Universidade do Minho

Escola de Engenharia

João Pedro Alves Fernandes

Probabilistic Logic Programming for Cancer Genomics

January 2018

Universidade do Minho

Escola de Engenharia

João Pedro Alves Fernandes

Probabilistic Logic Programming for Cancer Genomics

Master dissertation

Master Degree in Bioinformatics

Dissertation supervised by

Pedro G. Ferreira

Professor Rui Mendes

January 2018

DECLARAÇÃO

Nome: João Pedro Alves Fernandes

Endereço eletrónico: jopealfe@hotmail.com Telefone: 917896441

Número do Bilhete de Identidade: 14859066

Título dissertação: Probabilistic Logic Programming for Cancer Genomics

Orientador(es):
Pedro Gabriel Ferreira
Rui Mendes

Ano de conclusão: 2018

Mestrado em Bioinformática

Universidade do Minho, 26/10/2018

Assinatura: João Pedro Alves Fernandes

Acknowledgements/Agradecimentos

Como todos os trabalhos, este não foge à regra. Portanto, não seria possível finalizar a realização deste trabalho sem o grande número de pessoas que me acompanhou e apoiou ao longo de todo o ano.

Gostaria de agradecer aos meus orientadores, professor Pedro Ferreira e professor Rui Mendes pelo grande tempo prescindido para me ajudar na elaboração do trabalho. As opiniões e orientação tanto nos altos como nos baixos momentos foi, sem dúvida, um fator chave para a conclusão do trabalho.

Estou também gratificado a todos que, mesmo não estando diretamente ligados a este trabalho, dispensaram uma parte do seu tempo para me auxiliar em diversos assuntos relacionados com o tema. Esta gratidão vai com um especial endereço ao professor Rui Camacho.

Reconheço que todos os meus colegas de mestrado sempre foram um pedaço fundamental para conseguir concluir o curso. Sem a amizade, interajuda e momentos de descontração que me proporcionaram, este percurso teria sido bem mais difícil. Um obrigado especial ao Daniel Martins, à Marta Sampaio e à Beatriz Magalhães.

Um grande obrigado a todos os meus restantes amigos que estiveram sempre comigo, nos bons e piores momentos, e nunca deixaram de me apoiar e motivar.

Como é óbvio, quero deixar registado também um agradecimento especial para toda a minha família, especialmente para os meus pais e irmão. Tudo o que consegui alcançar na vida advém de tudo o que sempre me proporcionaram e do modo como me moldaram.

Quero agradecer ainda à minha namorada por ter sido sempre a minha primeira base de suporte e firmeza. Obrigado por sempre me incentivares a enfrentar desafios e exigir sempre mais e melhor dos meus objetivos.

A todos, o meu autêntico e honesto obrigado.

ABSTRACT

Over the past years, research on cancer genomics has been boosted by the advances in high throughput sequencing technologies. The Cancer Genome Atlas (TCGA) project is an effort to map the genomic alterations possibly associated with specific types of tumours and aims to improve the prevention, diagnosis and treatment of cancer. The generation of large and heterogeneous datasets, as a result of TCGA and other similar projects, creates the need to use advanced bioinformatics and computational tools for the analysis of cancer genomic data.

Despite different bioinformatics frameworks have been established in order to explore and perform comprehensive analysis of cancer datasets, the area of logic and probabilistic logic programming has not been sufficiently explored in the analysis of cancer data.

The main goal of this thesis was to explore Problog – a probabilistic logic programming (PLP) language – to encode interactions on heterogeneous cancer genomics datasets that may lead to new insights. To accomplish this objective, our work consisted in the elaboration of a python program and a Problog *framework*. The used datasets involved stomach cancer genomic data.

The python program – *ProceOmics* – aimed to process and format cancer genomic data so it could be used by Problog programs. The Problog *framework* – Problog Knowledge Base (KB) – intended to codify the data previously processed by *ProceOmics*. To evaluate the consistency of the developed framework and explore possible relations between the different types of genomic data, queries were formulated to the Problog KB.

Thus, this thesis provides a tool that establishes a link between the genomic data contained in public databases with probabilistic logic programs. We hope this work may help to overcome future efforts to use PLP on genomic data analysis.

Keywords: Cancer Genomics; Exploration; Problog; Stomach Cancer; TCGA; Data Processing

RESUMO

Ao longo dos últimos anos, devido aos avanços significativos nas áreas tecnológicas responsáveis pelo estudo do genoma humano, o estudo dos dados genómicos associados a casos de ocorrência de cancro tem crescido exponencialmente. *The Cancer Genome Atlas* (TCGA), é um projeto que consiste no mapeamento de mudanças a nível genómico que possam estar associadas com algum tipo específico de cancro e que, por sua vez, possam fornecer alternativas mais avançadas de prevenção, prognóstico e tratamento relativamente àquelas já existentes. No entanto, a geração de inúmeros e extensivos *datasets* tem, consequentemente, vindo a aumentar.

Apesar de já existir um número significativo de ferramentas e metodologias bioinformáticas que têm como objetivo explorar e realizar análises sobre os diferentes *datasets* relativos a variados tipos de cancro, a área da programação lógica, bem como da programação lógica probabilística, não têm sido frequentemente exploradas de modo a alcançar esse mesmo objetivo.

Posto isto, o objetivo principal desta tese consistiu na exploração de uma extensão probabilística de uma linguagem lógica – Problog – de modo a codificar e explorar interações complexas entre diferentes *datasets*, visando ainda a descoberta de novas relações entre eles. De modo a alcançar este objetivo, o trabalho desenvolvido consistiu na elaboração de um programa em python e de uma *framework* em Problog. Todos os dados utilizados nas análises realizadas nesta tese são relativos à genómica do cancro do estômago.

O programa em python – *ProceOmics* – teve como objetivo processar e formatar dados genómicos de cancro de modo a ser possível codificar esses mesmos dados em programas Problog. Por sua vez, a *framework* em Problog – Problog KB – foi criada com o intuito de codificar os dados previamente processados pelo programa. De modo a avaliar a consistência da *framework* desenvolvida e explorar possíveis relações entre os diferentes tipos de dados genómicos, foram colocadas queries à Problog KB.

Assim sendo, esta tese forneceu uma ferramenta que estabelece uma ligação entre os dados genómicos, contidos em base dados públicas, e programas lógico probabilísticos. Esta ligação poderá ajudar a ultrapassar os poucos esforços aplicados na utilização deste tipo de linguagem para estudar dados genómicos.

Palavras-chaves: Estudos Genómicos; Exploração; Problog; Cancro do Estômago; TCGA; Processamento de Dados

Contents

List of Figures

List of Tables

List of Abbreviations

LP – Logic Programming

AI – Artificial Intelligence

NGS – Next Generation Sequencing

WHO – World Health Organization

TCGA – The cancer Genome Atlas

NCI – National Cancer Institute

NHGRI – National Human Genome Research Institute

ICGS – International Cancer Genome Consortium

CGHub - Cancer Genomics Hub

EGA – Genome-phenome Archive

COSMIC – Catalogue Of Somatic Mutations In Cancer

CPRG – Cancer Program Resource Gateway

CGWB – Cancer Genome Work Bench

PLP – Probabilistic Logic Programming

PP – Programming Paradigms

SRL – Statistical Relational Learning

GC – Gastric Cancer

H. *pylori* – *Helicobacter pylori*

EB V - Epstein-Barr virus

TSGs – Tumor Suppressor Genes

MIN – Microsatellite Instability

CIN – Chromosomal Instability

ACRG – Asian Cancer Research Group

LOH – Loss Of Heterozygosity

Dnmts – DNA methyltransferases

SAM – S-adenyl methionine

BS – Bisulfite Sequencing

MSP – Methylation-specific PCR

Q-MSP – Quantitative Methylation-specific PCR

WGS – Whole Genome Sequencing

WES – Whole Exome Sequencing

SNVs – Single Nucleotide Variants

GEM – Gene Expression Matrix

RPKM – Reads Per Kilobase Million

FPKM –Fragments Per Kilobase Million

TPM – Transcripts per Kilobase Million

KB – Knowledge Base

NA – Non Attributed

SLPs – Stochastic Logic Programs

PHA – Probabilistic Horn abduction

1. Introduction

1.1 Context

When noxious modifications to DNA occur and the cells cannot repair or destroy them, they tend to attach definitively to the genome and a cascade of abnormal effects arise from it, leading to an improper function of an organism and its cells. A set of these events can develop a challenging, embracing and atrocious well-known genetic disease that control the way our cells function, especially how they grow and divide, the cancer (Sud, Kinnersley, & Houlston, 2017). Currently, is known that the hallmark of this genetic disease is the deregulation of gene expression profiles and disruption of molecular networks which are deployed, in most cases, by mutations and changes in DNA methylation (Sadikovic, Al-Romaih, Squire, & Zielenska, 2008).

Life expectancy increase comes at the cost of higher incidence of some diseases. A good example is the cancer incidence. The World Cancer report done in 2014 by World Health Organization (WHO) has shown the increasing from 12.7 million in 2008 to 14.1 million cancer incidences and that this numbers tend to rise a further 75%, which leads to 25 million cases panorama over the next two decades (Stewart & Wild, 2014).

There are at least 200 forms of cancer and many more subtypes. Over the past few years, a lot of efforts have been put in order to better understand the complexity of the disease by developing methods of treatment, early detection and prevention. The Cancer Genome Atlas (TGCA) (The Cancer Genome Atlas - National Human Genome Research, 2017) is a comprehensive and coordinated collaboration between the National Cancer Institute (NCI) and National Human Genome Research Institute (NHGRI) that aims is to understand the molecular basis of cancer that can be achieve through the application of genome analysis techniques like large-scale genome sequencing. Like TGCA, there are other projects in this field with the same or similar objectives, for instance, the International Cancer Genome Consortium (ICGS) (Consortium, 2017) and the Cancer Genome Project from Sanger Institute (Sanger Institute, 2017). Databases and web tools like Cancer Genomics Hub (CGHub), European Genome-phenome Archive (EGA), Catalogue Of Somatic Mutations In Cancer (COSMIC), Cancer Program Resource Gateway (CPRG), Broad's GDAC, SNP500Cancer, canEvolve, MethyCancer, SomamiR, Cancer Genome Work Bench (CGWB), among others more are also available (Yang et al., 2015). Along with the growing number of cancer projects, databases and web tools for cancer genomics, the number of collected data types in the field is increasing exponentially. However, despite the existence of few computational models to combine the different data types of cancer genomics, there is a lack of effective bioinformatics tools to centralize these different data types in an integrative analysis (Kristensen et al., 2014; Shen, Olshen, & Ladanyi, 2009).

Logic Programming (LP), uses logic and inference mechanisms to represent different information and provide logical outcomes (Kowalski, 1988).

In the early 1970, Prolog emerged as a programming tool to solve problems in artificial intelligence fields (Bratko, 1987; Kowalski, 1988). Prolog is a programming language for symbolic and non-numeric computation that is especially well suited for solving problems that involve objects and relations between objects (Bratko, 1987).

Probabilistic Logic Programming (PLP) is a probabilistic extension of LP that explores uncertainty by incrementing probabilities within the program code. This allows to infer and estimate parameters to further obtain organized probable cases instead of uncertainties (De Raedt & Kimmig, 2015).

Problog is a probabilistic extension of Prolog that defines a probabilistic distribution over logic programs by specifying, for each clause, the probability that it belongs to a randomly sampled program (De Raedt, Kimmig, & Toivonen, 2007).

Combining probability and logic for dealing with complex relational domains, such as the different cancer data types, can be a challenging and interesting study. As an example, Problog programs can be developed to learn new knowledge by encoding complex interactions in heterogeneous datasets.

1.2 Objectives

The principal aim of this thesis is to use Problog, a probabilistic logic programing language, to learn relations across data obtained from a cohort of cancer patients in multi-level genomics datasets (expression, methylation, genomic and clinical). We selected a specific cancer type, stomach cancer, to test and develop the proposed methodology.

The first goal will focus on the development of processing and formatting strategies in order to prepare unprocessed datasets. This goal aims to create a representation that provides a knowledge base for our PLP framework. Through the review of literature, additional knowledge base (KB) facts are included as domain knowledge.

Once the knowledge KB is created, the second goal of the thesis consists in generate a set of queries to allow the discovery of new insights. In particular, we are interested in developing queries that given the genomic and clinical characteristics of a patient allow evaluating its probability of having a certain cancer subtype.

More specifically, the thesis goals are:

- Retrieve cancer data from TCGA, in particular datasets that assay:
  - Gene expression.
  - Mutations.
  - Methylation.
  - Clinical.
- Develop a general workflow to subject unprocessed genomic cancer data
  - Apply different approaches in order to process and format the cancer data.
  - Create files that store the processed data.
- Create a knowledge base in Problog describing the processed data
  - Encode cancer background knowledge and processed genomic datasets into facts and rules.
- Formulate accurate questions (queries) to the knowledge base in order to discover novel relations in the data.

In the next chapter, we provide both computational and biological background for remainder of the thesis. A brief review will be provided on the three main topics of the thesis: Logic Programming, Probabilistic Logic Programming and Gastric Cancer.

2. State of Art

In the first part of this chapter we start by an introduction to logic programming and its main concepts, which will be necessary to later introduce the Problog language, the probabilistic extension of a logic programming language.

In the second part of the chapter, an overview about genomics of gastric cancer will be provided.

2.1 Logic Programming

Programming paradigms (PP) are the result of ideas about how computer programs should be constructed and can be defined as a set of coherent abstractions used to effectively model a problem/domain that allows the classification of programming languages (Gamper, 2015). There are several PP, however, four of them – Imperative, Functional, Logical and Object-oriented – can be highlighted since these four are the most common (American Dictionary Language Edition, & Merriam-webster, 2013).

Historically, logic programming as a field has its roots going back to 1915 to 1930. However, since there were no computers to run the logic procedures developed at that time, nobody was able to think in terms of programming. With the computers arrival in the early 1950's, the first attempt to encode the 25 years old logic methods was realized by Evert Beth and followed by many others. Regardless, all the efforts made through all these years in logic programming field, it was only at the beginning of the 1970's that LP had an abrupt paradigm change which was leaded by the creation of Prolog by Alain Colmerauer and Robert Kowalski (Lloyd, 1983). Although many different LP languages emerged since that decade, the current LP paradigm stands similar.

The effort to apply a universal definition to LP has generated a lot of controversy and different definitions. Kowalski defines LP shared with mathematical theorem proving as the use of logic to represent knowledge and the use of deduction to solve problems by deriving logical consequences (Kowalski, 1988). Contrariwise, Carl Hewitt defines LP broadly as "using logic to deduce computational steps from existing propositions" (Hewitt, 2008).

Logic Programming is part of the logical paradigm. A logic program is, basically, composed by a set of axioms that defines relations between objects. In this context, an axiom is a statement taken as true that serves as a starting point for further reasoning. They are represented by clauses that are either a fact or a rule (Kowalski, 1988). A detailed description of both types of clauses is provided in the next section of this chapter.

Rather than viewing a computer program as a step-by-step description of an algorithm, the program is conceived as a logical theory, and a procedure call is viewed as a theorem of which the truth needs to be established. Thus, executing a program means searching for a proof. A logic program concentrates on a declarative specification of what the problem is and differs from imperative programs since these last ones concentrates on a procedural specification of how a problem needs to be solved. Furthermore, LP is much closer to mathematical intuition than imperative programming (Sterling & Shapiro Ehud Y., 1999; Tzafestas, 1995).

The LP machine model (abstraction of the computer on which programs are executed) is not a dynamic one. Computer plus program represent a certain amount of knowledge about the world, which is used to answer queries (Tzafestas, 1995).

2.1.1 Prolog

Prolog roots emerged in the early 1971, as a result from the work of Alain Colmerauer and Robert Kowalski based in a natural language question-answering system developed at Marseille (Kowalski, 1988; White, 1989). However, the current form of this language was only obtained in the late 70s by Kowalski and colleagues in the UK (White, 1989). Actually is one of the most LP languages widely known.

The Prolog usage for solving real-world situations relies mainly in AI problems. However, its usage has a huge range of applications. Table 1 display some applications of Prolog in real cases.

Table 1 - Real-world Prolog applications.

Field	Application	Description	Reference
Decision Support Systems	"Options Trading Analysis System" (OTAS)	Stock options analyses and investment strategies.	(Tsadiras, 2009)
	"RoadWeather Pro"	Expert weather advisor	(Spreitzhofer, 1997)
Natural Language Processing	"CAT2"	Analysis, generation and translation of natural language sentences	(Sharp, 1988)
	"LMT"	Machine that performs the translation of English to German	(McCord, 1989)
Knowledge Base System	"AGATHA Electronic Diagnosis Knowledge Based System"	Test and diagnose complex printed circuit boards	(Allred, Preist, Bennett, & Gupta, 1991)
Scheduling and Planning	CAS/FPS	Multi-criteria design that provide a computer-aided synthesis of the production plans and schedules from the possible building elements	(Tsadiras, 2009)

Table 1 - Real-world Prolog applications (continuation).

Field	Application	Description	Reference
Computer Vision	GEONS	Recognize the class of a 3-D volumetric object in an image	(Dickson, 2003)
Game Playing	Chess	Prediction of moves	(Bain, 1994)

The Prolog mechanics includes pattern matching, tree-based data structuring and automatic backtracking. This small set of basic mechanisms makes Prolog especially well suited for problems that involve objects and relations between them (Bratko, 1987). A Prolog program is written as a set of facts and rules defining the relationships between data items (Predicates, 2001).

Prolog Syntax

The Prolog system recognizes the type of an object in the program by its syntactic form. This is possible because the syntax of Prolog specifies different forms for each type of data objects and no additional information (such as data-type declaration) has to be communicated to Prolog in order to recognize the type of an object (Bratko, 1987).

Terms

The basic structure in Prolog is a term. There are four kinds of terms: atoms, numbers, variables and compound terms (Endriss, 2016; Predicates, 2001).

Atoms. May be strings, starting with a lowercase letter, made up of lower and uppercase letters, digits, the underscore, any series of arbitrary characters enclosed in single quotes and strings made up solely of special characters like `+ - * = < > : &`.

Numbers. Integers or floats.

Variables. Start with a capital letter or an underscore. Represented as strings of letters, digits and underscore. They are unbound values that will later be attached to data.

The underscore may be used and constitutes a special case, which is named anonymous variable. This is the only variable where different occurrences represent different variables.

Compound terms. Made up by a functor – a Prolog atom – and a number of arguments – Prolog terms like atoms, numbers, variables or other compound terms – enclosed in parenthesis and separated by commas. A set of compound terms and atoms together form the set of Prolog predicates and a term that does not contain any variables is called a ground term.

Clauses and Queries

Prolog programs are made up by facts and rules. Facts and rules are also called clauses, which are axioms. A sequence of clauses is a Prolog program. Prolog programs are encoded in a knowledge base where queries are submitted in order to retrieve information from it.

Facts. The intuitive meaning of a fact is that we define a certain instance of a relation as being true. A fact must start with a predicate, which is an atom and end with a full stop. The predicate may be followed by one or more arguments, enclosed by parentheses. Those arguments are separated by commas and may be atoms, variables or numbers.

Rules. Consists of a head (predicate) and a body (sequence of predicates separated by commas). The head and the body are separated by the sign `:-` which represents a condition (if). Like every Prolog expression, has to be terminated by a full stop. The intuitive meaning of a rule is that the goal expressed by its head is true if the Prolog system can demonstrate that all the subgoals in the rule's body are true.

Queries. Represent statements starting with a predicate and followed by arguments. The predicate must have appeared in at least one fact or rule of the program. Usually are entered at the Prolog prompt. When a query is submitted, Prolog tries to verify if all the query predicates are true.

Table 2 exemplifies each simple Prolog syntax structure with examples.

Table 2- Basic Prolog syntax examples.

Basic Constructs		Examples
Terms	Atoms	banana b acbXYZ y_333 hello_world_again + <----> ***
	Numbers	-2 -1 0 3 16.403
	Variables	X Banana _333 X_1 MyVariable _
	Compound terms	date(1, may, 1983) point(X,Y,Z) ‘My Functor’(animal)
Clauses	Facts	parent(pam, bob). smokes(someone).
	Rules	offspring(Y,X) :- parent(X,Y). is_smaler(X,Y) :- is_bigger(Y,X).
Queries		?- offspring(liz,tom). ?-is_bigger(elephant, donkey). ?-small(X), green(X), slimy

2.2 Probabilistic Logic Programming

Probabilistic Logic Programs are logic programs in which facts are annotated with probabilities (Fierens, Van Den Broeck, & Renkens, 2013). Emerged from AI, PLPs principal goal is to extend logic programming languages with primitives to support probabilistic inference and learning.

The number of probabilistic logics has been increasing significantly with PRISM, PHA, SLPs pD and MLNs being the most used ones (De Raedt et al., 2007; Dries et al., 2015).

The major goal of this programming paradigm is to provide efficient tools for modelling and reasoning about uncertain domains that can arise from several different fields. Besides the resemblance with statistical relational learning (SRL), the art of probabilistic programming is more focussed on a programming language perspective rather than on a graphical model one (Dries et al., 2015).

2.2.1 Problog

The Problog language is a probabilistic programming language derived and extended from Prolog along the lines of Sato's distribution semantics (Dries et al., 2015). Such semantics defines the join of probabilistic choices with a logic program results in a distribution over possible worlds (DTAI, 2015). As De Raedt and colleagues (De Raedt et al., 2007) defines "Problog is the simplest probabilistic extension of Prolog one can design.".

Problog allows the user reasoning with relational data, parameters learning and dealing with uncertainty. These Problog characteristics allows Problog programs to encode complex interactions between large sets of heterogeneous components as well the uncertainties that are exclusively related to real-life events (Dries et al., 2015).

In order to perform its fundamental task, which is the efficient computation of a query's successes probability, Problog apply different inference methods and employs several state-of-the-art technologies (Mantadelis & Rocha, 2017).

From a modelling perspective, Problog programs have two different parts:

I. Probabilistic part

II. Logical part

Part I defines a probability distribution over truth-values of a subset of the program's atoms and part II derives truth-values of remaining atoms using a reasoning mechanism similar to Prolog. (Dries et al., 2015).

In terms of usage, Problog has not been quite explored and used for solving real-world biological issues. Only few studies on the application of this language to biological data have been published. Some of Problog studies include:

- Ong & Lewis (Ong & Lewis, 2012) introduced logic-based regulation models in order to prove that network hypothesis can be generated from existing gene expression data.
- Perez-Iratxeta (Perez-Iratxeta, Bork, & Andrade, 2002) developed a scoring system for the relationship of human genes to 445 inherited diseases from certain chromosomal regions that have not been associated with any particular gene.

Problog Syntax

The fundamental difference between Prolog and Problog is that Problog supports probabilistic predicates. Alongside with the Prolog syntax, Problog introduces an additional operator **::** and two predicates (**query** and **evidence**).

The **query** predicate enables and represents the inference task that the user may want specify according to the goal. The **evidence** predicate allows specifying atoms that are known to be true or false.

The :: operator allows to associate a probability P to a certain fact considered truth, i.e. P :: fact. The association of probabilities to facts may result in two distinct types of facts:

- **Non-probabilistic facts** are not associated to probabilities, thus, considered as completely true. They can be represented without the variable **P** and the additional operator **::**. However, these are easily transferred to probabilistic facts through the association of the variable *P* the value of 1.0. Both forms can be ground or not-grounded facts. The former one is related to cases when all variables are instantiated (assume a constant value) and the later is related to cases where not all the variables are instantiated.
  As an example, **father(X,Y).** is a non-probabilistic and not-grounded fact which is equivalent to its probabilistic form **1.0::father(X,Y)**.
- **Probabilistic facts** are divided in two forms: probabilistic facts and intensional probability facts. The former fact form is **Pi::Fi** which can be translated as the probability of grounded *Fi* representing random events. The true assignment is *Pi*. The later has the following form: **P::f(X1,X2,…,Xn) :- body**, where *body* is a conjunction of probabilistic and non-probabilistic facts defining the domain of variable *X1, X2, …, Xn* which can be translated into a set of facts with the same probability *P* if the body is true.

> However, intensional probabilistic facts have a particular characteristic. Due to Problog2 engine, intensional probability facts support flexible probabilities. This means that the probability is not pre-specified but it is an arithmetic expression that needs to be computed.

A simple Problog program example that aims to provide a general overview of the language syntax is presented at appendix I.

