	CCLE IDs
1	18	ACH-000927, ACH-000350, ACH-000345, ACH-000865 ACH-000596, ACH-000769, ACH-000444, ACH-000934 ACH-000523, ACH-000675, ACH-000379, ACH-000816 ACH-000203, ACH-000490, ACH-000461, ACH-000147 ACH-001302, ACH-001740
2	13	ACH-000844, ACH-000731, ACH-000690, ACH-000190 ACH-000523, ACH-000675, ACH-000870, ACH-000399 ACH-000871, ACH-000816, ACH-000358, ACH-000443 ACH-000232
3	34	ACH-000948, ACH-000956, ACH-000052, ACH-000880 ACH-000191, ACH-000245, ACH-000402, ACH-000867 ACH-000805, ACH-000256, ACH-000278, ACH-000352 ACH-000196, ACH-000668, ACH-000178, ACH-000724 ACH-000671, ACH-000595, ACH-000583, ACH-000849 ACH-000391, ACH-000666, ACH-000841, ACH-000610 ACH-000789, ACH-000116, ACH-000751, ACH-000617 ACH-000409, ACH-000090, ACH-000790, ACH-000087 ACH-000955, ACH-000430
4	13	ACH-000788, ACH-000248, ACH-000120, ACH-000253 ACH-000756, ACH-000631, ACH-000769, ACH-000733 ACH-000436, ACH-000363, ACH-000312, ACH-000661 ACH-001111

Table 5.1: CCLE models which responded to classification rules for clusters 1 through 4.

Cluster ID	Num. of CCLE models	CCLE IDs
5	24	ACH-000649, ACH-000593, ACH-000516, ACH-000392 ACH-000644, ACH-000978, ACH-000633, ACH-000575 ACH-000971, ACH-000994, ACH-000721, ACH-000178 ACH-000801, ACH-000632, ACH-000250, ACH-000313 ACH-000319, ACH-000980, ACH-000434, ACH-000121 ACH-000614, ACH-000811, ACH-000686, ACH-001302
6	11	ACH-000880, ACH-000867, ACH-000111, ACH-000196 ACH-000595, ACH-000573, ACH-000666, ACH-000177 ACH-000751, ACH-000617, ACH-000090
7	16	ACH-000940, ACH-000402, ACH-000805, ACH-000903 ACH-000239, ACH-000357, ACH-000176, ACH-000200 ACH-000062, ACH-000398, ACH-000149, ACH-000908 ACH-000898, ACH-000567, ACH-000656, ACH-000958
8	31	ACH-000046, ACH-000927, ACH-000457, ACH-000253 ACH-000868, ACH-000739, ACH-000799, ACH-000472 ACH-000476, ACH-000858, ACH-000604, ACH-000685 ACH-000301, ACH-000390, ACH-000019, ACH-000058 ACH-000453, ACH-000737, ACH-000675, ACH-000646 ACH-000442, ACH-000181, ACH-000465, ACH-000341 ACH-000016, ACH-000623, ACH-000344, ACH-000686 ACH-000147, ACH-000568, ACH-001053

Table 5.2: CCLE models which responded to classification rules for clusters 5 through 8.

5.4 Discussion

This chapter introduced the Semantic Web framework and the application of the framework to the Glioma Domain. The clustering and classification results are now represented in a simplified Glioma semantic model. The semantic model consists of the Glioma ontology, the patient ontology, the cell line ontology and instances of data. We present the Glioma semantic model as a dynamic model into which new data can be ingested and reasoned for the purposes of mapping to prognostically relevant clusters.

The power of the Semantic Web lies in its ability to integrate different domains and allow reasoners to reason over these domains. This semantic model serves to be a small foundation to build a larger, more comprehensive model of the Glioma domain. Ontologies from disparate domains such as patient symptoms and patient microbiome can now be created and integrated into the semantic model and rules can be executed which take into account these disparate domains. The development of the semantic model can ultimately become a powerful tool in the practice of precision medicine.

Chapter 6: **Conclusions and Future Work**

6.1 Conclusion

As advances in medical technologies continue to rapidly grow, it is imperative to create frameworks and algorithms that utilize these new streams of data to ultimately help the patient. In this paper, we provide a semantic model coupled with a machine learning framework to turn high dimensional patient data into a dynamic model. We provide an application of this model to bridge the gap between the disease and preclinical space. This Glioma specific model provides a foundation onto which other domains can be incorporated to create an all-encompassing patient model to aid in the practice of precision medicine. The human body is a true system of systems, and frameworks proposed in this thesis as well as others will be vital to the understanding of this complex system.