2.3 Gastric Cancer

Gastric cancer is one of the most common cancers worldwide. Although, due to the recognition of certain risk factors such as *H. Pylori* and other dietary and environmental risks, the GC incidence has declined rapidly over the recent decades. Alongside with additional stomach cancer features, such as epidemiology, pathology and etiology, the breakthroughs on GC genetic and epigenetic alterations considered genomic instability as the major trace for the GC development.

Several reports that use different techniques to study GC have been created in order to obtain a better understanding about its genomics, i.e. The Cancer Genome Atlas (TCGA). In the mentioned report, a new molecular stomach cancer classification was established. This new classification model, divided the stomach cancer into four subtypes: EBV, MSI, GS and CIN.

The genomic stomach cancer datasets involve three main types of data: genetic expression, mutations and methylation. Gene expression measures the activity/expression of a set of genes to create a global picture of cellular function. Mutations are characterized by permanent alterations of the nucleotide sequence. DNA methylation is a process where methyl groups are added to DNA, which can change the activity of the respective DNA segment without changing its sequence. All this three genomic events are stored respectively in genetic expression datasets, mutations datasets and methylation datasets, respectively.

The next topics of this section are intended to provide a broad understanding about the stomach cancer. The aims of these topics are:

- Provide a general overview about GC;
- Describe the basic mechanisms that are inherent to GC;
- Describe the genomic data on GC used in this thesis;

2.3.1 General Overview

Gastric cancer has been classified as one of the most common deadliest cancer worldwide (Qu, Dang, & Hou, 2013). Like many other cancer types, most of the GC cases are adenocarcinomas. It can be divided and classified in different subtypes and stages, with variable clinical utility (Bass et al., 2014).

Epidemiology

The high rates of GC incidence are comprehended between 60 to 80 years old individuals with a male preponderance. However, amplified geographic distribution studies shows that this age range is not standard for all countries and regions. India is the best example where the age range is much lower (35 to 55 years). Although the GC incidence in individuals younger than 30 years old are highly rare, they also occur (Nagini, 2012; Stewart & Wild, 2014).

Countries that have poor systems of dietary patterns, food storages and control of *Helicobacter Pylori* are the most likely to have high incidence and mortality rates. Unlike North America and Africa, Eastern Asia, Eastern Europe and South America are the documented regions with higher GC incidence rates (Nagini, 2012).

Pathology

The stomach cancer cases are, approximately 95%, epithelial in origin and designated adenocarcinomas. Any malignant neoplasm that arises from the region extending between the gastroesophageal junction and the pylorus are considered a case of stomach cancer (Nagini, 2012).

The World Health Organization and the Lauren classification systems are the two major histological classification systems that describes the types of gastric cancer. The system derived from the World Health Organization divides gastric cancer into papillary, tubular, mucinous and poorly cohesive carcinomas. The Lauren classification system divides gastric cancers into intestinal and diffuse types (Bass et al., 2014).

Intestinal GC types are most common in men and with a better prognosis. It arises from precancerous lesions and can be influenced by environmental factors such as *H. Pylori* infections, obesity, among other factors. Diffuse GC types are most frequent in women and younger individuals. It is related to endemic areas which suggests a genetic susceptibility since it is associated with blood group A (Nagini, 2012).

Etiology

GC etiology is considered a multifactorial and multistep process accompanied by accumulation of alterations of critical growth regulatory genes (Wu et al., 2005). The major cases are attributed to *H. Pylori*, genetic, lifestyle, diet and socioeconomic factors (Nagini, 2012). The majority of GCs are associated with infectious agents that vary across the globe. H. *Pylori* and Epstein-Barr virus (EB V) are the most common infectious agents (Bass et al., 2014).

Although these factors contribute in a large scale for the etiology of GC, its development is a complex and progressive process deep-seated to genetic and epigenetic alterations.

Genetic and Epigenetic alterations

Genetic and epigenetic alterations represent an extreme important role concerned to GC. Alongside all etiological factors, they are associated to GC episodes and can occur in oncogenes, tumor suppressor genes (TSGs), DNA repair genes, cell cycle regulators and signalling molecules. The major trace of GC is the genomic instability which could be either microsatellite instability (MSI) or chromosomal instability (CIN) (Nagini, 2012).

- MSI. Results from errors in DNA replication and represent 15-20 per cent of GC with a higher incidence in familial cases.
- CIN. Manifests as a gain or loss of aneuploidy or parts of chromosomes. It is the most common instability in sporadic GC cases.
- Oncogenes. Activation or amplification of several oncogenes through mutations in oncogenes may enhance the development of GC cases. In example, mutations at the codon-12 of the K-ras oncogene was found (Nagini, 2012).
- Tumor Suppressor Genes. Inactivation, deletions and hyper methylations of TSGs lead to aggressive progress of GC. In example, the p53 gene is frequently inactivated in gastric carcinomas as well in precursor lesions by LOH, missense mutations or frameshifts deletions (Yamashita, Sakuramoto, & Watanabe, 2011).
- Cell Cycle Regulators. Over and aberrational expression and downregulation of cell cycle regulators influence negatively the prognosis of GC types. In example, overexpression of cyclin E and CDK together with aberrant p53 expression and downregulation of p27 is a common event in gastric cancer (Bani-hani, Almasri, & Khader, 2005).

2.3.2 TCGA Report

Although all the efforts put in order to facilitate and upgrade the diagnostic of GC, this type of cancer is often diagnosed at an advanced stage and the prognosis is still poor (Qu et al., 2013). Therefore, a classification based only on histology is not totally able to distinguish different molecular subtypes and, as a consequence, is not possible to develop a targeted therapy in cases of GC. Hence, the need for a new classifier based on molecular biology is needed (Lin, Wu, Guo, & Li, 2015). In order to better understand the genomics of gastric cancer, two huge significant seminal reports that used Next-Generation-Sequencing (NGS) technique were developed. One report was done by The Cancer Genome Atlas (TCGA) and the other by the Asian Cancer Research Group (ACRG) (Katona & Rustgi, 2017).

The TCGA study evaluated 295 treatment-naïve primary gastric adenocarcinomas from multiple centers where several analysis including copy number analysis, whole-exome sequencing, DNA methylation and RNA analysis, microsatellite instability testing and, for a specific selected group of tumours, whole genome sequencing, were developed (Katona & Rustgi, 2017). The goal of this study was to develop a robust molecular classification of GC. A more detailed process about the methods and analyses perform in the TCGA study, is present in this report paper (Bass et al., 2014).

The final result of this report was a new molecular classification that divides GC into four major subtypes:

- EBV – infected tumours by the Epstein Virus;
- MSI – tumours that show Microsatellite instability;
- GS – genomically stable tumours;
- CIN – chromosomally unstable tumours.

2.3.2.1 Important TCGA data of GC

According to the thesis goals, it is important to have a larger knowledge about the following 3 types of TCGA genomic data referent to gastric cancer, which are:

- Mutations data
- Gene expression data
- DNA methylation data

Mutations

The DNA sequence of a gene can be altered in a well-defined number of ways which can have a large spectrum of results. The impact of the mutations depend on where they occur within the gene and whether they alter the function of essential genes (Stewart & Wild, 2014).

The advances in Next Generation Sequencing (NGS) as well in bioinformatics and computational tools have provided an abrupt development of several approaches to identify both new and known somatic mutations (Hsu, Hsiao, Kao, Chang, & Shieh, 2017; Illumina, 2015b). The genomic technique used for sequencing the majority of TCGA samples was the Whole Exome Sequencing (WES), which is a very cost-effective alternative to Whole Genome Sequencing (WGS). Differently from WGS technique, WES consists on sequencing only the coding genome regions which, is known as the exome (Magi et al., 2014). Studies like Magi and colleagues, (Magi et al., 2014), Przytycki and Singh (Przytycki & Singh, 2017) and Hsu and colleagues (Hsu et al., 2017) support the conclusion that WES is a very useful technique to discover and identify somatic mutations that can be common and rare single nucleotide variants (SNVs), small indels and breakpoints of structural variation.

There are 7 common types of mutations, represented in Figure 1 retrieved from U.S. National Library of Medicine. These types of mutations can occur in the process of development and maintenance of any kind of cancer.

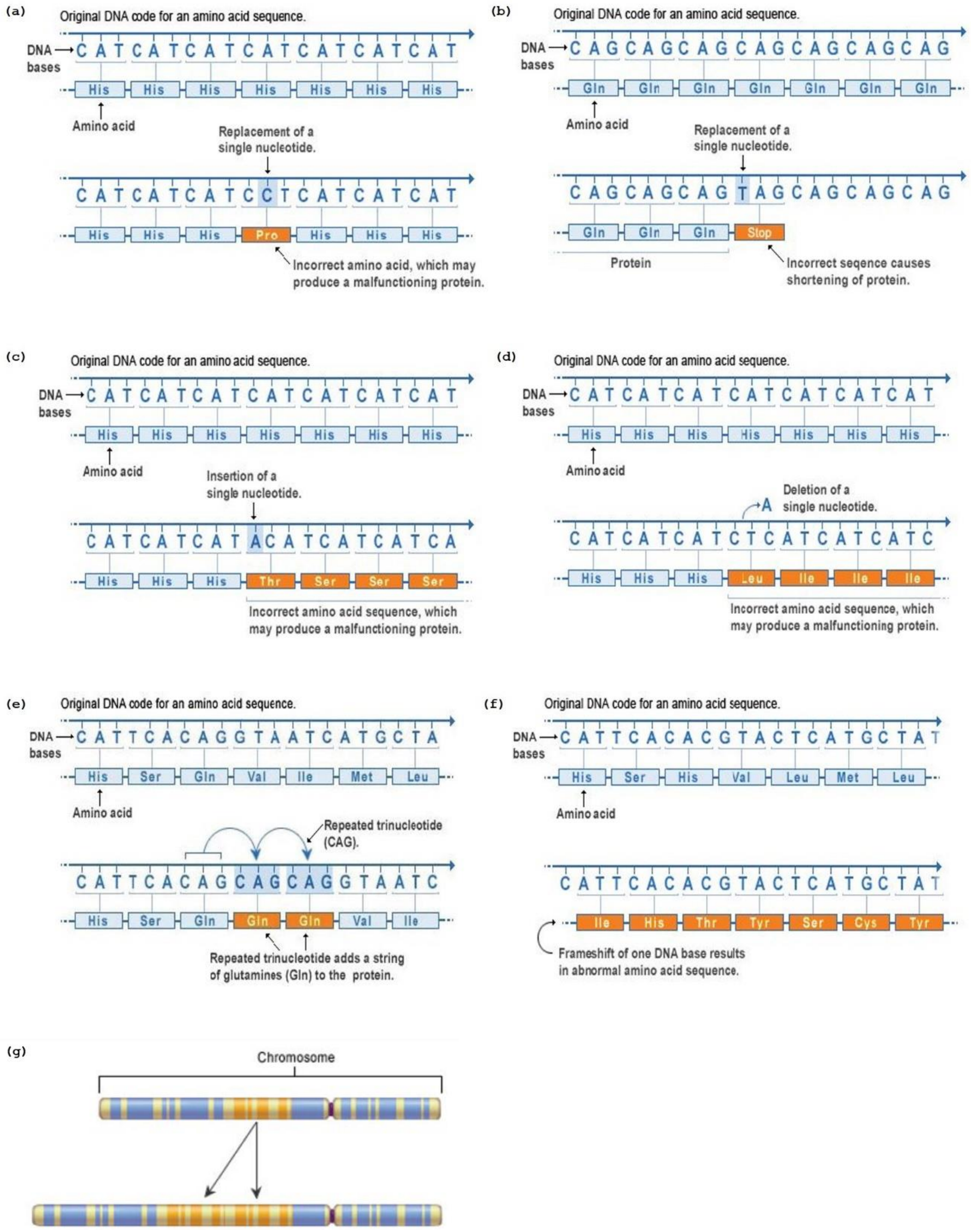

(a)
Original DNA code for an amino acid sequence.
DNA bases
C A T C A T C A T C A T C A T C A T C A T
His His His His His His His
Amino acid
Replacement of a single nucleotide.
C A T C A T C A T C C T C A T C A T C A T
His His His Pro His His His
Incorrect amino acid, which may produce a malfunctioning protein.
(b)
Original DNA code for an amino acid sequence.
DNA bases
C A G C A G C A G C A G C A G C A G C A G
Gln Gln Gln Gln Gln Gln Gln
Amino acid
Replacement of a single nucleotide.
C A G C A G C A G T A G C A G C A G C A G
Gln Gln Gln Stop
Protein
Incorrect seqence causes shortening of protein.
(c)
Original DNA code for an amino acid sequence.
DNA bases
C A T C A T C A T C A T C A T C A T C A T
His His His His His His His
Amino acid
Insertion of a single nucleotide.
C A T C A T C A T A C A T C A T C A T C A
His His His Thr Ser Ser Ser
Incorrect amino acid sequence, which may produce a malfunctioning protein.
(d)
Original DNA code for an amino acid sequence.
DNA bases
C A T C A T C A T C A T C A T C A T C A T
His His His His His His His
Amino acid
A
Deletion of a single nucleotide.
C A T C A T C A T C T C A T C A T C A T C
His His His Leu Ile Ile Ile
Incorrect amino acid sequence, which may produce a malfunctioning protein.
(e)
Original DNA code for an amino acid sequence.
DNA bases
C A T T C A C A G G T A A T C A T G C T A
His Ser Gln Val Ile Met Leu
Amino acid
Repeated trinucleotide (CAG).
C A T T C A C A G C A G C A G G T A A T C
His Ser Gln Gln Gln Val Ile
Repeated trinucleotide adds a string of glutamines (Gln) to the protein.
(f)
Original DNA code for an amino acid sequence.
DNA bases
C A T T C A C A C G T A C T C A T G C T A T
His Ser His Val Leu Met Leu
Amino acid
C A T T C A C A C G T A C T C A T G C T A T
Ile His Thr Tyr Ser Cys Tyr
Frameshift of one DNA base results in abnormal amino acid sequence.
(g)
Chromosome

Figure 1 - Seven types of common mutations. (a) missense mutation; (b) nonsense mutation; (c) insertion mutation; (d) deletion mutation; (e) repeated expansion mutation; (f) frameshift mutation; (g) duplication mutation. All the represent examples were retrieved and adapted from U.S. National Library of Medicine.

In the review of gene mutations in GC wrote by Lin and colleagues (Lin et al., 2015), the authors emphasize the need to full understand and learn the genetic composition of GC and construct a classifier that can guide clinical decisions. In order to overcome that needs that they appoint on the paper, the authors focused on newly discovered potential driver genes mutation in GC along with a short introduction to some establish and well-known driver mutations present at the TCGA database.

Genetic Expression

Gene expression is the process by which the genetic code – the nucleotide sequence – of a gene is used to direct protein synthesis. Expressed genes include genes that are transcribed into mRNA and further translated into protein, as well as genes that are transcribed into RNA, such as transfer and ribosomal RNAs, but not translated into protein (Morange, 1999).

The amount of mRNA produced during transcription provides a measure of the activity level of genes. Gene expression data reflects the disease state. These measurements are performed with microarrays and high-throughput sequencing of RNA. More concretely, these technologies are used to identify the gene expression levels between different experiments and/or to identify similarly expressed genes over multiple experiments. The results are represented in a gene expression matrix (GEM) that is structured with genes in the rows, the experiments in the columns and each cell in the matrix represents a normalized gene expression value (Fakoor, Ladhak, Nazi, & Huber, 2013).

The gene expression is represented in a quantitative way through Reads Per Kilobase Million (RPKM), Fragments Per Kilobase Million (FPKM) or Transcripts Per Kilobase Million (TPM) measures. All these three measures attempt to correct sequencing depth and feature length. To compare the expression levels of a transcript across runs, the count of reads must be normalized (Soneson, Love, & Robinson, 2016).

RPKM and FPKM are commonly used measures of gene expression. However, they present some limitations. To overcome some of the limitations of these two measures, the TPM measure has been proposed and now has been widely used. The TCGA gene expression data is available in a GEM matrix with the gene expression values derived from a FPKM formula (1).

$$FPKM_i = \frac{X_i}{\left(\frac{\tilde{l}_i}{10^3}\right)\left(\frac{N}{10^6}\right)} = \frac{X_i}{\tilde{l}_i N} \cdot 10^9 \qquad (1)$$

X_i – number of counts (number of reads that align to a particular feature)
$\tilde{l}_i$ - effective length (number of possible start sites a feature could have generated a fragment of that particular length)
N – total number of reads sequenced

DNA methylation is a major epigenetic factor that influences gene activities. It is catalysed by a family of DNA methyltransferases (Dnmts) which function consists in perform a transfer of a methyl group from S-adenyl methionine (SAM) to the fifth carbon of cytosine residue. This transfer results in 5mC which is a normal cytosine nucleotide that has been modified by the addition of a methyl group (Moore, Le, & Fan, 2013). Dnmt1 gene codify Dnmt enzyme that is involved in maintaining the methylation during the DNA replication while Dnmt3a and Dnmt3b genes codify the enzymes involved in *the novo* methylation processes to unmodified DNA strands (Moore et al., 2013; Qu et al., 2013).

It is also known that DNA methylation in different genomic regions may exert different influences on gene activities based on the underlying genetic sequence (Moore et al., 2013; Qu et al., 2013). In general, increased methylation profiles in the promoter region of a gene results in a reduced gene expression which leads to the conclusion that the occurrence of methylation in the transcribed region can have a variable effect on gene expression (Qu et al., 2013).

There are three different genomic regions were DNA methylation is more prominent:

- Intergenic Regions (DNA sequences regions located between genes, i.e. regions that do not code for genes).
- CpG Islands (regions that have higher CpG density which frequently are not methylated when compared to the rest of the genome).
- Gene Body (region between the first and last exon; methylation of those regions are related to variable levels of gene expression).

The initiation and progression of GC cases are extremely affected by promoter methylation and an aberrant methylation of a number of genes with different functions is significantly associated with the pathology of this types of cancer (Qu et al., 2013). Particularly, the methylated region that is often associated with cancer cells is CpG Islands (Siegmund & Laird, 2002).

Of the different technologies to assay methylation status we will focus on those based on DNA Microarrays. This technology remains universally used because the low-cost, high-throughput nature and the possibility to work alongside other techniques (Meaburn & Schulz, 2012). Among the many chips used for the performance of this technique, the Infinium HumanMethylation450K Bead Chip array by Illumina is the most common. This array technique pursuit differentially methylated specific regions, specifically gene promoter regions, and measure DNA methylation using a quantitative "genotyping" of bisulfite-converted genomic DNA (Illumina, 2015a). The integrated software analysis from this technique display valuable information such as chromosomal coordinates, GC percent, location in a CpG Island and the methylation β-values (Kurdyukov & Bullock, 2016).

Methylation β-value is the current unit of methylation level measurement and it is the recommended method by Illumina. Since, as mentioned above, DNA methylation arrays are usually restricted to the comparison between methylated and unmethylated CpG, Illumina Infinium assay resort a pair of probes – one methylated and the other unmethylated – which measure the methylation level through an accurate method – β-value (Du et al., 2010). The β-value is calculated as the ratio of the methylated probe intensity and the overall intensity through the formula (2):

$$Beta_i = \frac{max(y_{i,methy},0)}{\max(y_{i,unmethy},0)+max(y_{i,methy},0)+\alpha} \qquad (2)$$

$y_{i,methy}$- intensity measured by the i^{th} methylated probe

$y_{i,unmethy}$ - intensity measured by the i^{th} unmethylated probe

α – constant offset (100 by default)

The α constant is recommended by Illumina in order to regularize the β-value in cases where both probes intensities (methylated and unmethylated) are low. As a result, a number ranging from 0 to 1 is the output. To interpret the results, is necessary to understand that, generally, a 0 value indicates that no methylated molecules occur in the respective CpG site and a value of 1 indicates that every copy of the CpG site was methylated (Du et al., 2010).

3. Methods

3.1 Data

In order to create a Problog knowledge base (KB), which codes all different types of cancer genomic data to further management and exploration, it is first necessary to download all the respective datasets. The relevant data have their origins on TCGA studies and was retrieved from TCGA Data Portal (TCGA Data Portal, 2017) and cBioPortal (Cerami et al., 2014; Gao et al., 2013).

The downloaded datasets must be properly organized as text files into the different types of genomic data. The relevant cancer genomic data covers mutations, gene expression, methylation and clinical data.

In this project we will focus on the stomach cancer genomic data. Given that the supervisor is part of a group dedicated to research on stomach cancer, the choice of this cancer model allows to take advantage of the expertise in this model and to be able to discuss results with other experts within the group.

The mutations and metadata datasets were retrieved from cBioPortal Data Sets menu, named as Stomach Adenocarcinoma (TCGA, Nature 20014). All the others datasets were retrieved from the TCGA Data Portal repository through a created manifest that contain the data for the download of the required datasets.

All the downloaded datasets are described in the next section of this chapter. Most of them are very large text files. The full use of these original datasets would originate a very large Problog KB. Therefore, in order to facilitate the KB construction, all the cancer genomic data needs to be loaded as processed *csv* files to form the facts of the Problog knowledge base.

In order to yield and provide a filtered, selected and concise dataset, which can then be loaded to the Problog KB, we have developed an oriented pre-processing and formatting program. The Problog KB and the previously mentioned program are further described in the next chapter.

3.2 Datasets

All datasets can be categorized into one of the following four categories: metadata, gene expression, mutations and methylation.

It is possible that some categories may have more than one file containing different information, which is case of metadata and methylation classes for the stomach cancer.

In order to efficiently run the pre-processing program, the different datasets must follow a precise structure. Those requirements are described within this section and represented in Table 3.

Table 3 - Required structure of downloaded datasets.

Data Type	File Name	Columns	
		Must have	May have
Metadata	data_clinical_patient.txt	PATIENT_ID * AGE GENDER RACE	ETHNICITY COUNTRY LAUREN_CLASS
	data_clinical_sample.txt	PATIENT_ID * CANCER_SUBTYPE or MOLECLAR_SUBTYPE	TNMSTAGE
Mutations	data_mutations_extended.txt	Hugo_Symbol Gene Variant_Classification Tumor_Sample_Barcode *,**	-
Methylation	*files names differ*	Beta_value gene_symbol submitter_id	-
Genetic Expression	*file name may differ*	Tumor_Sample_Barcode *,**	-

* the PATIENT_ID and Tumor_Sample_Barcode columns contains the same information since the second column is a copy of the first with an additional version number. All the versions numbers were truncated to transform those different columns as the same one, which facilitates the further steps applied.

** genetic expression matrix must have the Tumor_Sample_Barcode in the columns and the gene ensembl ID in the rows.

3.2.1 Metadata

The metadata is defined as data that provide information on other data. It is useful to summarize basic information about data.

Two datasets concerning stomach cancer metadata were downloaded, which are:

- *Data_clinical_patient.txt* (1)
- *Data_clinical_sample.txt* (2)

The metadata dataset (1) contains additional information about the patients, such as their age, gender, among others.

Table 4 – Representation of the metadata dataset (1).

Note that only the relevant features were represented.

Most Important Features	Data Type	Feature Example
PATIENT_ID	String	TCGA-B7-5816-01 TCGA-B7-5818-01
AGE	Numeric	51.19 62.4
GENDER	String	FEMALE MALE
RACE	*String*	WHITE WHITE

The metadata dataset (2) contains information about the samples, such as the respective cancer subtype, cancer stage, among others.

Table 5 – Representation of the metadata dataset (2).

Note that only the relevant features were represented.

Most Important Features	Data Type	Feature Example
SAMPLE_ID	String	TCGA-B7-5816-01 TCGA-B7-5818-01
MOLECULAR_SUBTYPE	String	MSI EBV
TNMSTAGE	String	Stage_IIB Stage_IB

3.2.2 Mutations Data

Typically, the TCGA mutations datasets contains mutations identified by whole exome sequencing. The genomic somatic mutations dataset for stomach cancer has the structure described in table 6.

- *data_mutations_extended.txt* *(3)*

The mutations dataset (3) stores several columns with information regarding the occurred mutations registered by sample. However, the focused features were T*umor_Sample_Barcode*, *Hugo_Symbol*, *Variant_Classification* and *Gene*.

Table 6 – Representation of the mutations dataset (*data_mutations_extended.txt*).

All these features may have duplicates values and the first dataset row is often a version number, which is necessary to be manually removed. In addition, all of them are string data type and may contain missing data.

Most Important Features	Data Type	Feature Example
Tumor_Sample_Barcode	String	TCGA-B7-5816-01-Tumor-SM-1V6U3
Variant_Classification	String	Missense_Mutation
Hugo_Symbol	String	KLHL17
Gene	String	ENSG00000187961

The *Tumor_Sample_Barcode* column represent the samples subjected to the WES technique. All samples have an associated code, i.e. '*Tumor-SM-1V6U3'*.

The *Variant_Classification* column displays the type of somatic variant that is related to the impact of a DNA change. This is a very relevant feature since it can be used in the Problog KB as evidence in order to discard silent variants.

Each row of *Gene* and *Hugo_Symbol* columns are referent, respectively, to the ensembl gene ID and the common gene name.

3.2.3 Gene Expression Data

As mentioned in the previous chapter, FPKM is a measure of quantification of gene expression, i.e. the abundance of the copies of a certain mRNA molecule in the cell. The measure is normalized to allow intra and inter sample comparison. It values range from [0, +Inf[, typically following a power-law distribution per sample.

The stomach cancer gene expression dataset has the structure described in table 7.

- *STAD.fpkm.txt* **(4)**

Table 7 – Representation of the gene expression dataset (*STAD.fpkm.txt).*

The FPKM values must be a float type and should not exist any missing value.

Samples Genes	TCGA-CG-4462-01	TCGA-VQ-A8P3-01	TCGA-D7-A6EV-01
ENSG00000000003.13	4.75746614662	4.8661575768	4.45552240925
ENSG00000000005.5	0	0	0
ENSG00000000419.11	24.1066334708	28.7683191163	66.5659335117

The gene expression dataset (4) contains the FPKM values for each gene within a sample identifier in a GEM format. Gene ensemble IDs are displayed in the GEM rows and the various samples IDs are displayed in the GEM columns. Each row and column is attached to only one gene and sample, respectively. The cells of the matrix contain the FPKM values, which characterize the expression level of the particular gene in the particular sample.

3.2.4 Methylation Data

The methylation data is contained in different text files. Each file corresponds to the methylation data of a single gene. Therefore, all the files needed to be combined in a single dataset.

All the files store several columns with information regarding the occurred methylation level registered by sample. However, the focused features were *gene_symbol*, *submitter_id* and *Beta_value*.

For a better illustration of the methylation file structure and contents, Table 8 displays an example for the relevant features of the TP53 methylation data file.

Table 8 – Representation of the TP53 methylation dataset.

Most Important Features	Data Type	Feature Example
gene_symbol	String	TP53
submitter_id	String	TCGA-BR-8284-01
Beta_value	Numeric	0.613699241142926

The *gene_symbol* column displays the respective gene in a determined probe. Each probe may have more than one occurrence of the same gene. In this case, a semicolon must separate the different gene names.

The *Beta_value* column stores the β-value associated to genes present in the respective probe. Represented as the ratio of the methylated probe intensity and the overall intensity, these are float values that range from [0.0, 1.0].

The *submitter_id* column maps the β-value of the genes to their respective patient identifier.

3.3 Genomic Data Arrangement

All the different types of stomach cancer datasets were primarily processed via individual python and R scripts. However, we soon realized the importance and relevance of having a structured and streamlined workflow that can be applied to all genomic datasets. This would allow applying to other datasets from other types of cancer. Thus, a substantial effort of the thesis was dedicated to the development of a comprehensive and easy-to-use python program. The program was named as *ProceOmics*.

ProceOmics aims to process and format cancer genomic datasets which are later used by probabilistic logic knowledge-bases. In order to achieve this goal, the program relies on data dimensionality reduction, feature selection, data processing and data structuring operations resulting in the creation of several output files.

Data Dimensionality Reduction

Due to their large size, the load of the original datasets into the Problog KB can be prohibitive. Therefore, it is crucial to perform data dimensionality reduction on large genomic datasets. *ProceOmics* always performs a data dimensionality reduction based on the same approach. This method reduces datasets that contains the *Hugo_Symbol* and/or *Gene (Ensembl ID)* features. Its engine consists in selecting the data that contains a gene with a probability of being mutated in, at least, one of the four cancer subtypes, above a pre-defined threshold. The threshold may assume any value ranging from 0.0 to 1.0 and is provided by the user.

Note that this operation is performed on the mutation datasets. However, the program creates a *csv* file that stores the most mutated genes for a given threshold. The list of genes that have been selected in the previous step are then used to reduce the gene expression and methylation datasets.

Feature Selection

Using all the original datasets features may result in an extensive, redundant and unstructured Problog KB. Some of these features will not be considered. Therefore, it is critical to perform a feature selection process that filters out all the non-relevant information and only selects the features that were considered to be the most important from the cancer genomic datasets. *ProceOmics* automatically performs this process of feature selection for each dataset. The selected features from the different types of dataset are mentioned in the section 3.2 of this chapter.

Data Processing

The data processing operations handle the existence of features with values in a range that may cause problems when loading to the Problog KB. Furthermore, since we are working with a probabilistic logic programming framework, the probabilities of certain genomic events in the genomic datasets need to be estimated.

All data processing operations consist in a sequence of data processing steps that are applied to specific features of the dataset. The applied distinct data processing operations are dependent on the type of dataset and its respective features. *ProceOmics* data processing steps among others include, replacement or withdraw of omitted values, string manipulation, probabilities calculi or duplicate removal. All the applied data processing steps are displayed at Table 12 in appendix III.

Data Structuring

In order to avoid conflicts related to the loading of the genomic datasets in the Problog KB, the datasets must follow a specific structure. All datasets must contain a header and a comma that separate all the available features. After applying a processing operation on a genomic dataset, *ProceOmics* organizes the respective dataset in conformity to the required structure. This structuring procedure is always performed along with data processing operations.

File Creation

Problog can use various sources to collect pre-established facts. Therefore, we can use this Problog property to exchange knowledge between the genomic datasets and our Problog KB. *Csv* files are readable by Problog. Thus, after *ProceOmics* performs data processing and structuring operation on a dataset, the resulting processed and well-structured dataset is stored in a comma-separated values (*csv*) file. The number of *csv* files created by the program is always dependent on the types of data to be handled selected by the user.

3.4 Problog KB

How to Use

In order to use the Problog KB, we call the predicate *query* on the knowledge base code and run the script in command line environment, which returns the query results. Any time we want to query the knowledge base, the – ***>> <ProblogDirectory> problog scriptname.txt*** – command is executed. However, Problog also provides a shell within the command line environment that allows to interactively query the knowledge base.

How to Load Data

All the resulting files from the processing program have a *csv* format and must be in the same directory as the Problog KB. These files can be loaded into the Problog KB using the library **db** and the predicate **csv_load(+Filename, +Predicatename)**. Each time a file is uploaded into the Problog KB, a new fact with a new predicate is created. Therefore, the number of files loaded creates the exact same number of predicates.

Consider an example of a *csv* file named **test.csv**. This file contains data about a person name and its respective gender. Assuming that the **test.csv** file holds the following information:

```
"person","gender"
"John","male"
"Ruth","female"
```

The file may be loaded into the Problog KB as:

```
:- use_module(library(db)).
:- csv_load("test.csv", "person_gender").
```

After use the **`csv_load`** predicate to load the information of the **`test.csv`** file into Problog, two different facts were automatically created with the predicate *person_gender*. The resulting structure look like the following:

```
person_gender("John", "male").
person_gender("Ruth", "female").
```

The next chapter is devoted to introduce the basic logic underlying the *ProceOmics* program and the Problog KB development. It is divided into the two following sections:

1. *ProceOmics* Program.
2. Problog KB.

The first section starts by describing the *ProceOmics* program structure. It is followed by a briefly description of the program menus that performs the processing approaches and a detailed explanation of each probability inference mechanism developed and their purpose. The second section describes the reasoning under the Problog KB and its contents. It is also dedicated to provide the crucial background on the facts that compose rules that are further queried.

4. Development

4.1 *ProceOmics* Program

The *ProceOmics* program consists of a python module that uses other sub modules for data processing and Problog oriented-formatting. It is user-friendly and based on a menu system. It was built with python 3.5 under IDE Spyder.

As previously mentioned, the main goal of the program consists in performing a concise processing and formatting procedures to cancer genomic datasets downloaded from the databases, mentioned in chapter 3, section 3.1, to a probabilistic logic programming syntax. The final results are stored in *csv* files that follow a specific structure so they can be loaded to the Problog KB as facts.

The user must enter options and file names as input. In order to facilitate its usage, all the required python scripts are contained within a specific directory called *DataProcessProgram*. When the program processes an input file and writes a new *csv* file that contains the processed results for the first time, it automatically creates a new directory to store the output file, which is named *OutputFiles*. All the remaining new generated *csv* files will be stored in that same directory. Each of the output files the program creates are associated to a different action. The association between the action and the respective output file is represented at Table 9 present in appendix II.

The program flows through interactions between the computer and the user. Each selected option is interconnected to another menu until the user reaches a desired one and performs an action.

Along with the menu where the processing actions are available, the program contains two other menus. One of them allows the user to load the files to be processed into the specific directories within the program directory and the other menu allows the user to create auxiliary files.

After all the files are loaded and the auxiliary files created and stored, the program is ready to perform the processing approaches at full potential. Although it is possible to use the program without applying these last two mentioned menus (**Load Input Files** menu and **Get Auxiliary Files** menu), some of the processing approaches available may be incomplete.

In order to process correctly the input data, it is required that the structure of the genomic data follows the structure as specified in Tables 3 to Tables 8, even if it refers to different types of cancer.

4.1.1 Program Structure

The general structure of the program is represented at Figure 2.

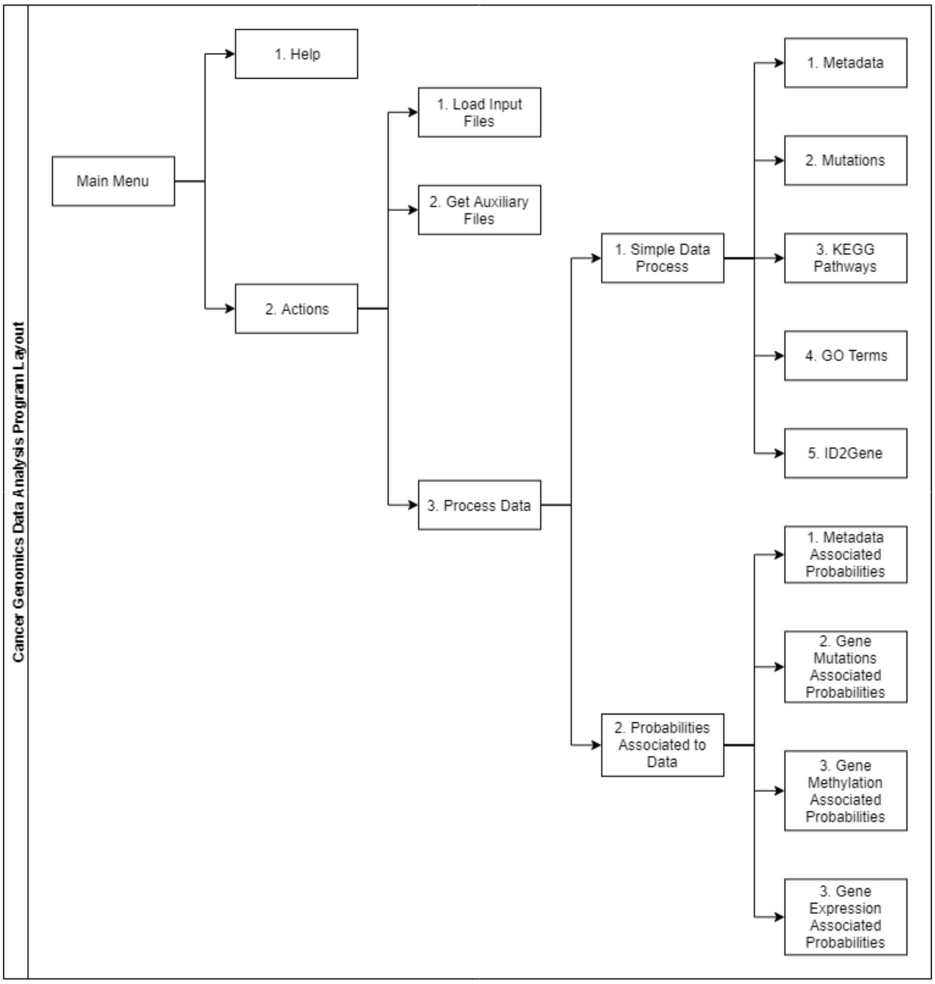

Cancer Genomics Data Analysis Program Layout
Main Menu
1. Help
2. Actions
1. Load Input Files
2. Get Auxiliary Files
3. Process Data
1. Simple Data Process
2. Probabilities Associated to Data
1. Metadata
2. Mutations
3. KEGG Pathways
4. GO Terms
5. ID2Gene
1. Metadata Associated Probabilities
2. Gene Mutations Associated Probabilities
3. Gene Methylation Associated Probabilities
3. Gene Expression Associated Probabilities

Figure 2 - *ProceOmics* structure layout.

The program structure may be compared to a tree data structure where the nodes are the available menus and the edges are the possible links between menus. Traceback between linked menus is always a possible option.

The represented leaves represent the core operations of the program. Each single one contains different options to choose, with available distinct actions. Each action consists in different processing approaches that are applied to the various cancer genomic datasets. Some of the approaches are simple and generally applied to most of the datasets while some others are more complex and specific to some datasets.

Although Figure 2 does not represent leaves for the **Load Input Files** and **Get Auxiliary Files** menus, they also contain different options for the user to choose. However, since none of those menus are strictly directed to data processing, their leaves were omitted in the program structure layout due to aesthetic reasons.

4.1.2 Data Processing Menus

Simple Data Process Menu

While in the **Simple Data Process** menu, the user has five available options to choose and each one of them containing different actions. These actions enable the user to apply several processing approaches on the intended data.

All processing approaches are applied to the metadata input files in **Metadata** menu and to the mutations input file in **Mutations** and **ID2Gene** menus.

Along with the feature selection procedure, all the input files are also subjected to very simple processing steps, such as non-existing value (NA) transformation\deletion, ambiguous character replacement, truncate IDs values, removal of duplicated values, conversion of non-numeric values to lower case, rename columns, among others.

Probabilities Associated to Data Menu

In the **Probabilities Associated to Data** menu, the user has four available options to choose and again each one of them contains different actions. Similar to the **Simple Data Process** menu, these actions enable the user to apply several processing steps. However, this menu has a particular goal. These actions are used to perform specific probability inference techniques.

Therefore, these actions generic workflow comprehend two sequential steps:

1. Data processing.
2. Probability inference.

The first step uses similar feature selection and data processing approaches to the ones applied on the **Simple Data Process** menu.

The second step applies specific calculations to infer probabilities. These calculations are differentially performed according to the type of data upon which they are applied. Their mechanisms are explained in the next section.

All of the processing steps as well the probabilities inference techniques applied in both menus available actions are discriminated in Table 11 and described in Table 12. Both tables are present at appendix III.

4.1.3 Probability Inference Techniques

Metadata Probabilities Inference

To perform this probability inference technique, it is required that the user give as input the metadata dataset (1) or metadata dataset (2) and the *pt_cs.csv* file that matches each patient identifier to its respective cancer subtype. This last file is generated by the first available action of the **Get Auxiliary Files** menu.

After all the inputs are correctly introduced, the program first performs feature selection and processing approaches on the original metadata dataset. Then all the patients' IDs and respective metadata feature of interest are mapped with the respective cancer subtype data, which is in the *pt_cs.csv* file.

Further details on fitting statistical distributions to each feature are described below.

Age Probabilities Inference

Originally, the metadata dataset (1), described in the previous chapter, contains a column where the patients' respective age values are stored. Attaching the patients' respective cancer subtype to the mentioned values provides a subsequent union between age and cancer subtype data. However, there is always a probability associated to the incidence of particular cancer subtype at a certain age. Therefore, the distribution of age by cancer subtype needs to be modelled in order to infer its associated probability.

The program computes the mean and the standard deviation of all age values by cancer subtype to further apply a Gaussian distribution, which is used to fit the probabilities of age according to each cancer subtype. The Gaussian distribution is mathematically performed through the Gaussian formula (3) and the resulting probabilities are stored. The (3) formula estimates the probability of an age value *t* to occur for a cancer subtype *i*. This formula is applied to all age values of the metadata dataset within all cancer subtypes.

$$f(t,i) = \frac{1}{\sigma_i\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}\frac{(t-\mu_i)^2}{\sigma_i{}^2}} \tag{3}$$

Other Metadata Probabilities Inference

Besides the age attribute, the probabilities were inferred from other features of interest from the metadata datasets (1) and (2) including gender, race, country, Lauren classification, ethnicity and cancer stage features.

Similarly as was done for the age attribute, the program binds the patient identifiers and the metadata feature to the respective cancer subtype. Then, it computes all the occurrences of each unique metadata value within a specific cancer subtype. This is later divided by the total frequency of the respective metadata value. This approach is given by formula (4) which estimates the probability of a specific metadata value *i* be associated to a cancer subtype *j*.

$$metadata_{(i)}\_cs_{(j)}\_prob = \frac{n\_samples_{(i,j)}}{n\_samples_{(i)}} \tag{4}$$

$n\_samples_{(i,j)}$ – frequency of a metadata value *i* within a cancer subtype *j*
$n\_samples_{(i)}$ – total frequency of a metadata value *i*

As an example, the program may apply formula (4) to gender feature in order to compute the probability of the two unique gender values – male and female – be associated to each cancer subtype. Therefore, given the four different subtypes of stomach cancer and since each one of them occurs in the two distinct genders, this formula outputs eight probabilities that corresponds to the association of each unique gender value to each unique cancer subtypes.

Mutations Probabilities Inference

The program contains two distinct actions to infer the probabilities on the mutations data.

In the **Mutated Gene + Cancer Subtype + Probability** action, the program allows the user to acquire a *csv* file that maps the probability of the mutated genes be present in a specific cancer subtype. This relation provides evidences to find the most common mutated genes in the various cancer subtypes.

In order to perform this probability inference approach, the program computes all the occurrence of each mutated gene in the different cancer subtypes. All the occurrence values are later divided by the occurrence number of the respective cancer subtype.

This probability inference approach is performed by the formula (5) which estimates the probability of a mutated gene *i* be present in a cancer subtype *k*. The resulting values of the applied formula are stored in a new column.

$$mutgene_{(i)}\_cs_{(k)}\_prob = \frac{n\_samples_{(i,k)}}{n\_samples_{(k)}} \tag{5}$$

$n\_samples_{(i,k)}$ - number of samples in a cancer subtype *k* with a mutated gene *i*
$n\_samples_{(k)}$ - number of samples in a cancer subtype *k*

In the **Mutated Gene + Variant Classification + Probability** action, the user has the possibility to obtain a *csv* file that has the probabilities of the mutated genes have a defined impact on the DNA. This type of information is valuable once it yields information about the most common somatic variants in specific mutated genes.

The approach applied in this action is achieved dividing the number of occurrences of a variant classification in a specific gene by the total number of that same variant classification in the whole dataset.

This calculation is represented and performed by formula (6) which estimate the probability of a mutated gene *i* have a variant classification *k*.

$$mutgene_{(i)}\_vc_{(k)}\_prob = \frac{n\_samples_{(i,k)}}{n\_samples_{(k)}} \tag{6}$$

$n\_samples_{(i,k)}$ - number of samples with a variant classification *k* in a mutated gene *i*
$n\_samples_{(k)}$ - number of samples with a variant classification *k*

Gene Expression Probabilities Inference

Similarly to previous probability inference techniques, the gene expression probability inference requires the join with the cancer subtype data. Therefore, when the user selects to perform this action, the program creates a *csv* file that stores information about the expression profiles of certain genes within a specific cancer subtype.

The expression profile is a scaled term for a certain range of FPKM values. This gene expression profile scale is represented at Table 10 in appendix II.

The gene expression probability inference is performed dividing the number of cases where a gene has an expression profile within a cancer subtype by the total case number of that same gene within that same cancer subtype. The probability inference technique is represented by formula (7) which estimates the probability of a gene *i* within a cancer subtype *j* have an expression profile *k*.

$$gene_{(i)}\_cs_{(j)}\_expprof_{(k)}\_prob = \frac{n\_cases_{(i,j,k)}}{n\_cases_{(i,j)}} \tag{7}$$

$n\_cases_{(i,j,k)}$ - number of cases in a gene *i* within a cancer subtype *j* with an expression profile *k*
$n\_cases_{(i,j)}$ - number of cases in a gene *i* within a cancer subtype *j*

In this action the program requires as input the directory path where all the methylation files are stored and the pt_cs.csv file. The program associates the probability of a gene be methylated in a specific cancer subtype. This data relation offers the user a better understanding of which genes have higher, or lower, methylation probabilities into the different cancer subtypes.

The methylation probabilities correspond to a direct mapping of the methylation status of the gene, i.e. its Beta-value. Therefore, in this case, the inference does not require the use of a function to derive the probabilities. Instead, the program reads the methylation text files contained in the *Methylation_Tables* directory and, one by one, perform several processing steps and add the final result to a dataframe. After all methylation files are processed, the relevant information from all the files is concatenated into just one dataframe structure.

The methylation status of each gene can be captured by one or more microarray probes. When more than one probe is available for a given gene, the mean *Beta value* is used to represent the probability of that gene being methylated.

4.2 Problog KB

A Problog KB was developed and oriented to use genomic data from stomach cancer. Currently, there is an absence of IDEs oriented to Problog. The knowledge bases may be developed under Prolog IDE, source code editors or simple text editors, i.e. notepad. Therefore, the Problog KB was built under notepad. We choose this simple text editor since it is already integrated into the operating system used and there is no need to install any new software.

The engine used under the Problog KB development was ProbLog2. This is a second generation engine that reasons with the Problog language. The main reason why we selected this engine is due to the possibility to learn the parameters of the Problog program from partial interpretations and support intensional probabilistic facts with a flexible probability.

The developed Problog KB was developed considering a specific organization and divided into four components, which are represented at Figure 3.

Problog KB General Structure	
	(A) Import Data Load data form csv files as background knowledge
	(B) Create New Facts Based on information present at literature
	(C) Develop Rules Rules codifying data relations and intereactions between facts
	(D) Query the Problog KB

Figure 3 – General structure of Problog KB.

Import Data

After the original genomic data is processed by *ProceOmics*, the resulting data, stored in *csv* files, are loaded into the Problog KB as facts. As previously mentioned, the *csv* files are uploaded with the ***csv_load*** predicate. Table 13 displays the Problog code lines that allow importing the *csv* files sequentially with an example of the respective predicate.

Table 13 – Code for loading stomach cancer genomic data into Problog KB.

Problog KB	Predicate Example	Fact ID
:- use_module(library(db)).	*db* library which allows to read *csv* files	-
:- csv_load('pt_age.csv',meta_age).	meta_age('TCGA-B7-5816',51).	A1
:- csv_load('pt_cs.csv',meta_cs).	meta_cs('TCGA-B7-5816',msi).	A2
:- csv_load('pt_gender.csv', meta_gender).	meta_gender('TCGA-B7-5816',female).	A3
:- csv_load('pt_race.csv',meta_race).	meta_race('TCGA-B7-5816',white).	A4
:- csv_load('pt_country.csv', meta_country).	meta_country('TCGA-B7-5816', russia).	A5
:- csv_load('pt_lc.csv',meta_lc).	meta_lc('TCGA-B7-5816',diffuse).	A6
:- csv_load('pt_stage.csv',meta_stage).	meta_stage('TCGA-B7-5816',stage_iib).	A7
:- csv_load('cs_age_prob.csv', cs_age).	cs_age(cin,51,0.0115).	A8
:- csv_load('cs_gender_prob.csv', cs_gender).	cs_gender(female, cin, 0.442).	A9
:- csv_load('cs_race_prob.csv', cs_race).	cs_race(asian, cin, 0.9.494).	A10
:- csv_load('cs_country_prob.csv', cs_country).	cs_country(canada, cin, 0.667).	A11
:- csv_load('cs_lc_prob.csv', cs_lc).	cs_lc(diffuse, cin, 0.261).	A12
:- csv_load('cs_stage_prob.csv', cs_stage).	cs_stage(stage_ia, cin, 0.5).	A13
:- csv_load('DR_mutgene_cs_prob.csv', mutgene_cs).	mutgene_cs('ABCA12',cin,0.108).	A14
:- csv_load('DR_GeneExpression_cs.csv', gene_cs_expprof).	gene_cs_expprof(ENSG0000000533 9,cin, high_exp, 0.563).	A15

Table 13 – Code for loading stomach cancer genomic data into Problog KB (continuation).