6.2 Future Work

The work done in this thesis makes progress towards a digital twin of a patient. The digital twin architecture for personalized medicine, presented in Figure 1.8 and Figure 1.10, lay the foundation to incorporate domains relevant to the treatment of a patient. These high-level schematics provide a framework to use semantic modeling to serve as an ‘operating system’ for patients and opens new digital ecosystems for improved services and treatment.

Symptoms management apps such as GlioNCI [45], provide an interface for patients to report their symptoms and their quality of life. And with the advent of smart wearable technologies, basic clinical instruments such as the electrocardiogram and other sensing instruments are within the reach of nearly every patient. The digital twin architecture enables the integration of these apps and other patient sensing systems to create a comprehensive and real-time model of a patient. We envision a comprehensive model, validated by medical practitioners, incorporating patient data, real-time patient dynamics, ontologies, models and rules governing treatment and diagnosis to aid future clinicians in practicing precision medicine.

Appendix A: Classification Rules

The following is the full set of rules outputted by the C4.5 Decision Tree algorithm.

```
Classifier model (full training set)

PART decision list
------------------
1
STAT3_scnaq > 1 AND
MAGOH_scnaq <= 0 AND
PPP2R1A_scnaq <= 1 AND
DDR2_scnaq > 0 AND
ATRX_scnaq <= 1 AND
FOXL2_scnaq <= 1 AND
NFE2L2_scnaq <= 1: Four (110.0)
2
DDR2_scnaq > 1 AND
JAK1_scnaq > 0 AND
ERBB2_scnaq <= 0 AND
NRAS_scnaq <= 1 AND
GNA11_scnaq <= 0: Six (6.0/1.0)
3
DDR2_scnaq > 1 AND
JAK1_scnaq > 0 AND
ERBB2_scnaq <= 0 AND
MAP2K2_scnaq <= 1 AND
JAK1_scnaq > 1 AND
EZH2_mRNA > 0 AND
NTRK3_mRNA <= 1: Three (27.0)
4
STAT3_scnaq <= 0 AND
IDH_mut > 0 AND
MAGOH_scnaq > 0 AND
TP53_scnaq <= 0 AND
MAP2K2_scnaq <= 0 AND
SIRT2_scnaq > 0 AND
KLF4_scnaq <= 1: Six (57.41)
5
H3F3A_scnaq > 1 AND
NRAS_scnaq > 1 AND
HRAS_scnaq <= 0 AND
DDR2_scnaq > 1 AND
MAP2K4_mRNA <= 1 AND
AKT3_scnaq > 1 AND
NTRK3_mRNA <= 1 AND
MAP2K2_mRNA > 0: Three (28.0)
6
PPM1D_scnaq > 1 AND
MAGOH_mRNA <= 0 AND
ATRX_mRNA <= 1 AND
KLF4_mRNA <= 1 AND
MAP2K2_mRNA <= 1: Eight (26.0/2.0)
7
PPM1D_scnaq > 1 AND
MAGOH_scnaq <= 0 AND
ERBB2_scnaq > 0 AND