Problog KB	Predicate Example	Fact ID
:- csv_load('DR_metgene_cs.csv', metgene_cs).	metgene_cs('ABCA12',msi,0.67).	A16
:- csv_load('DR_mutgene_vc_prob.csv', mutgene_vc).	mutgene_vc('ABCA12',frame_shift_del, 0.045).	A17
:- csv_load('DR_pt_gene_mutclass.csv', pt_mutgene_mutclass).	pt_mutgene_mutclass('TCGA-B7-5816', 'RERE', missense_mutation).	A18
:- csv_load('kegg.csv', gene_keggpath).	gene_keggpath('RYR3','Apelin signaling pathway').	A19
:- csv_load('GOids.csv', gene_goterm).	gene_goterm('ENSG00000005339', 'GO:0000122').	A20
:- csv_load('DR_Id2Gene.csv',id2gene).	id2gene(ENSG00000142599,'RERE') .	A21

As a result of each performed file loading, new facts, with a new predicate, are created. Each new predicate contains the same number of arguments as the number of columns of the respective loaded file.

A total of 21 files were loaded into the Problog KB. This action created 21 non-probabilistic facts that codify different stomach cancer genomic data types and auxiliary information.

Facts A1 to A7 encode simple metadata information. All the resulting predicates contain two arguments. The first argument is the patient identifier and the second argument is the respective metadata information.

Facts A8 to A17 represent the genomic and clinical events with the associated probabilities. The resulting predicates arity differ. However, in all these predicates, the last argument is the associated probability inferred by *ProceOmics* program.

The last three facts (facts A19 to A21) contained auxiliary information to the genomic events. All three predicates contain two arguments.

Create New Facts

The non-probabilistic A8 to A17 facts, mentioned above, were used in order to construct 10 intensional probabilistic facts with a flexible probability. For each one of these 10 non-probabilistic facts, a respective intensional probabilistic fact was manually created, see facts B23 to B32 displayed at Table 14 below. This allows to instantiate the probability of B23 to B32 facts accordingly to the respective A8 to A17 facts last arguments.

As an example, consider the following intensional probabilistic fact **P::mutgene_vc_prob(GENE, VC) :- mutgene_vc(GENE, VC, 0.7).** For each ground instantiation *P* for which **mutgene_vc(GENE, VC, 0.7)** occurs, there is a corresponding probabilistic fact **0.7::mutgene_vc_prob(GENE, VC)**.

Additional information about the cancer genomic data that lacked in the *csv* files was also manually introduced in the Problog KB as facts. This information was retrieved from the TCGA study (Bass et al., 2014).

The new literature-based facts include the ground probabilities of the cancer subtypes occurrence.

Like all Problog programs, all types of facts created, probabilistic or not, were later used in the program rules and/or queries. Auxiliary ground and intensional probabilistic facts created are represented at Table 14, which contains their encoding and description.

Table 14 - Problog KB facts.

Problog KB facts	Description	Fact ID
0.088::cancer_subtype(ebv). 0.217::cancer_subtype(msi). 0.197::cancer_subtype(gs). 0.498::cancer_subtype(cin).	Ground probability of each cancer subtype (cs).	B22
P::cs_age_prob(CS,AGE) :- cs_age(CS,AGE,P).	Intensional probabilistic fact of each age value be associated to the cancer subtypes.	B23
P::cs_gender_prob(CS,GDR) :- cs_gender(GDR,CS,P).	Intensional probabilistic fact of each gender value be associated to the cancer subtypes.	B24
P::cs_country_prob(CS,COUNTRY) :- cs_country(COUNTRY,CS,P).	Intensional probabilistic fact of each country value be associated to the cancer subtypes.	B25
P::cs_lc_prob(CS,LC) :- cs_lc(LC,CS,P).	Intensional probabilistic fact of each lauren class value be associated to the cancer subtypes.	B26
P::cs_race_prob(CS,RACE) :- cs_race(RACE,CS,P).	Intensional probabilistic fact of each race value be associated to the cancer subtypes.	B27
P::cs_stage_prob(CS,STG) :- cs_stage(STG,CS,P).	Intensional probabilistic fact of each cancer stage value be associated to the cancer subtypes.	B28

Table 14 - Problog KB facts (continuation).

Problog KB facts	Description	Fact ID
P::mutgene_cs_prob(GENE,CS) :- mutgene_cs(GENE,CS,P).	Intensional probabilistic fact of each gene be mutated in all four cancer subtypes.	B29
P::mutgene_vc_prob(GENE,VC) :- mutgene_vc(GENE,VC,P).	Intensional probabilistic fact of each mutated gene have a certain variant classification.	B30
P::gene_cs_expprof_prob(GENE,CS,EXPPRO) :- id2gene(GID,GENE), gene_cs_expprof(GID,CS,EXPPRO,P).	Intensional probabilistic fact of expression profiles by cs for a gene.	B31
P::metgene_cs_prob(GENE,CS) :- metgene_cs(GENE,CS,P).	Intensional probabilistic fact of each gene be methylated in all four cancer subtypes.	B32
P::phi_val(G1,G2) :- phi(G1,G2,P).	Intensional probabilistic fact that associates the phi coefficient as a probability value.	B33

Development of Rules

In order to interrogate the Problog KB, some rules were developed. They encode possible relations and interactions between the stomach cancer genomic, clinical and auxiliary data that are represented by the previously described facts. Some of the created rules only encode simple data interactions while the others encode more complex interactions. All the developed rules are displayed and described at Table 15. Each rule is also associated with a unique identifier that starts with the letter 'R' followed by a particular number.

Table 15 - Problog KB rules.

Problog KB rules	Description	Rule ID
match_clinical(CS,AGE,GDR,COUNT,LC,RACE,STAGE) :- cancer_subtype(CS), cs_age_prob(CS,AGE), cs_gender_prob(CS,GDR), cs_country_prob(CS,COUNT), cs_lc_prob(CS,LC), cs_race_prob(CS,RACE), cs_stage_prob(CS,STAGE).	Match all metadata. This rule aims to use partial or complete patient information to infer the probability of having one of the four cancer subtypes.	R1

Table 15 - Problog KB rules (continuation).

Problog KB rules	Description	Rule ID
match_clinical_genomic(CS,AGE,GDR,COUNT,LC,RACE,STAGE, MUTGENES) :- match_clinical(CS,AGE,GDR,COUNT,LC,RACE,STAGE), mutgene_list_prob(MUTGENES).	Match the probability of all partial or complete metadata to a given list of certain mutated genes.	R2
mutgene_list_prob([]). mutgene_list_prob([H|T]) :- mutgene_cs_prob(H,_), mutgene_list_prob(T).	Create a list of genes to insert in the match_metadata predicate.	R3
mutgene_vc_cs_prob(GENE,VC,CS) :- mutgene_vc_prob(GENE,VC), mutgene_cs_prob(GENE,CS).	Associate genes to a variant classification mutation type in a certain cancer subtype.	R4
mutgene_kp_cs_prob(GENE,KP,CS) :- mutgene_cs_prob(GENE,CS), gene_keggpath(GENE,KP).	Mutated gene has a Kegg pathway in a cancer subtype.	R5
mutgene_goterm_cs_prob(GENE,GO,CS) :- id2gene(GID,GENE), mutgene_cs_prob(GENE,CS), gene_goterm(GID,GO).	Mutated gene has an associated GO term in a cancer subtype.	R6
cs_metgene_expprof_prob(CS,GENE,EXPPRO) :- id2gene(GID,GENE), metgene_cs_prob(GENE,CS), gene_cs_expprof_prob(GENE,CS,EXPPRO).	Associate the methylated genes to a genetic expression within all cancer subtypes.	R7
cs_mutgene_metgene_expprof_prob(CS,GENE,EXP) :- mutgene_cs_prob(GENE,CS), metgene_cs_prob(GENE,CS), cs_mutgene_expprof(CS,GENE,EXP).	Associate the methylated and mutated genes to a genetic expression within all cancer subtypes.	R8
strangelen([],0). strangelen([H|T], Len) :- (H = [_|_], strangelen(H, LenH); LenH = 1),strangelen(T,LenT), Len is LenH + LenT.	Giving a certain list, get the respective list length.	R9
isNonElement(_,[]). isNonElement(X, [Y|Z]) :- X \= Y, isNonElement(X,Z).	Check if an element is not in a list.	R10
delete(_, [], []). delete(Y, [X|W], Z) :- member(X,Y), delete(Y,W,Z). delete(Y, [X|W], [X|Z]) :- isNonElement(X, Y), delete(Y, W, Z).	Delete an element from a list.	R11

Table 15 - Problog KB rules (continuation).

Problog KB rules	Description	Rule ID
all_samp(L) :- findall(PT, pt_mutgene_mutclass(PT,_,_), X), sort(X,L).	Get a list of all samples in the KB that has mutated genes (without duplicates).	R12
n_all_samp(N) :- all_samp(L), strangelen(L,N).	Count the number of total samples that has gene mutated genes without duplicates in the KB.	R13
same_samp(G1,G2,PT) :- pt_mutgene_mutclass(PT,G1,_), pt_mutgene_mutclass(PT,G2,_).	Get different genes that are mutated in the same sample.	R14
oc1_oc2(G1,G2,PT) :- findall(S, same_samp(G1,G2,S), PT).	Get a list of samples that have mutations in two different genes.	R15
oc1_noc2(G1,G2,PT) :- findall(PT1, pt_mutgene_mutclass(PT1,G1,_), L1), findall(PT2, pt_mutgene_mutclass(PT2,G2,_), L2), delete(L2,L1,PT).	Get a list of samples that just have mutations in the first of two given gene.	R16
noc1_oc2(G1,G2,PT) :- findall(PT1, pt_mutgene_mutclass(PT1,G1,_), L1), findall(PT2, pt_mutgene_mutclass(PT2,G2,_), L2), delete(L1,L2,PT).	Get a list of samples that just have mutations in the second given gene.	R17
noc1_noc2(G1,G2,PT) :- findall(PT3, same_samp(G1,G2,PT3), L), all_samp(AS), delete(L,AS,PT).	Get a list of samples that does not have mutations in any of the two given genes.	R18
freq_oc1_oc2(G1,G2,N) :- oc1_oc2(G1,G2,PT), strangelen(PT, N).	Get the number of samples that have mutations in both given genes.	R19
freq_oc1_noc2(G1,G2,N) :- oc1_noc2(G1,G2,PT), strangelen(PT,N).	Get the number of samples that just have mutations in the first of the two given genes.	R20

Table 15 - Problog KB rules (continuation).

Problog KB rules	Description	Rule ID
freq_noc1_oc2(G1,G2,N) :- noc1_oc2(G1,G2,PT), strangelen(PT,N).	Get the number of samples that just have mutations in the second given gene.	R21
freq_noc1_noc2(G1,G2,N) :- noc1_noc2(G1,G2,PT), strangelen(PT,N).	Get the number of samples that does not have mutations in any of the two given genes.	R22
phi(G1,G2,PHI) :- freq_oc1_oc2(G1,G2,N11), freq_oc1_noc2(G1,G2,N10), freq_noc1_oc2(G1,G2,N01), freq_noc1_noc2(G1,G2,N00), N11 > 0, N10 > 0, N01 > 0, N00 > 0, PHI is ((N11*N00)-(N10*N01))/sqrt((N11+N10)*(N01+N00)*(N11+N01)*(N10+N00)).	Calculate the phi coefficient for two given genes.	R23
P::phi_val(G1,G2) :- phi(G1,G2,P).	Convert the phi predicate result to a probability.	R24

With the goal of performing queries that satisfy the objectives mentioned in the beginning of this section, four different case studies were performed. They are described in the next chapter. With the first case study we intended to infer the probability of a patient having a certain cancer subtype using genomic information. The second case study is devoted to explore the possible relation between different types of genomic data, in particular methylation and gene expression. The third case study is dedicated to implement a measure of association between mutation data. With the last case study, we sought to fit a patient within a certain cancer subtype given its partial clinical information.

5. Results and Discussion

In this study we have as main goal to develop a set of queries that allow inferring the probability of patient having a certain cancer subtype. For that we use the available genomic and clinical information on the patient and with the application of the appropriate query we derive this probability. Note, that the extent of genomic data available for the query patient is variable. Therefore, the rules and queries reflect the extent of the available data. Other queries were developed to demonstrate the possibilities of probabilistic logic programming on combining different types of data and the application of measures of association between genomic data.

All the results obtained from the four case studies are presented and discussed in the next sections. Appendix IV contains the queries outputs when they are very extensive.

5.1.1 Case study I: Distribution of patients' somatic mutations by genes across the different stomach cancer subtypes

Given information on somatic variants and/or mutated genes we used the Problog KB rule R4 to compute the probabilities of a patient having a certain stomach cancer subtype. This rule is composed by two distinct intensional probabilistic facts, which predicates are: *mutgene_cs_prob* and *mutgene_vc_prob*. The former stores a flexible probability, which provides the likelihood that each gene has of being mutated under the different cancer subtypes. The later performs a similar task but, instead of associating flexible probabilities to mutated genes, it attaches those probabilities to genetic variant classifications and their respective cancer subtype. In order to explore all the possible interactions between the data, we developed three different approaches of the same query.

The first approach allowed to perceive which of the variant type of six known and randomly chosen patient mutated genes may occur more or less frequently in each stomach cancer subtype, respectively. A second approach sought to supply a more comprehensive survey on which mutated genes fitted in the distinct cancer subtypes, given a specific somatic variant. In the last approach, the connection between the mutated genes and their respective variant classes was analysed in more detail.

The queries and respective results are shown in appendix IV and a descriptive distribution of all the three query versions results are represented in Figure 4, Figure 5 and Figure 6 below.

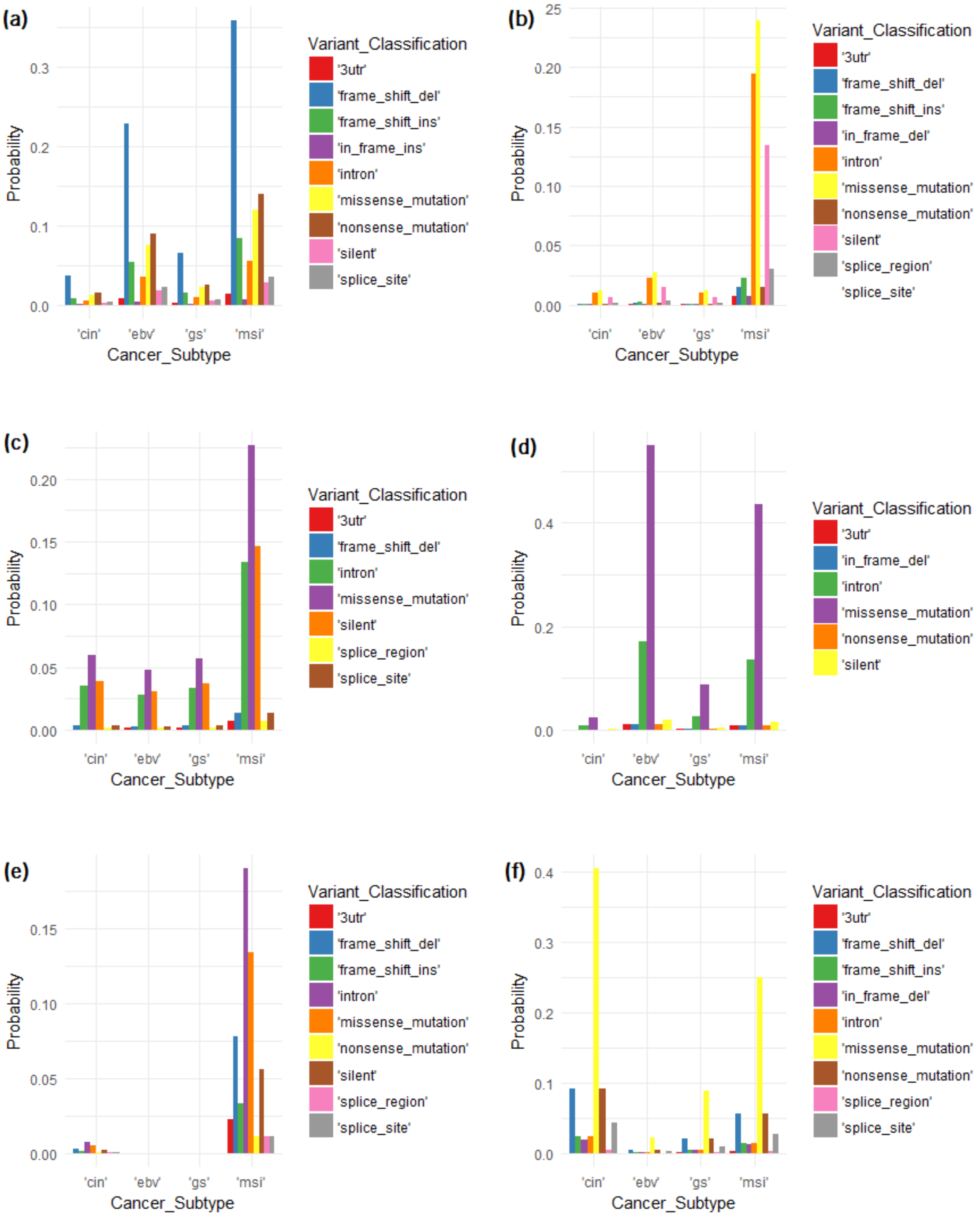


Figure 4 – First approach of case study I.

Distribution of variant classification occurrences within the stomach cancer subtypes for specific given mutated genes. (a) ARID1A mutated gene; (b) HERC2 mutated gene; (c) LAMA1 mutated gene; (d) PIK3CA mutated gene; (e) RERE mutated gene; (f) TP53 mutated gene.

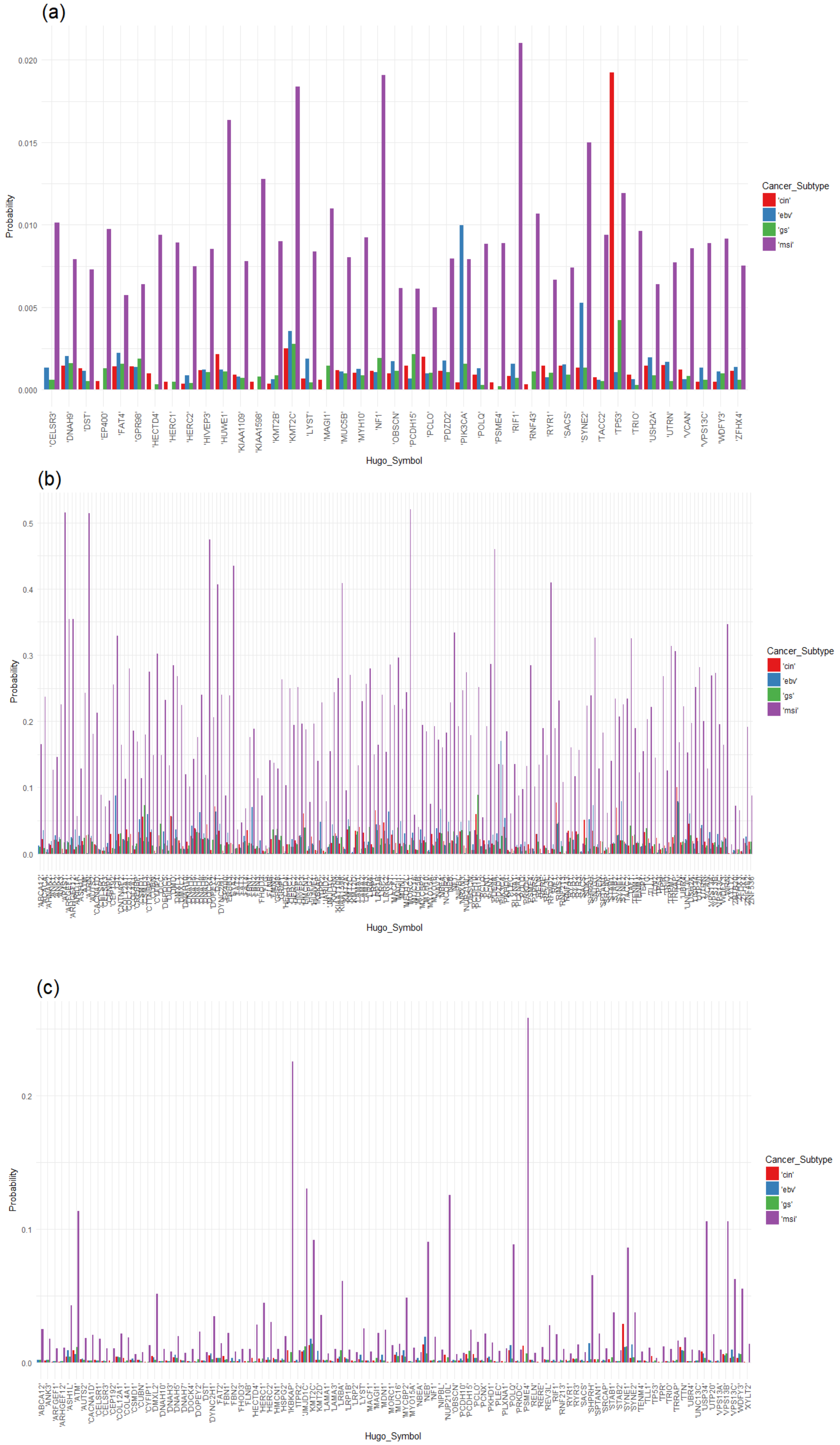


Figure 5 – Second approach of case study I.

Distribution of mutated genes within the cancer subtypes for a given variant classification. (a) in_frame_del variant; (b) intron variant; (c) splice region variant.

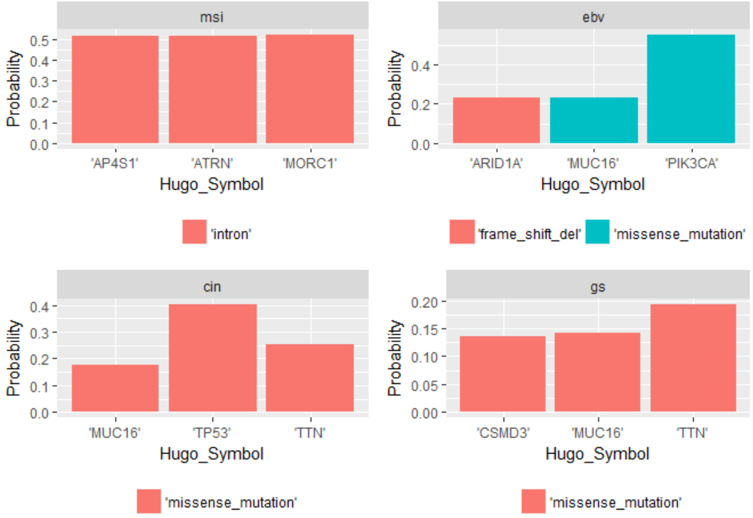


Figure 6 – Third Approach of Case Study III.
Three most mutated genes and respective somatic variant within cancer subtype.

A general overview of Figure 4 quickly suggests that, for each of the six mutated genes always exists a specific variant classification that has the highest probability of occurrence, regardless of the cancer subtype. This specific variant classification differs between the six mutated genes.

From Figure 5, it is possible to observe an absolute prevalence of mutated genes within *msi* subtype, independently the three tested variants. However, in cases of *in frame deletion* variants, PIK3CA and TP53 are most probable to be associated with this somatic variant in *ebv* and *cin* subtypes, respectively.

Figure 6 resutls show consistency with the other studied approaches in this case study. From its analysis, it is possible to verify that the most mutated genes between the different cancer subtypes are in agreement with the remaining results.

Most of the results are in accordance with literature papers. As an example, Wu and colleagues (Wu et al., 2005) provides a figure that contains a summary of the somatic mutations according to molecular subtype where somatic variants are differentiated for a set of genes. Accordingly to their figure, TP53 does have the highest number of somatic mutations in *cin* (56,5%) and they are mainly missense mutations. ARID1A contains low number of mutation in *cin* subtype (26,1%) and mostly *frameshift* variants at *msi* (73,7%). PIK3CA holds somatic mutations principally in *ebv* (28,6%) and *msi* (36,8%) subtypes and most of them are missense mutations. Other studies also support our results for TP53, ARID1A and PIK3CA somatic mutations (Chia & Tan, 2016), (Pan, Ji, Zhang, Zhou, & Zhong, 2018) and (Bass et al., 2014). The association of TP53 with in frame deletion variant is also supported by Bass and colleagues (Bass et al., 2014). Few information on the role of HERC2, LAMA1 and RERE in stomach cancer was found on literature.

5.1.2 Case Study II: Influence of methylation on gene expression

In this case study, we analysed the relation between two types genomic data: gene expression and methylation for the different cancer subtypes.