SPOP_mRNA > 0 AND
GATA2_scnaq > 0 AND
RAF1_mRNA > 0 AND
HRAS_mRNA > 0 AND
MYCN_mRNA > 0: One (17.0)
```

```
8
H3F3A_scnaq > 1 AND
MAGOH_scnaq <= 0 AND
SMO_mRNA > 1 AND
GATA2_mRNA > 1: Three (3.0)
9
STAT3_scnaq <= 0 AND
CCND1_scnaq <= 1 AND
SMARCB1_scnaq <= 1 AND
H3F3A_scnaq <= 1 AND
IDH_mut <= 0 AND
RAF1_scnaq <= 1 AND
EZH2_scnaq > 1 AND
KDR_scnaq > 0: Seven (45.64)
10
H3F3A_scnaq > 1 AND
MAGOH_scnaq <= 0 AND
SPOP_mRNA <= 0: Eight (8.0/1.0)
11
DDR2_scnaq > 1 AND
JAK1_scnaq <= 0 AND
CSF1R_mRNA <= 1 AND
NRAS_mRNA <= 1 AND
RAF1_mRNA > 0 AND
CCND3_mRNA > 0 AND
RAF1_mRNA > 1: One (10.0)
12
H3F3A_scnaq > 1 AND
MAGOH_scnaq <= 0 AND
SIRT2_scnaq <= 1 AND
ERCC2_mRNA <= 1 AND
CIC_mRNA > 0: One (7.0/1.0)
13
PPM1D_scnaq > 1 AND
MAGOH_scnaq <= 0 AND
RHOA_scnaq <= 1 AND
AR_mRNA <= 0: Four (9.0/1.0)
14
DDR2_scnaq > 1 AND
JAK1_scnaq <= 0 AND
STAT3_scnaq > 0 AND
ROS1_scnaq <= 1: Four (4.0/1.0)
15
DDR2_scnaq > 1 AND
JAK1_scnaq <= 0 AND
CCND3_mRNA > 1: Five (3.0)
16
DDR2_scnaq > 1 AND
GNAS_mRNA > 0 AND
ERBB2_scnaq <= 0 AND
GNAQ_scnaq <= 1 AND
EZH2_mRNA <= 1: Seven (12.0/1.0)
17
DDR2_scnaq > 1 AND
RAF1_mRNA <= 0 AND
SETD2_mRNA <= 1 AND
FLT3_scnaq <= 1 AND
GNAS_mRNA <= 0 AND
SMAD4_mRNA <= 1 AND
CTNNB1_mRNA <= 1: Eight (15.0/1.0)
18
DDR2_scnaq > 1 AND
PPP2R1A_scnaq > 1 AND
SMO_mRNA > 1 AND
MAX_scnaq <= 1: Three (9.0/1.0)
19
DDR2_scnaq > 1 AND
PPP2R1A_scnaq > 1 AND
SMO_mRNA <= 1 AND
HRAS_scnaq <= 1 AND
MAGOH_mRNA > 0: Three (14.0/1.0)
20
DDR2_scnaq > 1 AND
PPP2R1A_scnaq > 1 AND
GNAQ_mRNA > 0: Two (5.0)
21
DDR2_scnaq > 1 AND
PPP2R1A_scnaq > 1 AND
AKT3_mRNA <= 0: Seven (3.0)
22
DDR2_scnaq > 1 AND
NTRK1_scnaq <= 1 AND
CCND2_mRNA <= 1: Six (3.0/1.0)
```

```
23
DDR2_scnaq > 1 AND
ERBB2_scnaq <= 0 AND
BTK_mRNA <= 1 AND
SIRT2_mRNA <= 1: Three (11.0)
24
DDR2_scnaq > 1 AND
ERBB2_scnaq > 0 AND
NRAS_mRNA <= 0 AND
AKT2_mRNA <= 0: Three (2.0)
25
DDR2_scnaq > 1 AND
ERBB2_scnaq > 0 AND
NRAS_mRNA > 0 AND
SETD2_mRNA <= 0 AND
GATA2_scnaq > 0: Three (8.0)
26
DDR2_scnaq > 1 AND
RAF1_mRNA > 0 AND
SETD2_mRNA > 0 AND
IDH2_mRNA > 0 AND
NRAS_scnaq > 1 AND
MAPK1_mRNA > 0 AND
KDR_scnaq <= 1: One (21.0)
27
PPM1D_scnaq > 1 AND
CHEK1_scnaq > 0 AND
SMARCA4_mRNA > 0 AND
PIK3CA_mRNA > 0 AND
AXL_scnaq > 1 AND
GNA11_mRNA > 1: Two (5.0)
28
PPM1D_scnaq > 1 AND
GNAS_scnaq <= 0 AND
DDR2_scnaq <= 1 AND
NF1_mRNA > 0 AND