The Problog KB R7 was used to compute the probabilities of this relation. This rule contains two intensional probabilistic facts and one non-probabilistic fact within its body. The non-probabilistic fact, *id2gene*, maps each gene ensembl ID from the gene expression data with the respective common gene name from the methylation data. The intensional probabilistic fact *metgene_cs_prob* attribute a probability of a certain gene to be methylated in the given subtype while the other intensional probabilistic fact, *gene_cs_expprof_prob*, assign probabilities to each occurred gene expression profile of a certain gene, within the cancer subtypes.

A first approach explored the probability distribution of methylation events for a set of genes within the four stomach cancer subtypes. A second approach was intended to examine the impact of these methylation events in the gene expression profiles of those previously selected set of genes. The set of genes was restricted to the same six genes that were previously used in Case Study I.

The performed queries and results are shown in appendix IV. Figure 7 and Figure 8 represent a descriptive interpretation from both approaches.

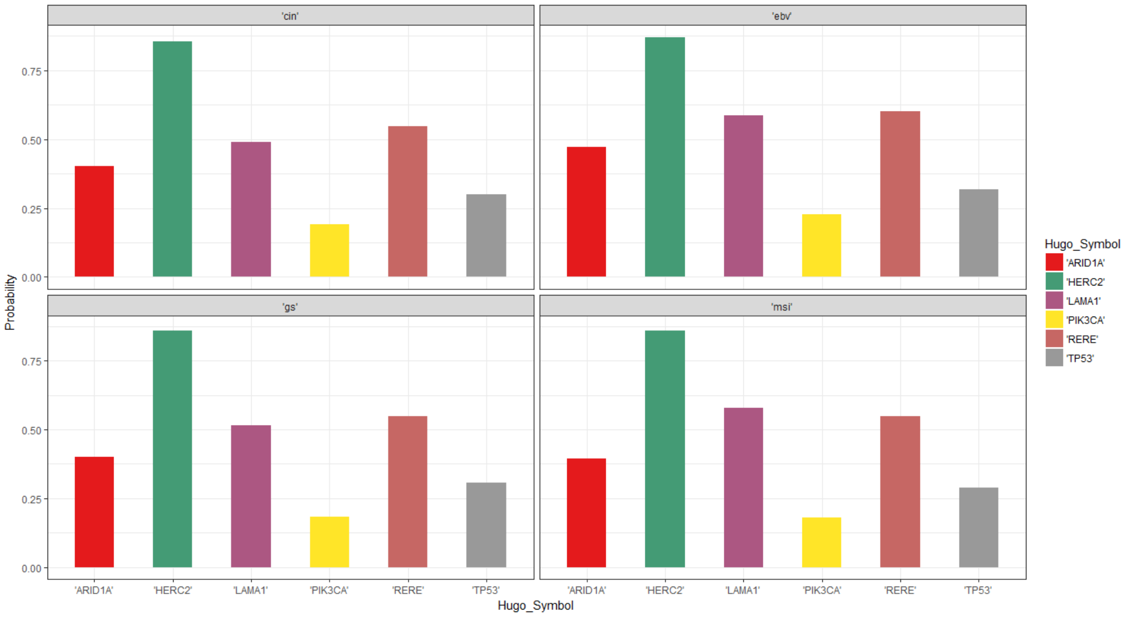


Figure 7 - Methylation probabilities for a set of genes within the four stomach cancer subtypes.

The results from Figure 7 demonstrate that, all genes show similar methylation probabilities, regardless the cancer subtype. Therefore, the probability methylation of a gene has a low relationship to the respective cancer subtype in which the event occurs.

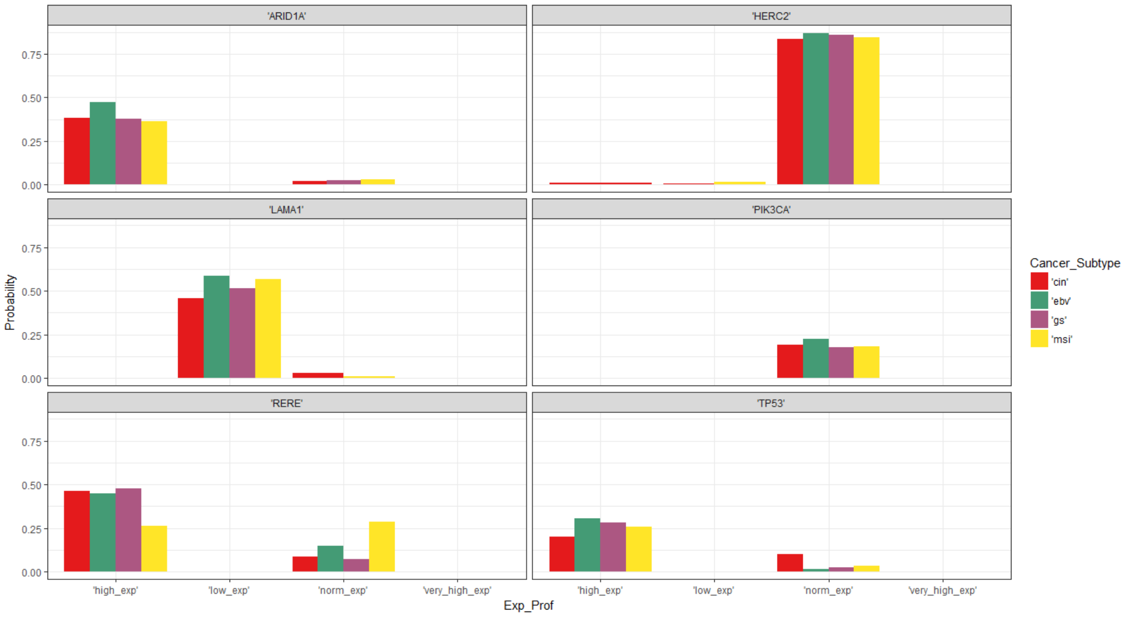


Figure 8 - Methylation events joined with gene expression profiles within each cancer subtype, for the selected set of genes.

Figure 8 shows a higher propensity of methylated genes to have an abnormal genetic expression – no normally expressed profiles – with subtype specific probabilities.

ARID1A, RERE, TP53 and LAMA1 genes are associated to higher probabilities of having abnormal genetic expression profiles when methylated while HERC2 and PIK3CA genes are more likely to have normal expression profiles. Most of these abnormalities usually are largely associated with *ebv* subtypes. Interestingly, this is also the most probable subtype in normally expressed genes.

Public reports have shown agreement with our results. Aso and colleagues (Aso, Uozaki, Morita, Kumagai, & Watanabe, 2015), stated that *ebv* status shown relation to ARID1A abnormality. It is also known that TP53 is often upregulated in stomach cancer cases (Wang, Stemmermann, & Noffsinger, 2003).

However, Riquelme *et al.* (Riquelme, Tapia, Espinoza, & Leal, 2016) describes that PIK3CA is pratically always found in overexpression scenarios in gastric cancer. Figure 8 results sugests that PIK3CA solely has normal expression profiles. Thus, this divergent information may suggest that methylation of PIK3CA, regardless the cancer subtype in which it occurs, regulates the overexpression of this gene in stomach cancer cases.

5.1.3 Case Study III: Co-occurrence mutations between genes

In this case study we measured the co-occurrence of mutations for each pair of genes. The phi coefficient, described by Om (Om, 2011), was used to measure the strength of co-occurrence. This query was developed in order to obtain the phi coefficient for the number of mutation occurrences (binary variable automatically calculated by the Problog KB rules R26 to R29) between two mutated genes. It determines if the mutated genes contain co-occurring or exclusively mutations. Phi coefficient values closer to zero indicate exclusivity of the occurrences. Phi values higher than zero indicate that two genes have mutations co-occurring, i.e. occurring in the same samples.

The rule that computed the *phi* coefficient is R31. However, in order to perform its calculations, this rule requires other auxiliary rules (R16 to R30). A total of 15 queries were performed. All of them used the same rule – phi_val. Table 16 displays each performed query and the respective output.

Table 16 – Individual query results from case study III.

Query	Output
phi_val('ARID1A','HERC2').	0.639
phi_val('ARID1A','LAMA1').	0.451
phi_val('ARID1A','PIK3CA').	0.569
phi_val('ARID1A','RERE').	0.531
phi_val('ARID1A','TP53').	0.134
phi_val('HERC2','LAMA1').	0.518
phi_val('HERC2','PIK3CA').	0.536
phi_val('HERC2','RERE').	0.625
phi_val('HERC2','TP53').	0.158
phi_val('LAMA1','PIK3CA').	0.349
phi_val('LAMA1','RERE').	0.517
phi_val('LAMA1','TP53').	0.176
phi_val('PIK3CA','RERE').	0.458
phi_val('PIK3CA','TP53').	0.072
phi_val('RERE','TP53').	0.152

In order to have a clear understanding of the results, all the queries outputs were converted into a co-occurrence table, which is represented by Table 17.

Table 17 – Co-occurrence table of case study III results.

	ARID1A	HERC2	LAMA1	PIK3CA	RERE	TP53
ARID1A	0					
HERC2	0.639	0				
LAMA1	0.451	0.518	0			
PIK3CA	0.569	0.536	0.349	0		
RERE	0.531	0.625	0.517	0.458	0	
TP53	0.134	0.158	0.176	0.072	0.152	0

The results show that ARID1A-HERC2 and ARID1A-PIK3CA have a high probability of co-occurring mutations. On the other hand, TP53-PIK3CA reveal a very low probability of mutation co-occurrence. All the remaining genes when paired with TP53 exhibit low co-occurrence mutation probability. These results are in agreement with the mutation co-occurrence events reported in several studies (Liang et al., 2012; Liu, Hu, Zhang, Hu, & Ye, 2018).

5.1.4 Case Study IV: Patient allocation in different cancer subtypes based on its incomplete clinical characteristics

For this case study we used Problog KB R1 to compute the probability of a patient having a certain cancer subtype given partial clinical information. This rule resorts to one probabilistic fact and six intensional probabilistic facts in order to achieve its goals. These utilized facts are B22 to B28. With this goal in mind, we defined a set of four constant clinical characteristics – country, lauren classification, race and cancer stage – which values were randomly chosen. Three clinical characteristics – cancer subtype, age and gender – remained as variables. However, in order to limit the proof search performed by the Problog engine, we limited the age values between 60 and 70 years old.

The performed query and results are shown in appendix IV. Figure 9 represent a descriptive interpretation of the results.

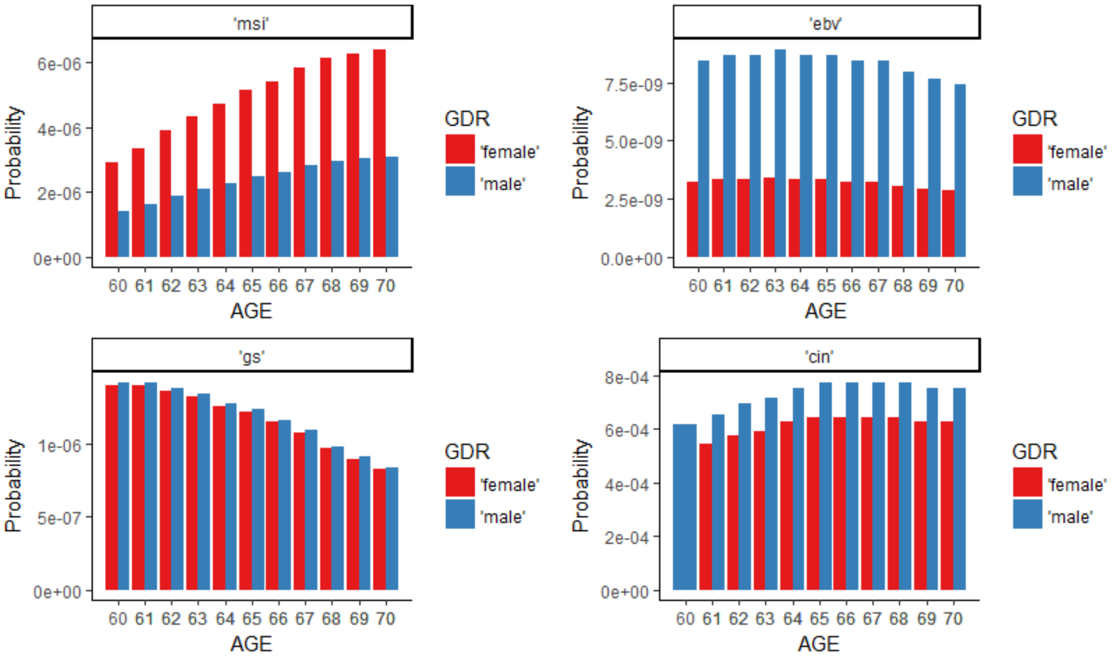


Figure 9 - Patient allocation in different cancer subtypes with variable age and gender attributes.

Assuming a possible white German individual with a mixed lauren classification and a stomach cancer stage IIa, Figure 10 shows the probabilities distribution of that patient be fitted in each cancer subtype and its possible age and gender instances. Individuals that follow those pre-specified clinical characteristics have higher probability of be fitted in *msi* subtypes at older ages and female cases. In case of *ebv* association, it is most probable that the patient tends to be male regardless its age. The patient has higher probabilities for earlier age cases when classified as a *gs* subtype. The male gender tendentially has higher probabilities than the female gender whenever a patient is fitted in one of the subtypes.

Our results reveal high conformity with Bass and colleagues (Bass et al., 2014). They published that *ebv* patients tended to be male, *msi* cases are mainly females and, in addition, that *gs* subtypes are diagnosed at earlier ages while *msi* subtypes are diagnosed at an older age.

6. Conclusions and Further Work

The application of probabilistic logic programming to the study of cancer genomic data was accomplished through the development of a program to process data and the development of queries that interrogate the respective data. *ProceOmics* program was developed in order to process and format stomach cancer genomic data. All the program operations produced well-strucutured and consistent *csv* files that stored the processed data. Although the program tries to either deal with minor or large datasets, by performing a data dimensionality reduction, often even the processed output file may still contain vast amounts of data. In those cases, the program may take considerable running time to finish, in particular in the processing of genetic expression and mutation datasets.

In order to test its utility when applied to genomic datasets of different types of cancer, genomic datasets on brain cancer were also processed. Although brain cancer datasets do not contain the exact same information as the stomach cancer, all the datasets that follow a structure similar to the stomach datasets was successfully processed and formatted. Therefore, we hope that the developed program can be applied to distinct types of cancer data, which was one of the original goals of this work.

The created Problog KB codes for queries on simple and complex relations between the different stomach cancer genomic data. However, this framework was not tested to genomic data related to other types of cancer.

It was possible to infer different conclusions based on the distribuiton of patient gene mutation data within the four different stomach cancer subtypes. The results of the first case study show higher probability of a patient to be classified in the *msi* subtype regardless of its somatic variants. Our results also revealed that *ebv* is the only cancer subtype that does have more than one different somatic variant in the three most frequently mutated genes.

The study case II demonstrates that when a gene is methylated, it has an unbalanced distribution over certain expression profiles. From the studied genes, with exception of RERE gene, the *ebv* subtype appears to be the most probable cancer subtype in a scenario of high expression. It is also notable that *cin* is the only cancer subtype that always has probabilty of expression in normal profiles. However, a sample of six genes is not significantly representative to infer a global overview on the relation between methylation and gene expression in stomach cancer.

Although the obtained results from the third case study reveals that TP53 has lower probability of co-ocurring mutations with the remaining analyzed genes, this case study should be performed with all Problog KB genes in order to confirm this statement.

Additionally to revealing consistent results with the literature, the case study IV explored the utiliy of Problog to deal with uncertainty.

Note that, to obtain a reasonable computational performance, this analysis was only performed on a reduced sub-set of data. Therefore, the explored relations in stomach cancer were not tested in whole panorama.

Even though the use of Problog in order to encode and manipulatie cancer genomic data has been so far unexplored, this thesis offers a seminal contribution on how genomic data can be modeled and further used under the probabilistic logic paradigm.

Although most of the proposed goals in this thesis were accomplished, there are still opportunities for further improvements. Furhter work may including new developments to the data processing program and the development of new queries for the Problog KB:

- Code optimization to increase the *ProceOmics* operation speed;
- Expand the *ProceOmics* processing approaches to other features;
- Expand the genomic information encoded in the Problog KB;
- Integrate graphical representation about features of interest within *ProceOmics*;
- Encode different cancer data as background knowledge on Problog KB;
- Create new rules that provides more accurate and precise queries;
- Generate new facts through machine learning algorithms based on association rules and decision trees.

Bibliography

Allred, D., Preist, C., Bennett, M., & Gupta, A. (1991). AGATHA: An Integrated Expert System to Test and Diagnose Complex PC Boards.

American, T., Dictionary, H., Language, E., Edition, T., & Merriam-webster, T. (2013). Programming Paradigms chapter, 1–8.

Aso, T., Uozaki, H., Morita, S., Kumagai, A., & Watanabe, M. (2015). Loss of ARID1A , ARID1B , and ARID2 Expression During Progression of Gastric Cancer, *6828*, 6819–6827.

Bain, M. (1994). Learning logical exceptions in chess.

Bani-hani, K. E., Almasri, N. M., & Khader, Y. S. (2005). Combined Evaluation of Expressions of Cyclin E and p53 Proteins as Prognostic Factors for Patients with Gastric Cancer Combined Evaluation of Expressions of Cyclin E and p53 Proteins as Prognostic Factors for Patients with Gastric Cancer. *Clinical Cancer Research*, *11*, 1447–1453.

Bass, A. J., Thorsson, V., Shmulevich, I., Reynolds, S. M., Miller, M., Bernard, B., ... Liu, J. (2014). Comprehensive molecular characterization of gastric adenocarcinoma. *Nature*, *513*(7517), 202–209.

Bratko, I. (1987). PROLOG Programming for Artificial Intelligence. *Information and Software Technology*, *29*(6), 339–340.

Cerami, E., Gao, J., Dogrusoz, U., Gross, B. E., Onur, S., Larsson, E., ... Sander, C. (2014). NIH Public Access, *2*(5), 401–404.

Chia, N., & Tan, P. (2016). reviews Molecular classi fi cation of gastric cancer, 763–769.

Consortium, I. C. G. (2017). International Cancer Genome Consortium. Retrieved from https://icgc.org/

De Raedt, L., & Kimmig, A. (2015). Probabilistic (logic) programming concepts. *Machine Learning*, *100*(1), 5–47.

De Raedt, L., Kimmig, A., & Toivonen, H. (2007). ProbLog : A Probabilistic Prolog and Its Applications to Link. *Ijcai*, 2468–2473.

Dickson. (2003). Panel report: the potential of geons for generic 3-D object recognition.

Dries, A., Kimmig, A., Meert, W., Renkens, J., Van Den Broeck, G., Vlasselaer, J., & de Raedt, L. (2015). ProbLog2 : Probabilistic logic programming. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 312–315.

DTAI, R. G. (2015). Problog Tutorial. Retrieved from https://dtai.cs.kuleuven.be/problog/tutorial/basic/01_coins.html

Du, P., Zhang, X., Huang, C. C., Jafari, N., Kibbe, W. A., Hou, L., & Lin, S. M. (2010). Comparison of Beta-value and M-value methods for quantifying methylation levels by microarray analysis. *BMC Bioinformatics*, *11*(1), 587.

Endriss, U. (2016). Lecture Notes An Introduction to Prolog Programming Ulle Endriss.

Fakoor, R., Ladhak, F., Nazi, A., & Huber, M. (2013). Using deep learning to enhance cancer diagnosis and classification. *Proceeding of the 30th International Conference on Machine Learning Atlanta, Georgia,USA*, *28*.

Fierens, D., Van Den Broeck, G., & Renkens, J. (2013). Inference and Learning in Probabilistic Logic Programs using Weighted Boolean Formulas. *Theory and Practice of Logic Programming*, *15:3*, 358–401.

Gamper, J. (2015). Programming Paradigms: Unit 1 — Introduction and Basic Concepts, 1–33.

Gao, J., Aksoy, B. A., Dogrusoz, U., Dresdner, G., Gross, B., Sumer, S. O., … Schultz, N. (2013). Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal. *Science Signaling*, *6*(269), 1–34.

Hewitt, C. (2008). Development of Logic Programming: What went wrong, what was done about it, and what it means for the future. *Aaai.Org*, 4–11.

Hsu, Y. C., Hsiao, Y. T., Kao, T. Y., Chang, J. G., & Shieh, G. S. (2017). Detection of Somatic Mutations in Exome Sequencing of Tumor-only Samples. *Scientific Reports*, *7*(1), 1–9.

Illumina. (2015a). Illumina Methylation BeadChips Achieve Breadth of Coverage Using 2 Infinium Chemistries. *Illumina Inc.*, 2–5.

Illumina. (2015b). Somatic Variant Discovery in Cancer, (Figure 2), 1–4.

Katona, B. W., & Rustgi, A. K. (2017). Gastric Cancer Genomics: Advances and Future Directions. *Cellular and Molecular Gastroenterology and Hepatology*, *3*(2), 211–217.

Kowalski, R. a. (1988). The early years of logic programming. *Communications of the ACM*, *31*(I), 38–43.

Kristensen, V. N., Lingjærde, O. C., Russnes, H. G., Vollan, H. K. M., Frigessi, A., & Børresen-Dale, A.-L. (2014). Principles and methods of integrative genomic analyses in cancer. *Nature Reviews Cancer*, *14*(5), 299–313.

Kurdyukov, S., & Bullock, M. (2016). DNA Methylation Analysis: Choosing the Right Method.

Biology, *5*(1), 3. https://doi.org/10.3390/biology5010003

Liang, H., Cheung, L. W. T., Li, J., Ju, Z., Yu, S., Stemke-hale, K., ... Mills, G. B. (2012). Whole-exome sequencing combined with functional genomics reveals novel candidate driver cancer genes in endometrial cancer, 2120–2129.

Lin, Y., Wu, Z., Guo, W., & Li, J. (2015). Gene mutations in gastric cancer: a review of recent next-generation sequencing studies. *Tumor Biology*, *36*(10), 7385–7394.

Liu, B., Hu, F., Zhang, Q., Hu, H., & Ye, Z. (2018). Genomic landscape and mutational impacts of recurrently mutated genes in cancers, (May), 1–14.

Lloyd, J. W. (1983). *Fundations of Logic Programming*.

Magi, A., Tattini, L., Cifola, I., D'Aurizio, R., Benelli, M., Mangano, EM., Battaglia, C., Bonora, E., Kurg, A., Giusti, B., Romeo, G., Pippucci, T., De, B., Abbate, R., Gensini, GF. (2014). EXCAVATOR: detecting copy number variants from whole-exome sequencing data.

Mantadelis, T., & Rocha, R. (2017). Using Iterative Deepening for Probabilistic Logic Inference, (i).

McCord, M. C. (1989). Design of LMT: a Prolog-based Machine Translation System. *Computational Linguistics*, *15*, 33–35.

Meaburn, E., & Schulz, R. (2012). Next generation sequencing in epigenetics: Insights and challenges. *Seminars in Cell and Developmental Biology*, *23*(2), 192–199.

Moore, L. D., Le, T., & Fan, G. (2013). DNA methylation and its basic function. *Neuropsychopharmacology*, *38*(1), 23–38.

Morange, M. (1999). Temporal regulation of gene expression. *Journal de La Societe de Biologie*, *193*(3), 395–400.

Nagini, S. (2012). Carcinoma of the stomach: A review of epidemiology, pathogenesis, molecular genetics and chemoprevention. *World Journal of Gastrointestinal Oncology*, *4*(7), 156.

Om, J. E. (2011). The phi-coefficient, the tetrachoric correlation coefficient, and the pearson-yule debate ¨, 1–19.

Ong, I. M., & Lewis, J. A. (2012). A Problog Model For Analyzing Gene Regulatory Networks.

Pan, X., Ji, X., Zhang, R., Zhou, Z., & Zhong, Y. (2018). Landscape of somatic mutations in gastric cancer assessed using next - generation sequencing analysis, 4863–4870.

Perez-Iratxeta, C., Bork, P., & Andrade, M. A. (2002). Association of genes to genetically inherited diseases using data mining. *Nature Genetics*, *31*(3), 316–319.

Predicates, M. B. (2001). Programming in Logic : Prolog, 1–37.

Przytycki, P. F., & Singh, M. (2017). Differential analysis between somatic mutation and germline variation profiles reveals cancer-related genes. *Genome Medicine*, *9*(1), 1–11.