AXL_scnaq <= 1 AND
CBL_mRNA <= 1: One (12.0)
29
PPM1D_scnaq > 1 AND
GNAS_scnaq <= 0 AND
NTRK2_mRNA > 0 AND
NTRK3_mRNA <= 1: Six (10.0)
30
PPM1D_scnaq > 1 AND
RAF1_mRNA <= 0 AND
CBL_scnaq > 0 AND
HIST1H3C_mRNA > 0: Eight (10.0/1.0)
31
PPM1D_scnaq > 1 AND
RAF1_mRNA <= 0 AND
AKT1_meth <= 1: Seven (2.0)
32
PPM1D_scnaq > 1 AND
CSF1R_scnaq > 1 AND
CDK6_mRNA > 0 AND
U2AF1_mRNA > 0: Three (8.0)
33
PPM1D_scnaq > 1 AND
NTRK1_mRNA <= 1 AND
MET_mRNA <= 1 AND
NRAS_mRNA > 0 AND
CCND2_mRNA > 0 AND
MED12_mRNA > 0: One (25.0/1.0)
34
MAP2K4_scnaq > 0 AND
MYC_scnaq > 1 AND
NRAS_scnaq > 0 AND
AKT3_scnaq <= 1 AND
JAK1_mRNA > 0 AND
CHEK2_scnaq > 0 AND
MAX_scnaq > 0: Two (35.0)
35
PPP2R1A_scnaq > 1 AND
IDH_mut > 0 AND
CIC_scnaq > 1 AND
NRAS_scnaq > 0 AND
SOX2_mRNA > 0 AND
BTK_mRNA > 0 AND
JAK1_mRNA > 0 AND
ATRX_scnaq > 0 AND
TP53_meth <= 1: Two (20.83/1.22)
36
MAP2K4_scnaq > 0 AND
FOXL2_scnaq <= 1 AND
DDR2_scnaq <= 1 AND
CCND1_scnaq <= 1 AND
JAK1_scnaq <= 1 AND
KRAS_scnaq <= 1 AND
CCND2_scnaq <= 1 AND
FGFR1_scnaq <= 1 AND
CCND3_scnaq > 0: Five (62.0/2.0)
37
PPP2R1A_scnaq > 1 AND
CCND1_scnaq <= 1 AND
AKT2_scnaq > 1 AND
KDR_scnaq > 0 AND
AKT1_scnaq > 1 AND
NF1_scnaq <= 0: Seven (6.0)
```

```
38
GNA11_scnaq > 1 AND
SMARCA4_scnaq <= 1 AND
AKT2_mRNA <= 0: Two (2.0/1.0)
39
GNA11_scnaq > 1 AND
SMARCA4_scnaq > 1 AND
MAP2K4_scnaq <= 0 AND
NRAS_scnaq <= 1 AND
EGFR_scnaq > 0 AND
KIT_scnaq > 1: Seven (16.78/0.78)
40
IDH_mut > 0 AND
NRAS_scnaq <= 0 AND
AKT2_scnaq <= 0 AND
ERBB2_mRNA <= 1: One (13.63/0.41)
41
SMARCB1_scnaq > 1 AND
NFE2L2_mRNA > 1 AND
ARAF_scnaq > 0 AND
EGFR_scnaq <= 1 AND
FGFR1_mRNA > 0: Five (13.59)
42
HRAS_scnaq <= 0 AND
NRAS_scnaq > 0 AND
SIRT2_scnaq > 0 AND
JAK3_mRNA > 0 AND
NFE2L2_scnaq > 1 AND
FGFR1_mRNA > 0: Six (13.0/1.0)
43
SMARCB1_scnaq > 1 AND
NFE2L2_mRNA > 1 AND
GNAS_mRNA <= 0: Two (5.7/1.0)
44
SMARCB1_scnaq > 1 AND
ROS1_mRNA > 0 AND
CCND2_mRNA <= 1 AND
PIK3CB_mRNA <= 1 AND
FGFR4_mRNA > 0: Eight (18.0)
45
HRAS_scnaq > 0 AND
NF1_scnaq > 1 AND
CDK4_mRNA > 0 AND
TERT_mRNA > 0 AND
CIC_mRNA > 0: Eight (7.0)
46
GNA11_scnaq > 1 AND
ATRX_scnaq > 0 AND
H3F3A_scnaq <= 1 AND
ROS1_scnaq > 0 AND
MAP2K1_scnaq > 0 AND
EGFR_scnaq <= 1 AND
XPO1_scnaq <= 1 AND
HNF1A_scnaq > 0: Five (19.0)
47
MYD88_scnaq > 1 AND
CHEK2_scnaq > 1 AND
MAPK1_mRNA <= 1: One (4.0/1.0)
48
MYD88_scnaq > 1 AND
CTNNB1_scnaq > 1 AND