Qu, Y., Dang, S., & Hou, P. (2013). Gene methylation in gastric cancer. *Clinica Chimica Acta*, *424*, 53–65.

Riquelme, I., Tapia, O., Espinoza, J. A., & Leal, P. (2016). Tissues and cell lines The Gene Expression Status of the PI3K / AKT / mTOR Pathway in Gastric Cancer Tissues and Cell Lines. *Pathology & Oncology Research*, (May).

Sadikovic, B., Al-Romaih, K., Squire, J. a, & Zielenska, M. (2008). Cause and consequences of genetic and epigenetic alterations in human cancer. *Current Genomics*, *9*(6), 394–408.

Sanger Institute. (2017). Sanger Institute. Retrieved from https://www.sanger.ac.uk/

Sharp, R. (1988). CAT2 — Implementing a formalism for multi-lingual MT. *2nd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language, 3-6 June 1988*.

Shen, R., Olshen, A. B., & Ladanyi, M. (2009). Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, *25*(22), 2906–2912.

Siegmund, K. D., & Laird, P. W. (2002). Analysis of complex methylation data. *Methods*, *27*(2), 170–178.

Soneson, C., Love, M. I., & Robinson, M. D. (2016). Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. *F1000Research*, *4*(2), 1521.

Spreitzhofer, G. (1997). Application of post-processing tools to improve visualisation and quality of numerical short-range predictions over Central Europe. *Meteorological Applications*, *4*(3),

Sterling, L. S., & Shapiro Ehud Y. (1999). The Art of Prolog.

Stewart, B. W., & Wild, C. P. (2014). World cancer report 2014. *World Health Organization*, 1–2.

Sud, A., Kinnersley, B., & Houlston, R. S. (2017). Genome-wide association studies of cancer: current insights and future perspectives. *Nature Reviews Cancer*, nrc.2017.82.

TCGA Data Portal. (2017). GDC Data Portal. Retrieved from https://portal.gdc.cancer.gov/

The Cancer Genome Atlas - National Human Genome Research, I. (2017). The Cancer Genome Atlas - National Human Genome Research Institute (NHGRI). Retrieved from

Tsadiras, A. (2009). Using Prolog for Developing Real World Artificial Intelligence Applications, 3960–3962.

Tzafestas, S. (1995). Simply logical: Intelligent reasoning by example. *Data & Knowledge Engineering*, *14*, 290.

Wang, J., Stemmermann, G. N., & Noffsinger, A. (2003). TP53 and Gastric Carcinoma : A Review, *270*, 258–270.

White, R. D. (1989). Foundations of logic programming - SCAN Chapter 1/2. *Artificial Intelligence in Medicine*, *1*(3), 147.

Wu, M.-S., Lin, Y.-S., Chang, Y.-T., Shun, C.-T., Lin, M.-T., & Lin, J.-T. (2005). Gene expression profiling of gastric cancer by microarray combined with laser capture microdissection. *World Journal of Gastroenterology*, *11*(47), 7405–12.

Yamashita, K., Sakuramoto, S., & Watanabe, M. (2011). Genomic and epigenetic profiles of gastric cancer: Potential diagnostic and therapeutic applications. *Surgery Today*, *41*(1), 24–38.

Yang, Y., Dong, X., Xie, B., Ding, N., Chen, J., Li, Y., ... Fang, X. (2015). Databases and web tools for cancer genomics study. *Genomics, Proteomics and Bioinformatics*, *13*(1), 46–50.

Appendix I

ProbLog Program Example

The following simple ProbLog program example was retrieved and adapted from (DTAI, 2015) and it represents an experiment of tossing two coins, one fair and other biased. The fair coin has 0.5 probability of land on head and the biased coin has 0.6 probability of land on head. Those are the facts. There is also a rule that represents a case when both coins land on heads.

```
% Probabilistic facts:
0.5::heads1.
0.6::heads2.
% Rules:
twoHeads :- heads1, heads2.

% Queries:
query(heads1).
query(heads2).
query(twoHeads).
```

The first two queries are performed in order to obtain the probability of getting head when the fair coin is tossed and when the biased coin is tossed. The third query aims to obtain the probability of both coins landing on heads.

```
Query Probability
Heads1 10.5
Heads2 20.6
twoHeads 0.3
```

The first two queries are simple questions once the answer is, immediately, represented in the KB as a fact and the result will be the attached probabilities to the respective fact.

In order to be true, the third query implies that the fair coin land on head and the biased coin also land on head. Therefore, ProbLog performs the product of both probabilities (0.6*0.5).

Table 9 – *ProceOmics* output files by available actions.

Menu	Option	Output File
Get Auxiliary Files	1. Patient and its respective Cancer Subtype	pt_cs.csv
	2. Genes To Work	GenesToWork.csv
	3. Ensembl ID to Gene Name	DR_Id2Gene.csv
Metadata	1. Age	pt_age.csv
	2. Country	pt_country.csv
	3. Gender	pt_gender.csv
	4. Lauren Classification	pt_lc.csv
	5. Race	pt_race.csv
	6. Ethnicity	pt_ethnicity.csv
	7. Cancer Stage	pt_stage.csv
	8. Molecular Subtype (CS)	pt_cs.csv
Mutations	1. Patient ID + Mutated Gene + Variant Classification (N_RD)	pt_gene_mutclass.csv
	2. Patient ID + Mutated Gene + Variant Classification (RD)	DR_pt_gene_mutclass.csv
	3. Patient ID + Mutated Gene (N_RD)	pt_mutgene.csv
	4. Patient ID + Mutated Gene (RD)	DR_pt_mutgene.csv
KEEG Pathways	1. Gene Name + KEEG Pathway	kegg.csv
GO Terms	1. Ensembl ID to GO Term	GOids.csv
Id2Gene	1. Ensembl ID to Gene Name (N_RD)	Id2Gene.csv
	2. Ensembl ID to Gene Name (RD)	DR_Id2Gene.csv
Metadata Associated Probabilities	1. Cancer Subtype + Probability	cs_prob.csv
	2. Cancer Subtype + Age + Probability	cs_age_prob.csv
	3. Cancer Subtype + Country + Probability	cs_country_prob.csv
	4. Cancer Subtype + Gender + Probability	cs_gender_prob.csv
	5. Cancer Subtype + Lauren Classification + Probability	cs_lc_prob.csv
	6. Cancer Subtype + Race + Probability	cs_race_prob.csv
	7. Cancer Subtype + Ethnicity + Probability	cs_ethnicity_prob.csv
	8. Cancer Subtype + Cancer Stage + Probability	cs_stage_prob.csv

Table 9 – *ProceOmics* output files by available actions (continuation).

Menu	Option	Output File
Gene Mutations Associated Probabilities	1. Mutated Gene + Cancer Subtype + Probability (N_RD)	mutgene_cs_prob.csv
	2. Mutated Gene + Cancer Subtype + Probability (RD)	DR_mutgene_cs_prob.csv
	3. Mutated Gene + Variant Classification + Probability (N_RD)	mutgene_vc_prob.csv
	4. Mutated Gene + Variant Classification + Probability (RD)	DR_mutgene_vc_prob.csv
Methylation Associated Probabilities	1. Gene + Cancer Subtype + Probability (RD)	DR_metgene_cs.csv
Gene Expression Associated Probabilities	1. GEM + Cancer Subtype + Expression Profile + Probability	DR_GeneExpression_cs.csv

Table 10 - Developed genetic expression profile scale.

FPKM values	Applied String	Applied String Description
0	non_exp	Non expressed
]0, 1[	low_exp	Low expressed
[1, 10[	norm_exp	Normally expressed
[10, 100[	high_exp	Highly expressed
> 100	very_high_exp	Very highly expressed

Table 11 - Applied data processing steps.

Menu	Option	Data Processing Steps
Metadata	1. Age	1, 2, 3, 5
	2. Country	1, 2, 3, 4, 5
	3. Gender	1, 2,3, 4, 5
	4. Lauren Classification	1, 2, 3, 4, 5
	5. Race	1, 2, 3, 4, 5
	6. Ethnicity	1, 2, 3, 4, 5
	7. Cancer Stage	1, 2, 3, 4, 5
	8. Molecular Subtype (CS)	1, 2, 3, 4, 5
Mutations	1. Patient ID + Mutated Gene + Variant Classification (N_RD)	1, 2, 3, 4, 5, 6, 7
	2. Patient ID + Mutated Gene + Variant Classification (RD)	1, 2, 3, 4, 5, 6, 7, 8
	3. Patient ID + Mutated Gene (N_RD)	1, 2, 3, 4, 5, 6, 7
	4. Patient ID + Mutated Gene (RD)	1, 2, 3, 4 ,5, 6, 7, 8
KEEG Pathways	1. Gene Name + KEEG Pathway	5, 18, 19
GO Terms	1. Ensembl ID to GO Term	5, 19, 20
Id2Gene	1. Ensembl ID to Gene Name (N_RD)	1, 5, 6
	2. Ensembl ID to Gene Name (RD)	1, 5, 6, 8
Metadata Associated Probabilities	1. Cancer Subtype + Age + Probability	1, 2, 3, 4, 5, 6, 21, 21, 23, 24
Gene Mutations Associated Probabilities	1. Mutated Gene + Cancer Subtype + Probability (N_RD)	1, 2, 3, 5, 6, 11, 19, 21, 25, 26
	2. Mutated Gene + Cancer Subtype + Probability (RD)	1, 2, 3, 5, 6, 8, 11, 19, 21, 25, 26
	3. Mutated Gene + Variant Classification + Probability (N_RD)	1, 2, 3, 5, 6, 7, 28, 29
	4. Mutated Gene + Variant Classification + Probability (RD)	1, 2, 3, 5, 6, 7, 8, 28, 29
Methylation Associated Probabilities	1. Gene + Cancer Subtype + Probability (RD)	1, 2, 3, 5, 6, 8, 9, 10, 11
Gene Expression Associated Probabilities	1. GEM + Cancer Subtype + Expression Profile + Probability	1, 2, 5, 8, 11, 12, 13, 14, 15, 16, 17, 21

Table 12 - Processing steps enumeration.

Data Processing Steps	Reference Numbers
Select the interest features/columns	1
Truncate the patient/Tumor_Sample_Barcode/Ensembl IDs	2
Remove or replace omitted values In case of NA's replacement, the value to be replaced assume the most common value when the data type is non-numeric or the mean of all the non-Na values or a well-marked value when the data type is numeric	3
Convert the non-numeric values to lower case	4
Create a csv file to be loaded into the problog KB	5
Removal of duplicated values	6
Replace ambiguous characters on the Variant_Classification column In example, 5'UTR → 5UTR	7
Data dimensionality reduction This approach allows to get genes in a specific threshold which improve the problog KB efficiency and velocity. Those genes are stored in the GenesToWork.csv file. The threshold value is given by the user. After the selection of the genes to work with, this approach can also be applied to several dataframes of different information, to retrieve only the features to the previously genes selected.	8
Unlist the column values	9
Replace the multiple associated β -values of each gene by the average value	10
Rename and/or reorder columns	11
Transpose the GEM (gene expression matrix) Aims to swap the columns to the index and vice-versa	12
Subset the transposed GEM into N new data frames N corresponds to the number of columns (number of genes). The new data frames contain the same number of rows (samples) as the transposed GEM	13
Create an expression profile scale and apply it to FPKM values The expression profile scale is based on the expression value of each gene in each sample which aims to facilitate the results interpretations and replace as much numeric data as possible in the problog KB. This approach is applied to all the data frames that result from step 13. The expression profile scale is represented at Table 10 in appendix II.	14
FPKM value transformation of GEM values into probability of expression This approach is applied to all the data frames that result from step 13.	15
Concatenate the expression profile and probability value to the respective value This approach is applied to all the data frames that result from step 13	16

Table 12 - Processing steps enumeration (continuation).

Data Processing Steps	Reference Numbers
Merge all sub dataframes into just one	17
Get all the kegg pathways Performed for all genes in the GenesToWork.csv file	18
Convert the final dataframe/structure to the desired structure that is most adequate to the problog KB	19
Get all the GO Terms Performed for all genes in the GenesToWork.csv file	20
Merge different dataframes based on column values	21
Calculate the mean and standard deviation to all age values grouped by cancer subtype	22
Duplicate each dataframe row n times This n times duplication is required to further apply a Gaussian distribution. The n is equal to the number of cancer subtypes	23
Apply a Gaussian distribution to all the age values This step uses the step 22 and its applied to step 23	24
Perform a cross validation table Allows to obtain the occurrence number of gene mutations in each cancer subtype	25
Count the number of occurrence of each cancer subtype	26
Probability calculation of each gene be mutated in each cancer subtype This step is applied to steps 25 and 26 The probabilities values are obtained by dividing the occurrence number of a mutated gene in a subtype by the respective cancer subtype counts in the original dataframe	27
Remove undesired values	28
Probability calculation of each gene be mutated with a specific variant classification The probabilities values are obtained by dividing the occurrence number of a mutated gene in a variant classification by the respective variant classification counts in all the dataframe	29

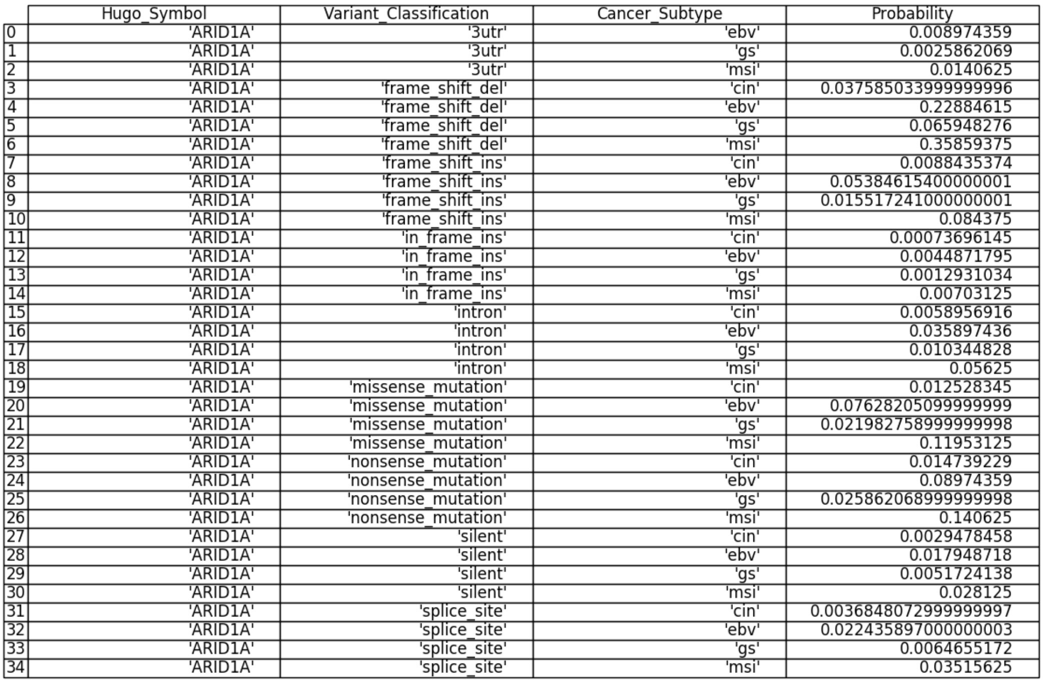
Appendix IV

Case Study I

Approach I

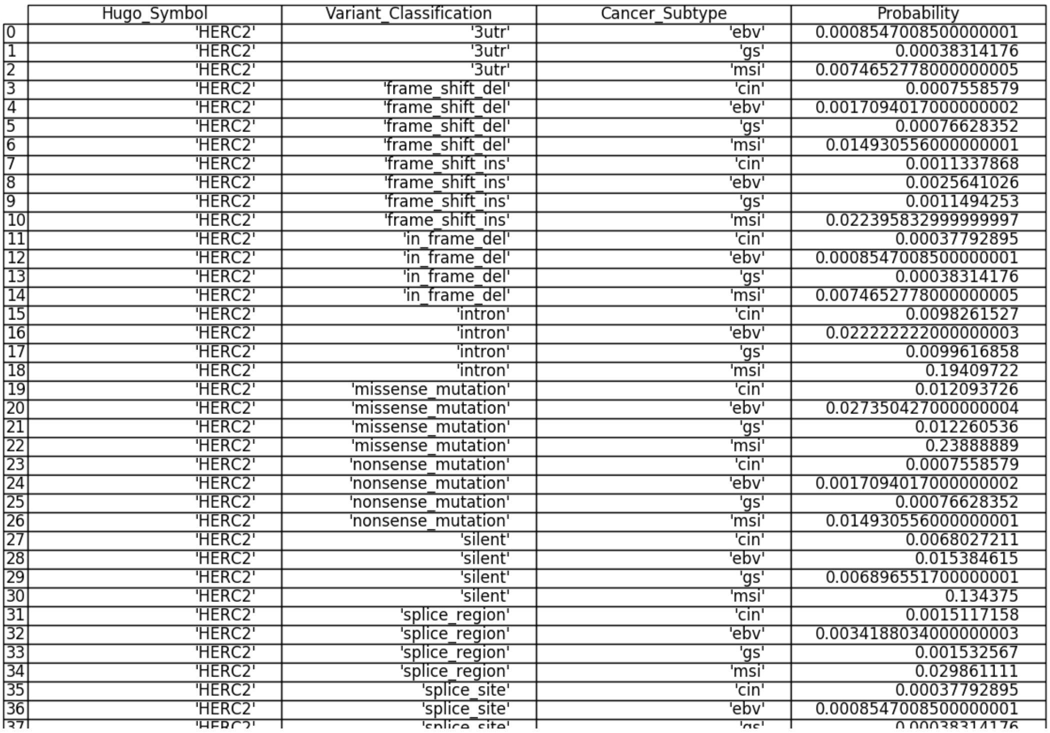
Queries

- query(mutgene_vc_cs_prob('ARID1A',VC,CS)). (1)
- query(mutgene_vc_cs_prob('HERC2',VC,CS)). (2)
- query(mutgene_vc_cs_prob('LAMA1',VC,CS)). (3)
- query(mutgene_vc_cs_prob('PIK3CA',VC,CS)). (4)
- query(mutgene_vc_cs_prob('RERE',VC,CS)). (5)
- query(mutgene_vc_cs_prob('TP53',VC,CS)). (6)



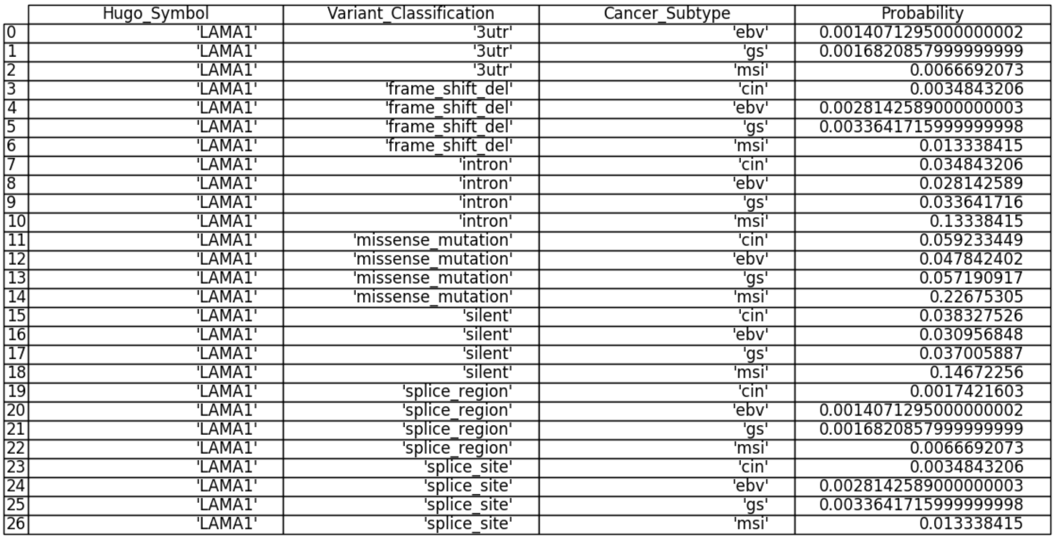
	Hugo_Symbol	Variant_Classification	Cancer_Subtype	Probability
0	'ARID1A'	'3utr'	'ebv'	0.008974359
1	'ARID1A'	'3utr'	'gs'	0.0025862069
2	'ARID1A'	'3utr'	'msi'	0.0140625
3	'ARID1A'	'frame_shift_del'	'cin'	0.037585033999999996
4	'ARID1A'	'frame_shift_del'	'ebv'	0.22884615
5	'ARID1A'	'frame_shift_del'	'gs'	0.065948276
6	'ARID1A'	'frame_shift_del'	'msi'	0.35859375
7	'ARID1A'	'frame_shift_ins'	'cin'	0.0088435374
8	'ARID1A'	'frame_shift_ins'	'ebv'	0.05384615400000001
9	'ARID1A'	'frame_shift_ins'	'gs'	0.015517241000000001
10	'ARID1A'	'frame_shift_ins'	'msi'	0.084375
11	'ARID1A'	'in_frame_ins'	'cin'	0.00073696145
12	'ARID1A'	'in_frame_ins'	'ebv'	0.0044871795
13	'ARID1A'	'in_frame_ins'	'gs'	0.0012931034
14	'ARID1A'	'in_frame_ins'	'msi'	0.00703125
15	'ARID1A'	'intron'	'cin'	0.0058956916
16	'ARID1A'	'intron'	'ebv'	0.035897436
17	'ARID1A'	'intron'	'gs'	0.010344828
18	'ARID1A'	'intron'	'msi'	0.05625
19	'ARID1A'	'missense_mutation'	'cin'	0.012528345
20	'ARID1A'	'missense_mutation'	'ebv'	0.07628205099999999
21	'ARID1A'	'missense_mutation'	'gs'	0.021982758999999998
22	'ARID1A'	'missense_mutation'	'msi'	0.11953125
23	'ARID1A'	'nonsense_mutation'	'cin'	0.014739229
24	'ARID1A'	'nonsense_mutation'	'ebv'	0.08974359
25	'ARID1A'	'nonsense_mutation'	'gs'	0.025862068999999998
26	'ARID1A'	'nonsense_mutation'	'msi'	0.140625
27	'ARID1A'	'silent'	'cin'	0.0029478458
28	'ARID1A'	'silent'	'ebv'	0.017948718
29	'ARID1A'	'silent'	'gs'	0.0051724138
30	'ARID1A'	'silent'	'msi'	0.028125
31	'ARID1A'	'splice_site'	'cin'	0.0036848072999999997
32	'ARID1A'	'splice_site'	'ebv'	0.022435897000000003
33	'ARID1A'	'splice_site'	'gs'	0.0064655172
34	'ARID1A'	'splice_site'	'msi'	0.03515625

Figure 10 - Query (1) results from approach I of case study I.