CCND1_scnaq > 1 AND
SMAD4_mRNA > 0 AND
GNA11_scnaq <= 1 AND
IGF1R_mRNA > 0: Six (16.59/1.59)
49
HRAS_scnaq <= 0 AND
NRAS_scnaq > 0 AND
MDM2_mRNA <= 1 AND
PDGFRA_scnaq <= 1 AND
BTK_mRNA > 0 AND
CCND3_scnaq <= 1: Two (15.59/0.59)
50
HRAS_scnaq <= 0 AND
NRAS_scnaq > 0 AND
CHEK2_mRNA > 0 AND
CDK4_scnaq <= 0 AND
NTRK1_mRNA > 0: Six (7.0/1.0)
51
HRAS_scnaq <= 0 AND
GNA11_scnaq > 1 AND
CCND2_mRNA > 1: Seven (4.0)
52
HRAS_scnaq <= 0 AND
IGF1R_mRNA <= 1 AND
RB1_scnaq <= 1 AND
FOXL2_mRNA > 0: Three (19.0/2.0)
```

```
53
PPP2R1A_scnaq > 1 AND
HRAS_scnaq > 0 AND
MYCN_mRNA <= 1 AND
MYC_scnaq <= 1 AND
RET_scnaq <= 1 AND
CCND1_scnaq <= 1 AND
SMO_mRNA > 0 AND
ERCC2_mRNA > 0 AND
NF1_mRNA <= 1: Seven (21.0)
54
STAT3_scnaq > 0 AND
DDR2_scnaq > 1 AND
MET_mRNA > 0: One (6.0/1.0)
55
STAT3_scnaq > 0 AND
DDR2_scnaq > 1 AND
AKT3_mRNA <= 1: Two (2.0)
56
STAT3_scnaq > 0 AND
DDR2_scnaq <= 1 AND
TP53_mRNA > 1 AND
KLF4_scnaq <= 1 AND
MYD88_mRNA > 1: Five (10.0/1.0)
57
STAT3_scnaq > 0 AND
DDR2_scnaq <= 1 AND
TP53_mRNA > 1 AND
SIRT2_mRNA > 0 AND
EGFR_scnaq <= 1 AND
EGFR_mRNA <= 1: One (8.0)
58
STAT3_scnaq > 0 AND
DDR2_scnaq <= 1 AND
SOX2_mRNA > 1 AND
ERBB4_mRNA > 0 AND
PIK3R1_scnaq <= 1: Two (12.7)
59
STAT3_scnaq > 0 AND
DDR2_scnaq <= 1 AND
SPOP_scnaq <= 0: Two (3.0/1.0)
60
SMARCA4_scnaq > 1 AND
PPM1D_scnaq <= 0 AND
MAPK1_mRNA > 0: Seven (8.18)
61
MYD88_scnaq > 1 AND
PTEN_scnaq > 0 AND
BRAF_scnaq > 0 AND
JAK2_mRNA > 0: Six (7.0)
62
MYD88_scnaq > 1 AND
ERBB2_scnaq <= 0: Seven (7.77/1.59)
63
PPP2R1A_scnaq > 1 AND
IDH_mut <= 0 AND
PDGFRA_mRNA <= 1 AND
ALK_scnaq > 0: Five (14.0)
64
PPP2R1A_scnaq > 1 AND
ATRX_mRNA > 1: Two (4.0)
65
PPP2R1A_scnaq <= 1 AND
NRAS_scnaq > 1 AND
KLF4_mRNA <= 0 AND
BRAF_mRNA > 1: Three (3.0)
66
PPP2R1A_scnaq <= 1 AND
SMARCA4_mRNA > 1 AND
ERBB2_mRNA > 0: Six (6.0/1.0)
67
PPP2R1A_scnaq > 1: Seven (3.0)
68
SMARCA4_mRNA > 1 AND
AXL_mRNA <= 0: One (2.59)
69
SETD2_scnaq <= 1 AND
RHEB_mRNA <= 1 AND
TP53_mRNA <= 1 AND
PPM1D_mRNA <= 1 AND
BRAF_mRNA <= 1: Eight (12.0)
70
RAC1_mRNA <= 0 AND
MAP2K1_mRNA > 1 AND
KLF4_scnaq <= 1: Five (6.0)
71
RAC1_mRNA > 0 AND
PPP2R1A_mRNA <= 1 AND
AKT1_scnaq <= 1 AND
CCND1_scnaq > 0 AND
GNA11_mRNA <= 1 AND
CDK4_mRNA <= 1: Five (9.0)
72
RAC1_mRNA > 0 AND
PPP2R1A_mRNA <= 1 AND
FLT3_mRNA <= 1: Two (9.0)
73
U2AF1_scnaq > 0 AND
RHEB_mRNA <= 1: Eight (8.0/1.0)
```

Appendix B: Jaccard Algorithm

```
function [jaccardNum,jaccardDenom] = jaccard(dataset)
%jaccard: This funciton calcualtes the jaccard values b/w datasets
%For each columnn of data, simply take intersection/union

data = size(dataset, 2) - 1;
jaccardNum = ones(data, data)*-1;
jaccardDenom = ones(data, data)*-1;

%loop through columns first, then row
for i=2:1:size(dataset, 2)

    for j=i+1:1:size(dataset, 2)
        numerator = 0;
        denom = 1;
        %loop through row
        for k = 1:1:size(dataset, 1)
            %see if data is available
            if (~isnan(dataset(k, i)) & ~isnan(dataset(k, j)))
                denom = denom + 1;
                %check if data is equal
                if dataset(k, i) == dataset(k, j)
                    numerator = numerator + 1;
                end
            end
        end
        jaccardNum(i-1, j-1) = numerator;
        jaccardDenom(i-1, j-1) = denom;
        jaccardNum(j-1, i-1) = numerator;
        jaccardDenom(j-1, i-1) = denom;
    end
end
end
```

Figure B.1: Jaccard Algorithm implemented in matlab.

Appendix C: Neo4j Cypher Queries

```
Counting all Nodes in the Database:
MATCH(n)
RETURN COUNT(n)

Identifying TCGA\CCLE Nodes:
MATCH(n:tcga)
RETURN COUNT (n)

Counting nodes with a certain numerical attribute:
MATCH(n:tcga)
WHERE n.propertyName <= value
RETURN COUNT(n)

Averaging numerical attributes:
MATCH(n:tcga)
WHERE n.propertyName <= value
RETURN avg(n)

Aggregating attributes:
MATCH(n:tcga)
WHERE n.property1 = value1 AND n.property2 = value2
RETURN COUNT(n)

Identifying relationships between tcga nodes:
MATCH (n:tcga)-[r:relationshipName]-(k:tcga)
WHERE n.property = value
RETURN COUNT(n)

Creating relationships between tcga nodes:
MATCH (n:tcga)-[r:relationshipName]-(k:tcga)
WHERE n.property = value
CREATE (n)-[r:newRelationship]->(k)
```

Bibliography

[1] Antoniou G. and Van Harmelen F. Web Ontology Language: OWL. *Handbook on Ontologies, Springer*, pages 67–92, 2004.

[2] Apache Jena:. An Open Source Java framework for building Semantic Web and Linked Data Applications. For details, see https://jena.apache.org/, 2016.

[3] Arp R., Smith B., and Spear A.D. *Building Ontologies with Basic Formal Ontology*. The MIT Press, 2015.

[4] Austin M.A., Coelho M., Heidarinejad M., and Delgoshaei P. Architecting Smart City Digital Twins: A Combined Semantic Model and Machine Learning Approach (Paper in Review). *ASCE Journal of Management*, 2019.

[5] Austin M.A., Delgoshaei P., and Nguyen A. Distributed Systems Behavior Modeling with Ontologies, Rules, and Message Passing Mechanisms. In *Thirteenth Annual Conference on Systems Engineering Research (CSER 2015)*, Hoboken, New Jersey, March 17-19 2015.

[6] Austin M.A., Mayank V., and Shmunis N. Ontology-Based Validation of Connectivity Relationships in a Home Theater System. *21st International Journal of Intelligent Systems*, 21(10):1111–1125, October 2006.

[7] Barretina J., et al. The Cancer Cell Line Encyclopedia enables Predictive Modeling of Anticancer Drug Sensitivity. *Nature*, 483, 2012.

[8] Bhargava, N., et al. Decision Tree Analysis on J48 Algorithm for Data Mining. *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering*, 3(6), 2013.

[9] Bolouri et al. Big Data Visualization identifies Multidimensional Molecular Landscape of Human Gliomas. *PNAS*, 113, 2016.

[10] Boschert S., Heinrich G., and Rosen R. Next Generation Digital Twin. In *Proceedings of TMCE, Las Palmas de Gran Canaria, Spain Edited by: Horvath I., Suarez Rivero J.P. and Hernandez Castellano P.M.*, May 7-11 2018.