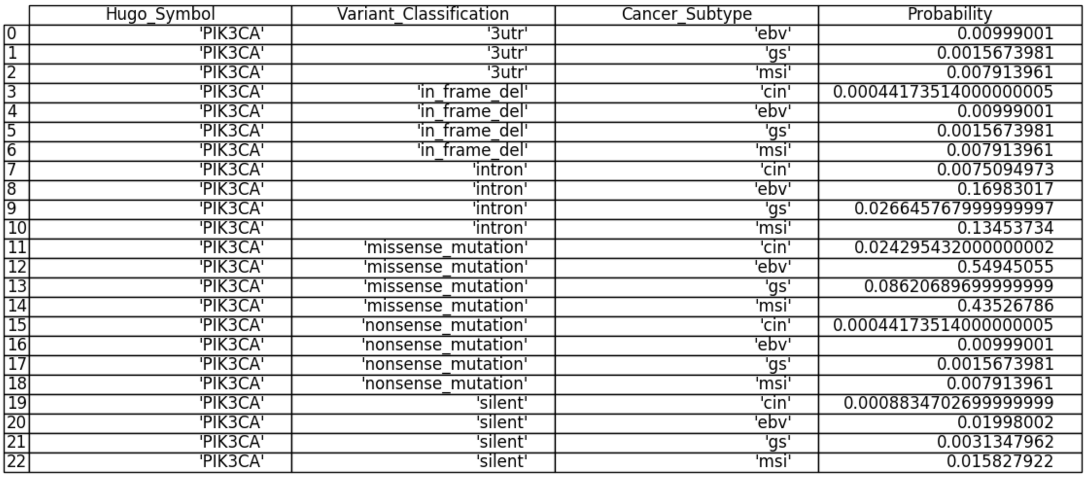
	Hugo_Symbol	Variant_Classification	Cancer_Subtype	Probability
0	'HERC2'	'3utr'	'ebv'	0.0008547008500000001
1	'HERC2'	'3utr'	'gs'	0.00038314176
2	'HERC2'	'3utr'	'msi'	0.0074652778000000005
3	'HERC2'	'frame_shift_del'	'cin'	0.0007558579
4	'HERC2'	'frame_shift_del'	'ebv'	0.0017094017000000002
5	'HERC2'	'frame_shift_del'	'gs'	0.00076628352
6	'HERC2'	'frame_shift_del'	'msi'	0.014930556000000001
7	'HERC2'	'frame_shift_ins'	'cin'	0.0011337868
8	'HERC2'	'frame_shift_ins'	'ebv'	0.0025641026
9	'HERC2'	'frame_shift_ins'	'gs'	0.0011494253
10	'HERC2'	'frame_shift_ins'	'msi'	0.022395832999999997
11	'HERC2'	'in_frame_del'	'cin'	0.00037792895
12	'HERC2'	'in_frame_del'	'ebv'	0.0008547008500000001
13	'HERC2'	'in_frame_del'	'gs'	0.00038314176
14	'HERC2'	'in_frame_del'	'msi'	0.0074652778000000005
15	'HERC2'	'intron'	'cin'	0.0098261527
16	'HERC2'	'intron'	'ebv'	0.022222222000000003
17	'HERC2'	'intron'	'gs'	0.0099616858
18	'HERC2'	'intron'	'msi'	0.19409722
19	'HERC2'	'missense_mutation'	'cin'	0.012093726
20	'HERC2'	'missense_mutation'	'ebv'	0.027350427000000004
21	'HERC2'	'missense_mutation'	'gs'	0.012260536
22	'HERC2'	'missense_mutation'	'msi'	0.23888889
23	'HERC2'	'nonsense_mutation'	'cin'	0.0007558579
24	'HERC2'	'nonsense_mutation'	'ebv'	0.0017094017000000002
25	'HERC2'	'nonsense_mutation'	'gs'	0.00076628352
26	'HERC2'	'nonsense_mutation'	'msi'	0.014930556000000001
27	'HERC2'	'silent'	'cin'	0.0068027211
28	'HERC2'	'silent'	'ebv'	0.015384615
29	'HERC2'	'silent'	'gs'	0.006896551700000001
30	'HERC2'	'silent'	'msi'	0.134375
31	'HERC2'	'splice_region'	'cin'	0.0015117158
32	'HERC2'	'splice_region'	'ebv'	0.0034188034000000003
33	'HERC2'	'splice_region'	'gs'	0.001532567
34	'HERC2'	'splice_region'	'msi'	0.029861111
35	'HERC2'	'splice_site'	'cin'	0.00037792895
36	'HERC2'	'splice_site'	'ebv'	0.0008547008500000001
37	'HERC2'	'splice_site'	'gs'	0.00038314176

Figure 11 - Query (2) results from approach I of case study I.
This figure is only a portion of the total query output.



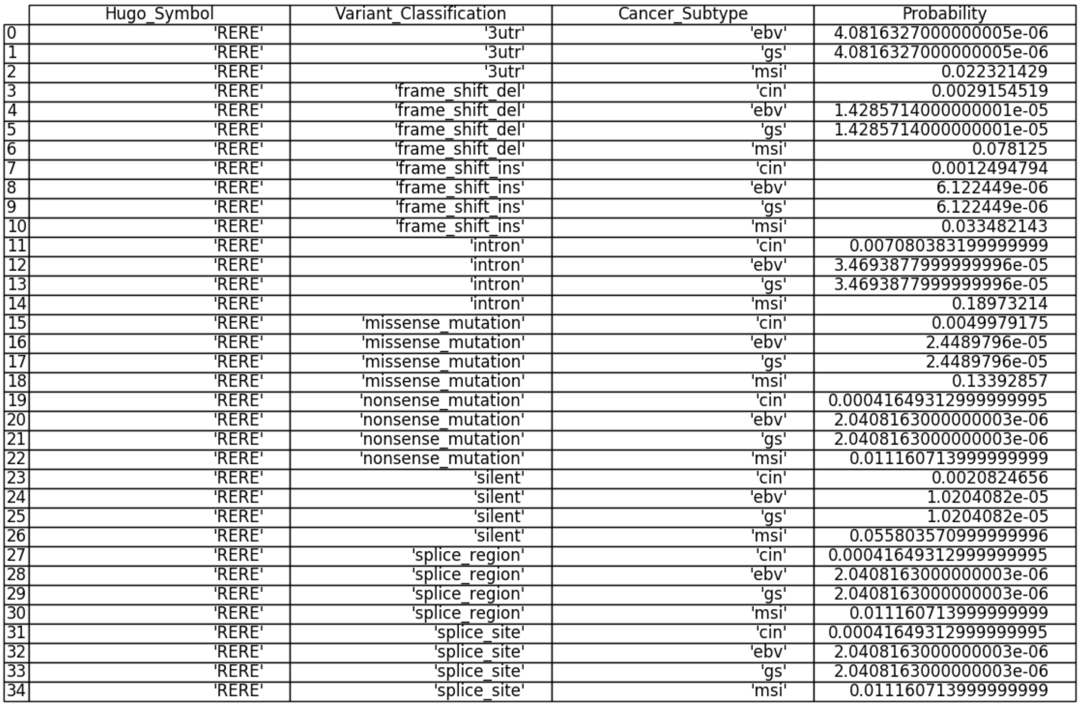
	Hugo_Symbol	Variant_Classification	Cancer_Subtype	Probability
0	'LAMA1'	'3utr'	'ebv'	0.0014071295000000002
1	'LAMA1'	'3utr'	'gs'	0.0016820857999999999
2	'LAMA1'	'3utr'	'msi'	0.0066692073
3	'LAMA1'	'frame_shift_del'	'cin'	0.0034843206
4	'LAMA1'	'frame_shift_del'	'ebv'	0.0028142589000000003
5	'LAMA1'	'frame_shift_del'	'gs'	0.0033641715999999998
6	'LAMA1'	'frame_shift_del'	'msi'	0.013338415
7	'LAMA1'	'intron'	'cin'	0.034843206
8	'LAMA1'	'intron'	'ebv'	0.028142589
9	'LAMA1'	'intron'	'gs'	0.033641716
10	'LAMA1'	'intron'	'msi'	0.13338415
11	'LAMA1'	'missense_mutation'	'cin'	0.059233449
12	'LAMA1'	'missense_mutation'	'ebv'	0.047842402
13	'LAMA1'	'missense_mutation'	'gs'	0.057190917
14	'LAMA1'	'missense_mutation'	'msi'	0.22675305
15	'LAMA1'	'silent'	'cin'	0.038327526
16	'LAMA1'	'silent'	'ebv'	0.030956848
17	'LAMA1'	'silent'	'gs'	0.037005887
18	'LAMA1'	'silent'	'msi'	0.14672256
19	'LAMA1'	'splice_region'	'cin'	0.0017421603
20	'LAMA1'	'splice_region'	'ebv'	0.0014071295000000002
21	'LAMA1'	'splice_region'	'gs'	0.0016820857999999999
22	'LAMA1'	'splice_region'	'msi'	0.0066692073
23	'LAMA1'	'splice_site'	'cin'	0.0034843206
24	'LAMA1'	'splice_site'	'ebv'	0.0028142589000000003
25	'LAMA1'	'splice_site'	'gs'	0.0033641715999999998
26	'LAMA1'	'splice_site'	'msi'	0.013338415

Figure 12 - Query (3) results from approach I of case study I.



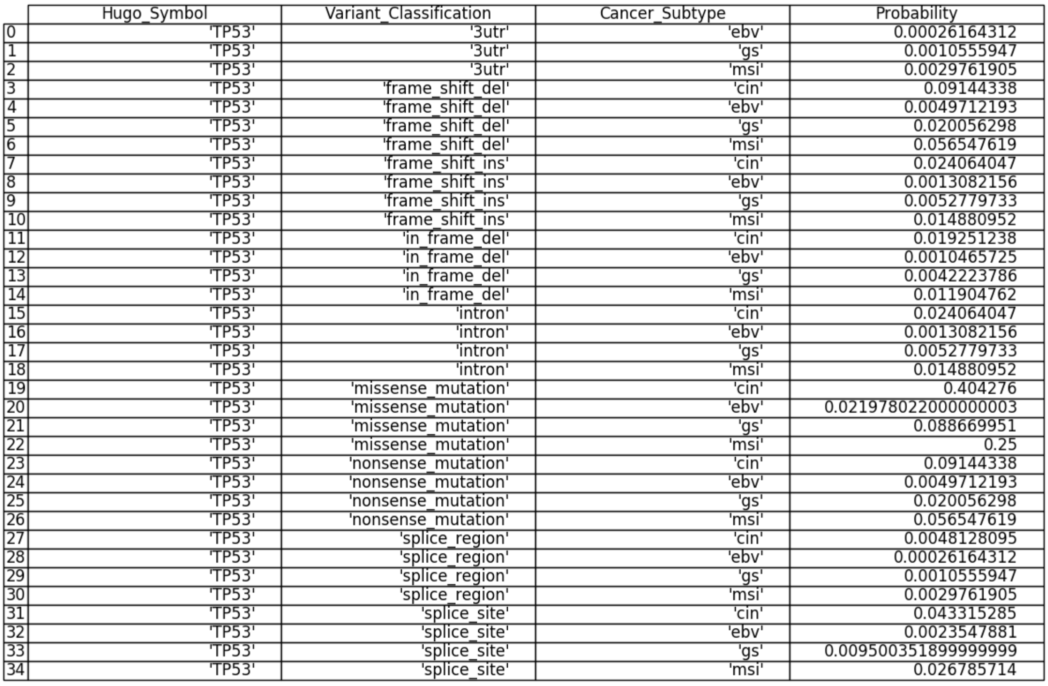
	Hugo_Symbol	Variant_Classification	Cancer_Subtype	Probability
0	'PIK3CA'	'3utr'	'ebv'	0.00999001
1	'PIK3CA'	'3utr'	'gs'	0.0015673981
2	'PIK3CA'	'3utr'	'msi'	0.007913961
3	'PIK3CA'	'in_frame_del'	'cin'	0.00044173514000000005
4	'PIK3CA'	'in_frame_del'	'ebv'	0.00999001
5	'PIK3CA'	'in_frame_del'	'gs'	0.0015673981
6	'PIK3CA'	'in_frame_del'	'msi'	0.007913961
7	'PIK3CA'	'intron'	'cin'	0.0075094973
8	'PIK3CA'	'intron'	'ebv'	0.16983017
9	'PIK3CA'	'intron'	'gs'	0.026645767999999997
10	'PIK3CA'	'intron'	'msi'	0.13453734
11	'PIK3CA'	'missense_mutation'	'cin'	0.024295432000000002
12	'PIK3CA'	'missense_mutation'	'ebv'	0.54945055
13	'PIK3CA'	'missense_mutation'	'gs'	0.08620689699999999
14	'PIK3CA'	'missense_mutation'	'msi'	0.43526786
15	'PIK3CA'	'nonsense_mutation'	'cin'	0.00044173514000000005
16	'PIK3CA'	'nonsense_mutation'	'ebv'	0.00999001
17	'PIK3CA'	'nonsense_mutation'	'gs'	0.0015673981
18	'PIK3CA'	'nonsense_mutation'	'msi'	0.007913961
19	'PIK3CA'	'silent'	'cin'	0.0008834702699999999
20	'PIK3CA'	'silent'	'ebv'	0.01998002
21	'PIK3CA'	'silent'	'gs'	0.0031347962
22	'PIK3CA'	'silent'	'msi'	0.015827922

Figure 13 - Query (4) results from approach I of case study I.



	Hugo_Symbol	Variant_Classification	Cancer_Subtype	Probability
0	'RERE'	'3utr'	'ebv'	4.0816327000000005e-06
1	'RERE'	'3utr'	'gs'	4.0816327000000005e-06
2	'RERE'	'3utr'	'msi'	0.022321429
3	'RERE'	'frame_shift_del'	'cin'	0.0029154519
4	'RERE'	'frame_shift_del'	'ebv'	1.4285714000000001e-05
5	'RERE'	'frame_shift_del'	'gs'	1.4285714000000001e-05
6	'RERE'	'frame_shift_del'	'msi'	0.078125
7	'RERE'	'frame_shift_ins'	'cin'	0.0012494794
8	'RERE'	'frame_shift_ins'	'ebv'	6.122449e-06
9	'RERE'	'frame_shift_ins'	'gs'	6.122449e-06
10	'RERE'	'frame_shift_ins'	'msi'	0.033482143
11	'RERE'	'intron'	'cin'	0.007080383199999999
12	'RERE'	'intron'	'ebv'	3.4693877999999996e-05
13	'RERE'	'intron'	'gs'	3.4693877999999996e-05
14	'RERE'	'intron'	'msi'	0.18973214
15	'RERE'	'missense_mutation'	'cin'	0.0049979175
16	'RERE'	'missense_mutation'	'ebv'	2.4489796e-05
17	'RERE'	'missense_mutation'	'gs'	2.4489796e-05
18	'RERE'	'missense_mutation'	'msi'	0.13392857
19	'RERE'	'nonsense_mutation'	'cin'	0.00041649312999999995
20	'RERE'	'nonsense_mutation'	'ebv'	2.0408163000000003e-06
21	'RERE'	'nonsense_mutation'	'gs'	2.0408163000000003e-06
22	'RERE'	'nonsense_mutation'	'msi'	0.011160713999999999
23	'RERE'	'silent'	'cin'	0.0020824656
24	'RERE'	'silent'	'ebv'	1.0204082e-05
25	'RERE'	'silent'	'gs'	1.0204082e-05
26	'RERE'	'silent'	'msi'	0.055803570999999996
27	'RERE'	'splice_region'	'cin'	0.00041649312999999995
28	'RERE'	'splice_region'	'ebv'	2.0408163000000003e-06
29	'RERE'	'splice_region'	'gs'	2.0408163000000003e-06
30	'RERE'	'splice_region'	'msi'	0.011160713999999999
31	'RERE'	'splice_site'	'cin'	0.00041649312999999995
32	'RERE'	'splice_site'	'ebv'	2.0408163000000003e-06
33	'RERE'	'splice_site'	'gs'	2.0408163000000003e-06
34	'RERE'	'splice_site'	'msi'	0.011160713999999999

Figure 14 - Query (5) results from approach I of case study I.



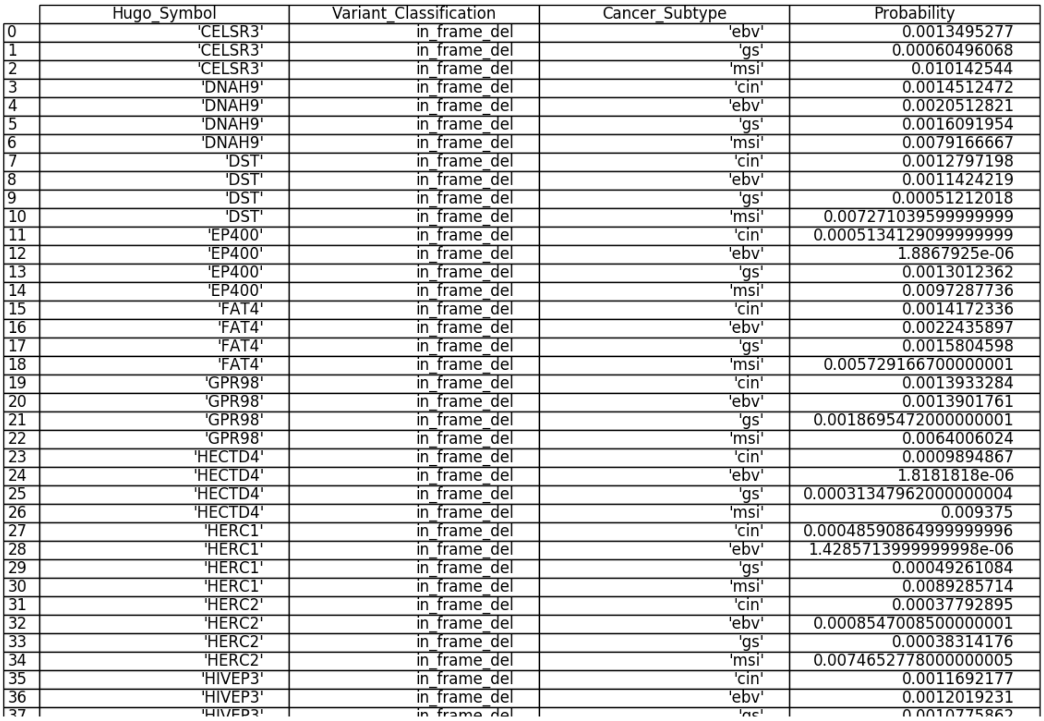
	Hugo_Symbol	Variant_Classification	Cancer_Subtype	Probability
0	'TP53'	'3utr'	'ebv'	0.00026164312
1	'TP53'	'3utr'	'gs'	0.0010555947
2	'TP53'	'3utr'	'msi'	0.0029761905
3	'TP53'	'frame_shift_del'	'cin'	0.09144338
4	'TP53'	'frame_shift_del'	'ebv'	0.0049712193
5	'TP53'	'frame_shift_del'	'gs'	0.020056298
6	'TP53'	'frame_shift_del'	'msi'	0.056547619
7	'TP53'	'frame_shift_ins'	'cin'	0.024064047
8	'TP53'	'frame_shift_ins'	'ebv'	0.0013082156
9	'TP53'	'frame_shift_ins'	'gs'	0.0052779733
10	'TP53'	'frame_shift_ins'	'msi'	0.014880952
11	'TP53'	'in_frame_del'	'cin'	0.019251238
12	'TP53'	'in_frame_del'	'ebv'	0.0010465725
13	'TP53'	'in_frame_del'	'gs'	0.0042223786
14	'TP53'	'in_frame_del'	'msi'	0.011904762
15	'TP53'	'intron'	'cin'	0.024064047
16	'TP53'	'intron'	'ebv'	0.0013082156
17	'TP53'	'intron'	'gs'	0.0052779733
18	'TP53'	'intron'	'msi'	0.014880952
19	'TP53'	'missense_mutation'	'cin'	0.404276
20	'TP53'	'missense_mutation'	'ebv'	0.021978022000000003
21	'TP53'	'missense_mutation'	'gs'	0.088669951
22	'TP53'	'missense_mutation'	'msi'	0.25
23	'TP53'	'nonsense_mutation'	'cin'	0.09144338
24	'TP53'	'nonsense_mutation'	'ebv'	0.0049712193
25	'TP53'	'nonsense_mutation'	'gs'	0.020056298
26	'TP53'	'nonsense_mutation'	'msi'	0.056547619
27	'TP53'	'splice_region'	'cin'	0.0048128095
28	'TP53'	'splice_region'	'ebv'	0.00026164312
29	'TP53'	'splice_region'	'gs'	0.0010555947
30	'TP53'	'splice_region'	'msi'	0.0029761905
31	'TP53'	'splice_site'	'cin'	0.043315285
32	'TP53'	'splice_site'	'ebv'	0.0023547881
33	'TP53'	'splice_site'	'gs'	0.009500351899999999
34	'TP53'	'splice_site'	'msi'	0.026785714

Figure 15 - Query (6) results from approach I of case study I.

Approach II

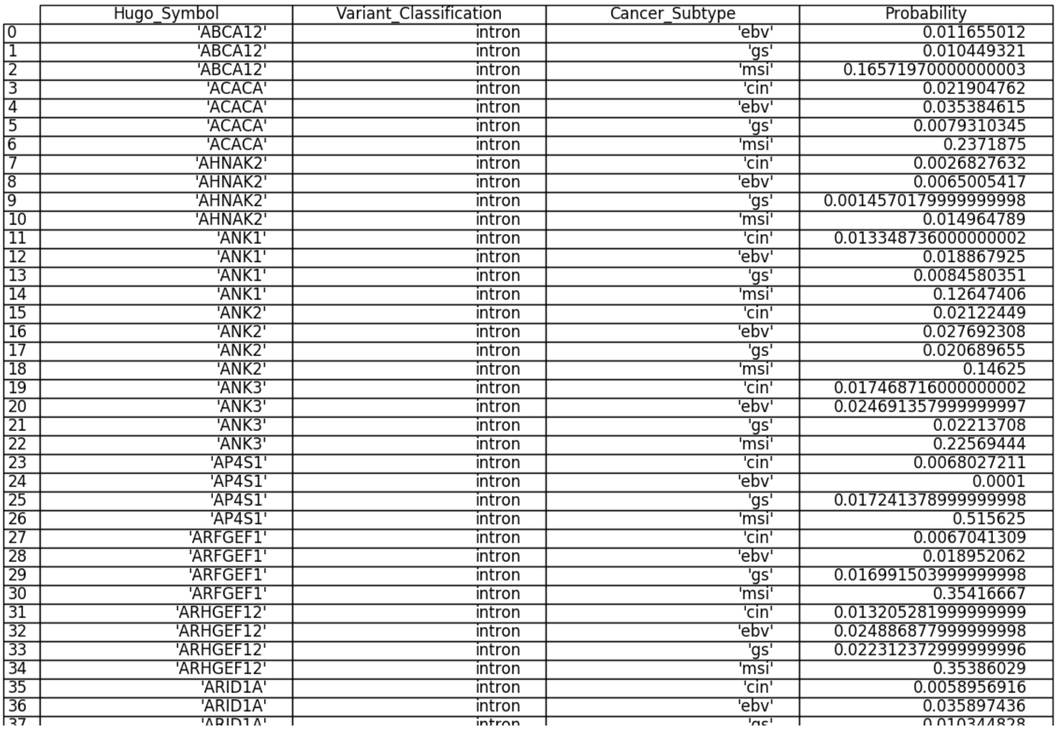
Queries

- query(mutgene_vc_cs_prob(GENE,in_frame_del,CS)). (1)
- query(mutgene_vc_cs_prob(GENE,intron,CS)). (2)
- query(mutgene_vc_cs_prob(GENE,splice_region,CS)). (3)



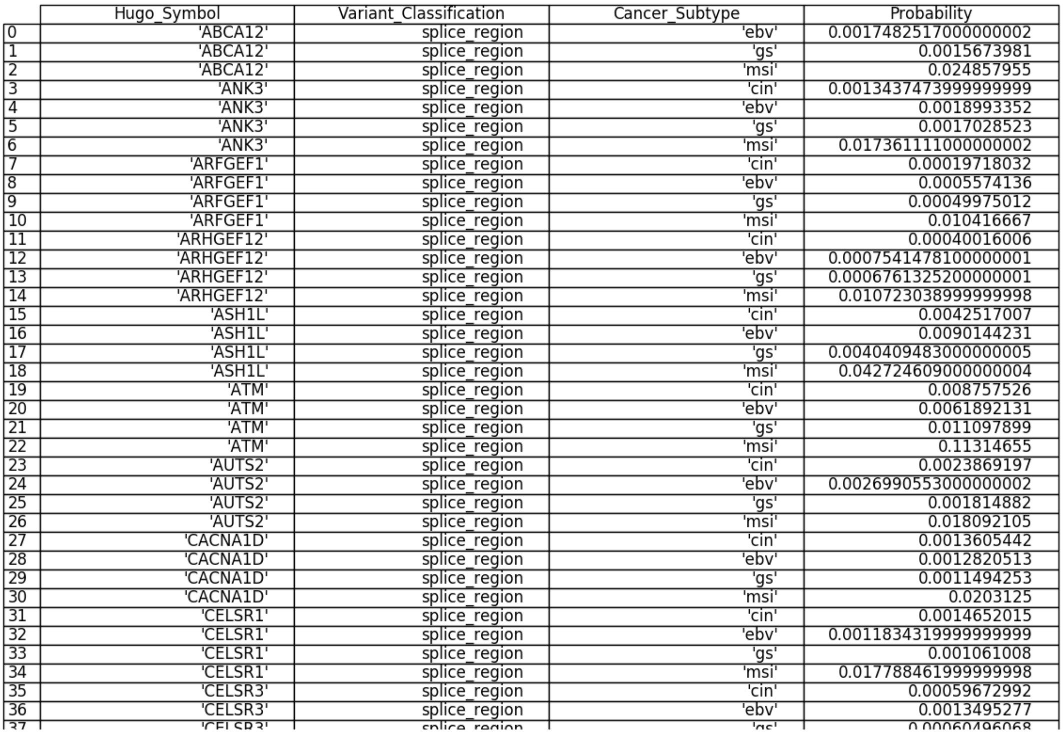
	Hugo_Symbol	Variant_Classification	Cancer_Subtype	Probability
0	'CELSR3'	in_frame_del	'ebv'	0.0013495277
1	'CELSR3'	in_frame_del	'gs'	0.00060496068
2	'CELSR3'	in_frame_del	'msi'	0.010142544
3	'DNAH9'	in_frame_del	'cin'	0.0014512472
4	'DNAH9'	in_frame_del	'ebv'	0.0020512821
5	'DNAH9'	in_frame_del	'gs'	0.0016091954
6	'DNAH9'	in_frame_del	'msi'	0.0079166667
7	'DST'	in_frame_del	'cin'	0.0012797198
8	'DST'	in_frame_del	'ebv'	0.0011424219
9	'DST'	in_frame_del	'gs'	0.00051212018
10	'DST'	in_frame_del	'msi'	0.007271039599999999
11	'EP400'	in_frame_del	'cin'	0.0005134129099999999
12	'EP400'	in_frame_del	'ebv'	1.8867925e-06
13	'EP400'	in_frame_del	'gs'	0.0013012362
14	'EP400'	in_frame_del	'msi'	0.0097287736
15	'FAT4'	in_frame_del	'cin'	0.0014172336
16	'FAT4'	in_frame_del	'ebv'	0.0022435897
17	'FAT4'	in_frame_del	'gs'	0.0015804598
18	'FAT4'	in_frame_del	'msi'	0.005729166700000001
19	'GPR98'	in_frame_del	'cin'	0.0013933284
20	'GPR98'	in_frame_del	'ebv'	0.0013901761
21	'GPR98'	in_frame_del	'gs'	0.0018695472000000001
22	'GPR98'	in_frame_del	'msi'	0.0064006024
23	'HECTD4'	in_frame_del	'cin'	0.0009894867
24	'HECTD4'	in_frame_del	'ebv'	1.8181818e-06
25	'HECTD4'	in_frame_del	'gs'	0.00031347962000000004
26	'HECTD4'	in_frame_del	'msi'	0.009375
27	'HERC1'	in_frame_del	'cin'	0.00048590864999999996
28	'HERC1'	in_frame_del	'ebv'	1.4285713999999998e-06
29	'HERC1'	in_frame_del	'gs'	0.00049261084
30	'HERC1'	in_frame_del	'msi'	0.0089285714
31	'HERC2'	in_frame_del	'cin'	0.00037792895
32	'HERC2'	in_frame_del	'ebv'	0.0008547008500000001
33	'HERC2'	in_frame_del	'gs'	0.00038314176
34	'HERC2'	in_frame_del	'msi'	0.007465277800000005
35	'HIVEP3'	in_frame_del	'cin'	0.0011692177
36	'HIVEP3'	in_frame_del	'ebv'	0.0012019231
37	'HIVEP3'	in_frame_del	'gs'	0.0010775862

Figure 16 - Query (1) results from approach II of case study I.
This figure is only a portion of the total query output.