[11] Ceccarelli M., Barthel F.P., Malta T.M., Abedot T.S., Salama S.R., Murray B.A., Morozova O., Newton Y., Radenbaugh A., Pagnotta S.M., Anjum S., Wang J., Manyam G., Zoppoli P., Ling S., Rao A.A., Grifford M., Cherniack A.D., Zhang H., Poisson L., Carlotti C.C., da Cunha Tirapelli D.P. Rao A., Mikkelsen T., Lau C.C., Alfred Yung W.K., Rabadan R., Huse J., Brat D.J., Lehman N.L., Barnholtz-Sloan J.S., Zheng S., Hess K., Rao G., Meyerson M., Beroukhim R., Cooper L., Akbani R., Wensch M., Haussler D., Aldape K.D., Laird P.W., Gutmann D.H., Noushmehr H., Iavarone A., and Verhaak R.G.W. Molecular Profiling Reaveals Biologically Discrete Subsets and Pathways of Progression in Diffuse Glioma. *Cell*, 164:550–563, January 2016.

[12] Coelho M., Austin M.A., and Blackburn M.R. The Data-Ontology-Rule Footing: A Building Block for Knowledge-Based Development and Event-Driven Execution of Multi-Domain Systems. In *2018 Conference on Systems Engineering Research*, Charlottsville, VA, May 8-9 2018.

[13] Delgoshaei P., and Austin M.A. Software Patterns for Traceability of Requirements to Finite-State Machine Behavior: Application to Rail Transit Systems Design and Management. In *22nd Annual International Symposium of The International Council on Systems Engineering (INCOSE 2012)*, Rome, Italy, 2012.

[14] Delgoshaei P., Austin M.A., and Pertzborn A. A Semantic Framework for Modeling and Simulation of Cyber-Physical Systems. *International Journal On Advances in Systems and Measurements*, 7(3-4):223–238, December 2014.

[15] Delgoshaei P., Austin M.A., and Veronica D.A. A Semantic Platform Infrastructure for Requirements Traceability and System Assessment. *The Ninth International Conference on Systems (ICONS 2014)*, pages 215–219, February 2014.

[16] Delgoshaei P., Heidarinejad M., and Austin M.A. Combined Ontology-Driven and Machine Learning Approach to Monitoring of Building Energy Consumption. In *2018 Building Performance Modeling Conference and SimBuild*, Chicago, IL, September 26-28 2018.

[17] Feigenbaum L., 2006. Semantic Web Technologies in the Enterprise.

[18] Ferik A. Healthcare Digital Transformation, Director, Software Platform Engineering, GE Healthcare, 2013.

[19] Francis et al. Cypher: An Evolving Query Language for Property Graphso. In *International Conference on Management of Data*, 2018.

[20] Freitas A.A., Wieser D.C. and Apweiler R. On the Importance of Comprehensible Classification Models for Protein Function Prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 7(1):172–182, 2010.

[21] Glaessgen E.H., and Stargel D.S. The Digital Twin Paradigm for Future NASA and U. S. Air Force Vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf.*, 2012.

[22] Goodenberger M.L., and Jenkins R.B. Genetics of Adult Glioma. *Cancer Genetics*, 12, 2012.

[23] Hendler J. Agents and the Semantic Web. *IEEE Intelligent Systems*, pages 30–37, March/April 2001.

[24] Johns Hopkins Medicine Health Library. Gliomas.

[25] Kaplan E.L., and Meier P. Nonparametric Estimation from Incomplete Observations. *Journal of the American Statistical Association*, 53(282):457–481, 1958.

[26] Keutzer K., Malik S., Newton A.R., Sangiovanni-Vincentelli A. System-Level Design : Orthogonalization of Concerns and Platform-Based Design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(12), December 2000.

[27] Lee J., Bagheri B., Kao H. A Cyber-Physical Systems Architecture for Industry 4.0-based Manufacturing Systems. *Manufacturing Letters*, 3:18–23, 2015.

[28] MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, 1967.

[29] Mak I.W., et al. Lost in Translation: Animal Models and Clinical Trials in Cancer Treatment. *American Journal of Translational Research*, 6, 2014.

[30] Mantel N. Evaluation of Survival Data and two new Rank Order Statistics arising in its Consideration. *Cancer Chemother Rep*, 50:163–170, 1966.

[31] McNeil C. NCI-MATCH Launch Highlights New Trial Design in Precision-Medicine Era. *JNCI: Journal of the National Cancer Institute*, 107(7), 2015.

[32] Miller J.J. Graph Database Applications and Concepts with Neo4j. In *Proceedings of the Southern Association for Information Systems Conference*, 2013.

[33] Mohammadi N., and Taylor J.E. Smart City Digital Twins, 2018 Global Smart Industry Conference (GloSIC). In *Proceedings of IEEE Symposium Series on Computational Intelligence*, pages 1–5, January-February 2018.

[34] Mosteller M., Austin M.A., Ghodssi R. and Yang S. Platforms for Engineering Experimental Biomedical Systems. *IEEE Systems Journal*, 9(4):1218–1228, December 2015.

[35] Nassar N., and Austin M.A. Model-Based Systems Engineering Design and Trade-Off Analysis with RDF Graphs. In *11th Annual Conference on Systems Engineering Research (CSER 2013)*, pages 216–225, Georgia Institute of Technology, Atlanta, GA, March 19-22 2013.

[36] NCI. About Cancer.

[37] Noy N.F., Crubézy M., Fergerson R.W., Knublauch H., Tu S.W., Vendetti J. and Musen M.A. Protégé-2000: An Open-Source Ontology-Development and Knowledge-Acquisition Environment. In *AMIA Annual Symposium Proceedings*, volume 2003, pages 953–953. American Medical Informatics Association, 2003.

[38] Ohgaki H., and Kelihues P. Population-based Studies on Incidence, Survival Rates, and Genetic Alterations in Astrocytic and Oligodendroglial Gliomas. *Journal of Neuropathology and Experimental Neurology*, 64, 2005.

[39] Ostrom et al. Primary Brain and Other Central Nervous System Tumors Diagnosed in the United States in 2009-2013. *Neuro-Oncology*, 18, 2016.

[40] Pareja-Tobes P., et al. Bio4j: A High-Performance Cloud-Enabled Graph-based Data Platform. *BioRxiv*, page 016758, 2015.

[41] Patel et al. Single-cell RNA-seq Highlights Intratumoral Heterogeneity in Primary Glioblastoma. *Science*, 344, 2014.

[42] Petnga L., and Austin M.A. An Ontological Framework for Knowledge Modeling and Decision Support in Cyber-Physical Systems. *Advanced Engineering Informatics*, 30(1):77–94, January 2016.

[43] Quinlan J.R. Improved Use of Continuous Attributes in C4.5. *Journal of Artificial Intelligence Research*, 4, 1996.

[44] Rhodes D.R., et al. . ONCOMINE: A Cancer Microarray Database and Integrated Data-Mining Platform. *Neoplasia*, 6(1):1–6, 2004.

[45] Lindsay Rowe, Tuo Dong, Terri Armstrong, Megan Mackey, Mark Gilbert, Andra Krauze, and Kevin Camphausen. A mobile app for health related quality of life and symptom assessment in patients with primary brain tumors in an outpatient oncology clinic, 2016.

[46] Rudolf G. Some Guidelines For Deciding Whether To Use A Rules Engine. 2003. Sandia National Labs.

[47] Sabbagh K. *Twenty-First Century Jet: The Making and Marketing of the Boeing 777*. Scribner Publishing, New York, New York, 1996.

[48] Sangiovanni-Vincentelli A. Automotive Electronics: Trends and Challenges. In *Presented at Convergence 2000*, Detroit, MI, October 2000.

[49] Segaran T., Taylor J. and Evans C. *Programming the Semantic Web.* O'Reilly, Beijing, 2009.

[50] Simpson T.W., Jonathan R.A., and Mistree F. Product Platform Design:Method and Application. *Research in Engineering Design*, 13:2–22, 2001.

[51] Smith B., Ceusters W., Klagges B., Kohler J., Kumar A., Lomax J., Mungall C., Neuhaus F., Rector A.L. and Rosse C. Relations in Biomedical Ontologies. *Genome Biology*, 6(5), 2005.

[52] Steiner T., et al. Adding Real-time Coverage to the Google Knowledge Graph. In *1th International Semantic Web Conference (ISWC 2012).* Citeseer, 2012.

[53] The Cancer Genome Atlas Research Network, Weinstein J.N., Collisson E.A., Mills G.B., Mills Shaw K.R., Ozenberger B.A., Ellrott K., Shmulevich I., Chris Sander C., and Stuart J.M. The Cancer Genome Atlas Pan-Cancer Analysis Project. *Nature Genetics*, 45(10):1113–1120, October 2013.

[54] Topol, E.J. Individualized Medicine from Prewomb to Tomb. *Cell*, 157:241–253, March 2014.

[55] Verzani J. *Using R for Introductory Statistics.* Chapman and Hall/CRC, 2014.

[56] Witten I.H., Frank E., Hall M.A., and Christopher J.P. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kauffmann, 2017.

[57] Zheng A., and Casari A. *Feature Engineering for Machine Learning.* O'Reilly Media, 2018.