	Hugo_Symbol	Variant_Classification	Cancer_Subtype	Probability
0	'ABCA12'	intron	'ebv'	0.011655012
1	'ABCA12'	intron	'gs'	0.010449321
2	'ABCA12'	intron	'msi'	0.16571970000000003
3	'ACACA'	intron	'cin'	0.021904762
4	'ACACA'	intron	'ebv'	0.035384615
5	'ACACA'	intron	'gs'	0.0079310345
6	'ACACA'	intron	'msi'	0.2371875
7	'AHNAK2'	intron	'cin'	0.0026827632
8	'AHNAK2'	intron	'ebv'	0.0065005417
9	'AHNAK2'	intron	'gs'	0.0014570179999999998
10	'AHNAK2'	intron	'msi'	0.014964789
11	'ANK1'	intron	'cin'	0.013348736000000002
12	'ANK1'	intron	'ebv'	0.018867925
13	'ANK1'	intron	'gs'	0.0084580351
14	'ANK1'	intron	'msi'	0.12647406
15	'ANK2'	intron	'cin'	0.02122449
16	'ANK2'	intron	'ebv'	0.027692308
17	'ANK2'	intron	'gs'	0.020689655
18	'ANK2'	intron	'msi'	0.14625
19	'ANK3'	intron	'cin'	0.017468716000000002
20	'ANK3'	intron	'ebv'	0.024691357999999997
21	'ANK3'	intron	'gs'	0.02213708
22	'ANK3'	intron	'msi'	0.22569444
23	'AP4S1'	intron	'cin'	0.0068027211
24	'AP4S1'	intron	'ebv'	0.0001
25	'AP4S1'	intron	'gs'	0.017241378999999998
26	'AP4S1'	intron	'msi'	0.515625
27	'ARFGEF1'	intron	'cin'	0.0067041309
28	'ARFGEF1'	intron	'ebv'	0.018952062
29	'ARFGEF1'	intron	'gs'	0.016991503999999998
30	'ARFGEF1'	intron	'msi'	0.35416667
31	'ARHGEF12'	intron	'cin'	0.013205281999999999
32	'ARHGEF12'	intron	'ebv'	0.024886877999999998
33	'ARHGEF12'	intron	'gs'	0.022312372999999996
34	'ARHGEF12'	intron	'msi'	0.35386029
35	'ARID1A'	intron	'cin'	0.0058956916
36	'ARID1A'	intron	'ebv'	0.035897436
37	'ARID1A'	intron	'gs'	0.010344828

Figure 17 - Query (2) results from approach II of case study I.
This figure is only a portion of the total query output.



	Hugo_Symbol	Variant_Classification	Cancer_Subtype	Probability
0	'ABCA12'	splice_region	'ebv'	0.0017482517000000002
1	'ABCA12'	splice_region	'gs'	0.0015673981
2	'ABCA12'	splice_region	'msi'	0.024857955
3	'ANK3'	splice_region	'cin'	0.0013437473999999999
4	'ANK3'	splice_region	'ebv'	0.0018993352
5	'ANK3'	splice_region	'gs'	0.0017028523
6	'ANK3'	splice_region	'msi'	0.017361111000000002
7	'ARFGEF1'	splice_region	'cin'	0.00019718032
8	'ARFGEF1'	splice_region	'ebv'	0.0005574136
9	'ARFGEF1'	splice_region	'gs'	0.00049975012
10	'ARFGEF1'	splice_region	'msi'	0.010416667
11	'ARHGEF12'	splice_region	'cin'	0.00040016006
12	'ARHGEF12'	splice_region	'ebv'	0.0007541478100000001
13	'ARHGEF12'	splice_region	'gs'	0.0006761325200000001
14	'ARHGEF12'	splice_region	'msi'	0.010723038999999998
15	'ASH1L'	splice_region	'cin'	0.0042517007
16	'ASH1L'	splice_region	'ebv'	0.0090144231
17	'ASH1L'	splice_region	'gs'	0.0040409483000000005
18	'ASH1L'	splice_region	'msi'	0.042724609000000004
19	'ATM'	splice_region	'cin'	0.008757526
20	'ATM'	splice_region	'ebv'	0.0061892131
21	'ATM'	splice_region	'gs'	0.011097899
22	'ATM'	splice_region	'msi'	0.11314655
23	'AUTS2'	splice_region	'cin'	0.0023869197
24	'AUTS2'	splice_region	'ebv'	0.0026990553000000002
25	'AUTS2'	splice_region	'gs'	0.001814882
26	'AUTS2'	splice_region	'msi'	0.018092105
27	'CACNA1D'	splice_region	'cin'	0.0013605442
28	'CACNA1D'	splice_region	'ebv'	0.0012820513
29	'CACNA1D'	splice_region	'gs'	0.0011494253
30	'CACNA1D'	splice_region	'msi'	0.0203125
31	'CELSR1'	splice_region	'cin'	0.0014652015
32	'CELSR1'	splice_region	'ebv'	0.0011834319999999999
33	'CELSR1'	splice_region	'gs'	0.001061008
34	'CELSR1'	splice_region	'msi'	0.017788461999999998
35	'CELSR3'	splice_region	'cin'	0.00059672992
36	'CELSR3'	splice_region	'ebv'	0.0013495277
37	'CELSR3'	splice_region	'gs'	0.00060496068

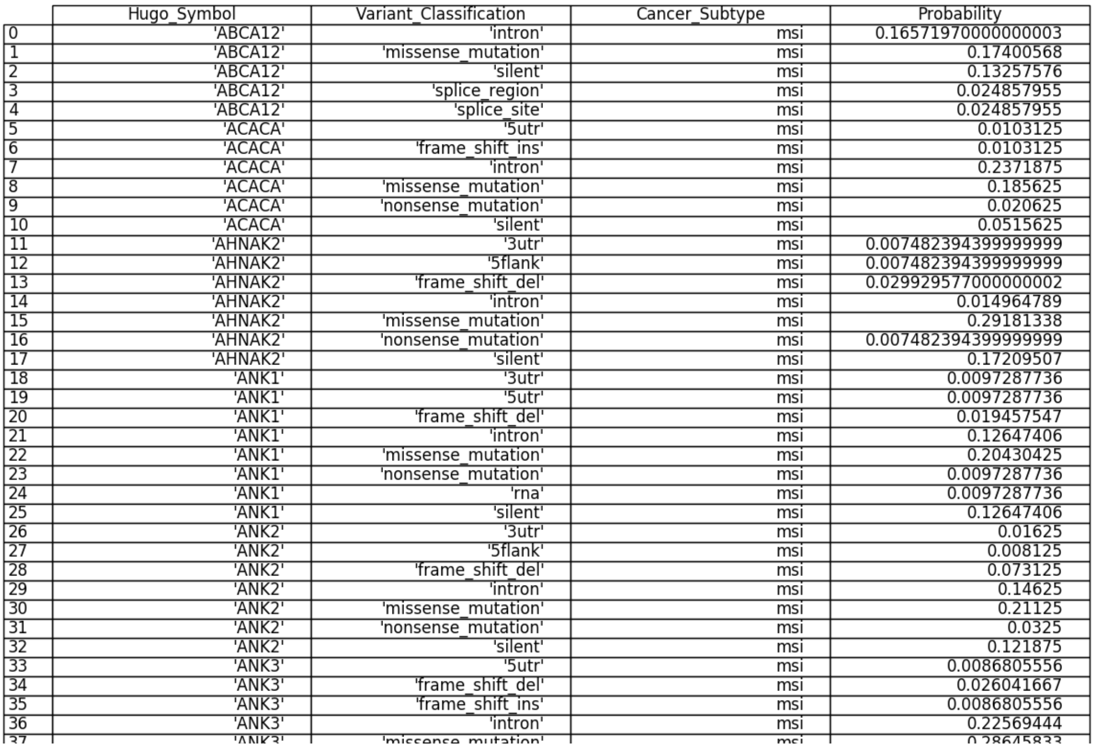
Figure 18 - Query (3) results from approach II of case study I.
This figure is only a portion of the total query output.

Approach III

Queries

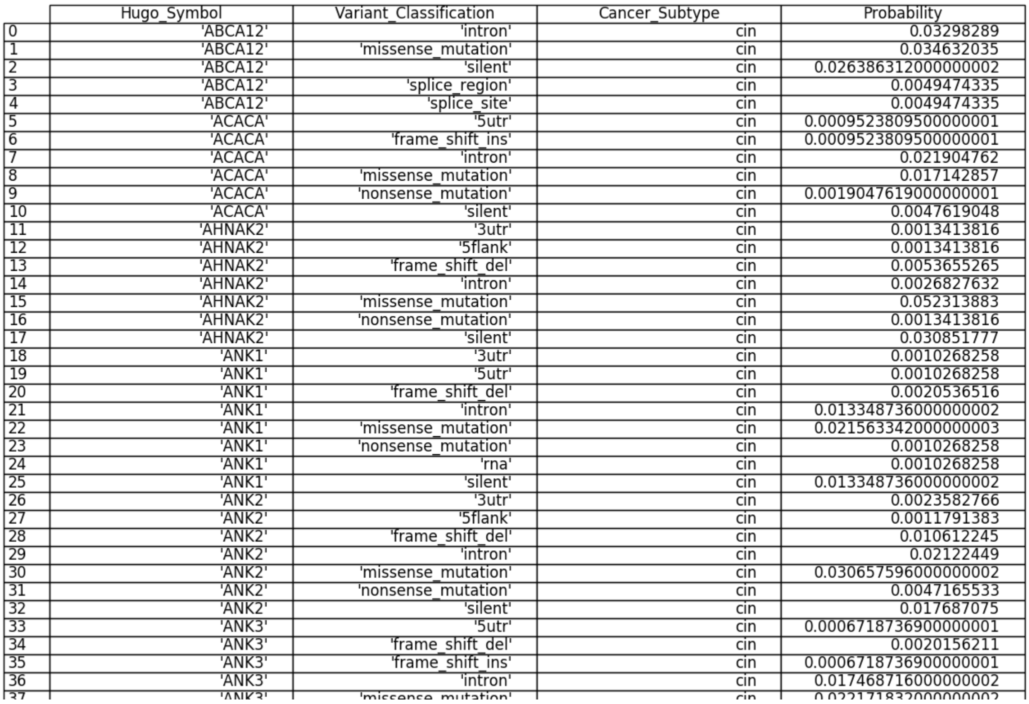
- query(mutgene_vc_cs_prob(GENE,VC,msi)). (1)
- query(mutgene_vc_cs_prob(GENE,VC,cin)). (2)

- query(mutgene_vc_cs_prob(GENE,VC,ebv)). (3)
- query(mutgene_vc_cs_prob(GENE,VC,gs)) (4)



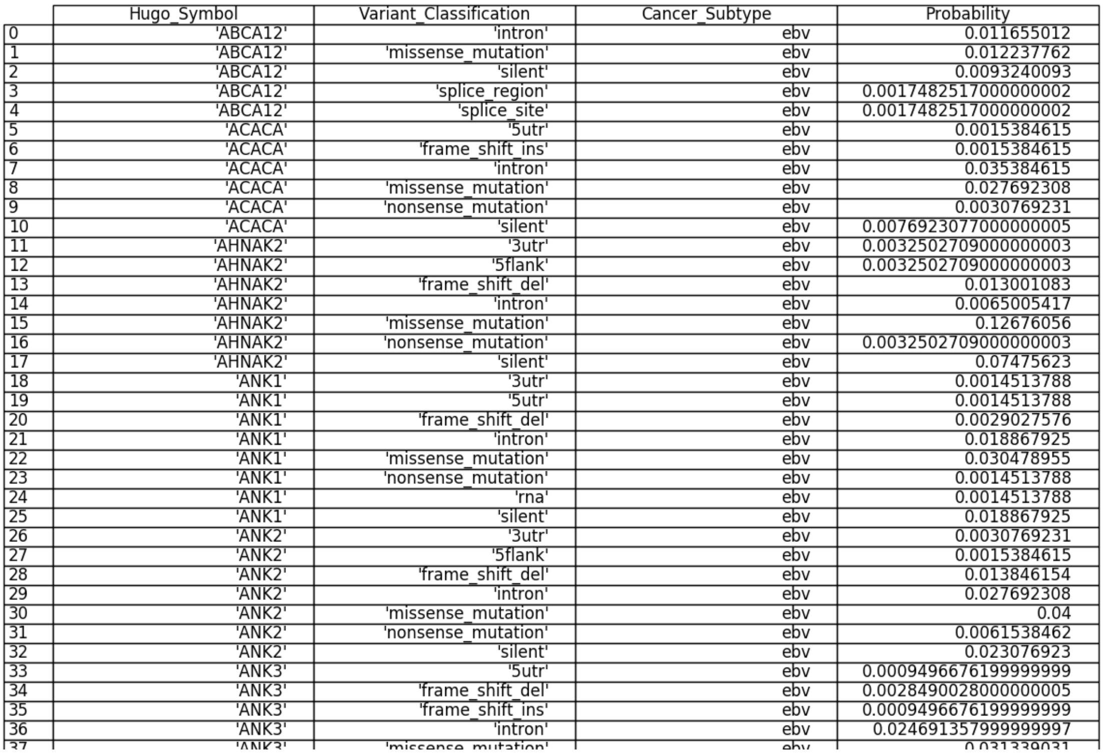
	Hugo_Symbol	Variant_Classification	Cancer_Subtype	Probability
0	'ABCA12'	'intron'	msi	0.16571970000000003
1	'ABCA12'	'missense_mutation'	msi	0.17400568
2	'ABCA12'	'silent'	msi	0.13257576
3	'ABCA12'	'splice_region'	msi	0.024857955
4	'ABCA12'	'splice_site'	msi	0.024857955
5	'ACACA'	'5utr'	msi	0.0103125
6	'ACACA'	'frame_shift_ins'	msi	0.0103125
7	'ACACA'	'intron'	msi	0.2371875
8	'ACACA'	'missense_mutation'	msi	0.185625
9	'ACACA'	'nonsense_mutation'	msi	0.020625
10	'ACACA'	'silent'	msi	0.0515625
11	'AHNAK2'	'3utr'	msi	0.007482394399999999
12	'AHNAK2'	'5flank'	msi	0.007482394399999999
13	'AHNAK2'	'frame_shift_del'	msi	0.029929577000000002
14	'AHNAK2'	'intron'	msi	0.014964789
15	'AHNAK2'	'missense_mutation'	msi	0.29181338
16	'AHNAK2'	'nonsense_mutation'	msi	0.007482394399999999
17	'AHNAK2'	'silent'	msi	0.17209507
18	'ANK1'	'3utr'	msi	0.0097287736
19	'ANK1'	'5utr'	msi	0.0097287736
20	'ANK1'	'frame_shift_del'	msi	0.019457547
21	'ANK1'	'intron'	msi	0.12647406
22	'ANK1'	'missense_mutation'	msi	0.20430425
23	'ANK1'	'nonsense_mutation'	msi	0.0097287736
24	'ANK1'	'rna'	msi	0.0097287736
25	'ANK1'	'silent'	msi	0.12647406
26	'ANK2'	'3utr'	msi	0.01625
27	'ANK2'	'5flank'	msi	0.008125
28	'ANK2'	'frame_shift_del'	msi	0.073125
29	'ANK2'	'intron'	msi	0.14625
30	'ANK2'	'missense_mutation'	msi	0.21125
31	'ANK2'	'nonsense_mutation'	msi	0.0325
32	'ANK2'	'silent'	msi	0.121875
33	'ANK3'	'5utr'	msi	0.0086805556
34	'ANK3'	'frame_shift_del'	msi	0.026041667
35	'ANK3'	'frame_shift_ins'	msi	0.0086805556
36	'ANK3'	'intron'	msi	0.22569444
37	'ANK3'	'missense_mutation'	msi	0.28645833

Figure 19 - Query (1) results from approach III of case study I.
This figure is only a portion of the total query output.



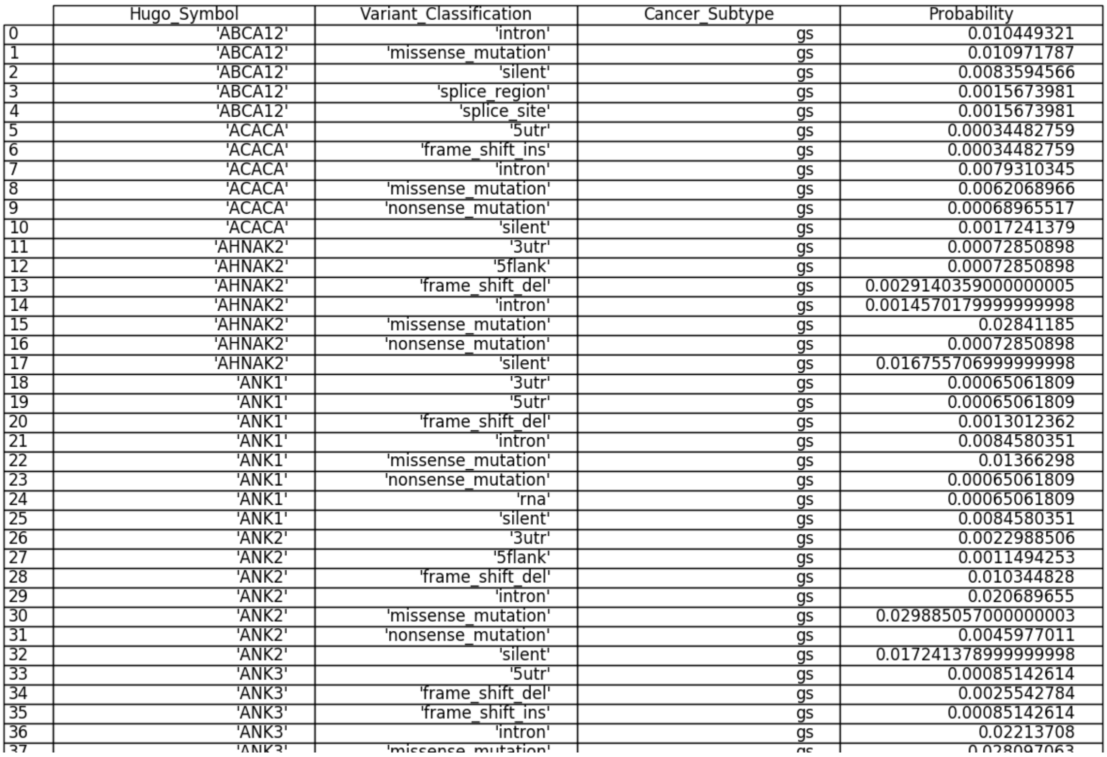
	Hugo_Symbol	Variant_Classification	Cancer_Subtype	Probability
0	'ABCA12'	'intron'	cin	0.03298289
1	'ABCA12'	'missense_mutation'	cin	0.034632035
2	'ABCA12'	'silent'	cin	0.026386312000000002
3	'ABCA12'	'splice_region'	cin	0.0049474335
4	'ABCA12'	'splice_site'	cin	0.0049474335
5	'ACACA'	'5utr'	cin	0.0009523809500000001
6	'ACACA'	'frame_shift_ins'	cin	0.0009523809500000001
7	'ACACA'	'intron'	cin	0.021904762
8	'ACACA'	'missense_mutation'	cin	0.017142857
9	'ACACA'	'nonsense_mutation'	cin	0.0019047619000000001
10	'ACACA'	'silent'	cin	0.0047619048
11	'AHNAK2'	'3utr'	cin	0.0013413816
12	'AHNAK2'	'5flank'	cin	0.0013413816
13	'AHNAK2'	'frame_shift_del'	cin	0.0053655265
14	'AHNAK2'	'intron'	cin	0.0026827632
15	'AHNAK2'	'missense_mutation'	cin	0.052313883
16	'AHNAK2'	'nonsense_mutation'	cin	0.0013413816
17	'AHNAK2'	'silent'	cin	0.030851777
18	'ANK1'	'3utr'	cin	0.0010268258
19	'ANK1'	'5utr'	cin	0.0010268258
20	'ANK1'	'frame_shift_del'	cin	0.0020536516
21	'ANK1'	'intron'	cin	0.013348736000000002
22	'ANK1'	'missense_mutation'	cin	0.021563342000000003
23	'ANK1'	'nonsense_mutation'	cin	0.0010268258
24	'ANK1'	'rna'	cin	0.0010268258
25	'ANK1'	'silent'	cin	0.013348736000000002
26	'ANK2'	'3utr'	cin	0.0023582766
27	'ANK2'	'5flank'	cin	0.0011791383
28	'ANK2'	'frame_shift_del'	cin	0.010612245
29	'ANK2'	'intron'	cin	0.02122449
30	'ANK2'	'missense_mutation'	cin	0.030657596000000002
31	'ANK2'	'nonsense_mutation'	cin	0.0047165533
32	'ANK2'	'silent'	cin	0.017687075
33	'ANK3'	'5utr'	cin	0.0006718736900000001
34	'ANK3'	'frame_shift_del'	cin	0.0020156211
35	'ANK3'	'frame_shift_ins'	cin	0.0006718736900000001
36	'ANK3'	'intron'	cin	0.017468716000000002
37	'ANK3'	'missense_mutation'	cin	0.022171832000000002

Figure 20 - Query (2) results from approach III of case study I.
This figure is only a portion of the total query output.



	Hugo_Symbol	Variant_Classification	Cancer_Subtype	Probability
0	'ABCA12'	'intron'	ebv	0.011655012
1	'ABCA12'	'missense_mutation'	ebv	0.012237762
2	'ABCA12'	'silent'	ebv	0.0093240093
3	'ABCA12'	'splice_region'	ebv	0.0017482517000000002
4	'ABCA12'	'splice_site'	ebv	0.0017482517000000002
5	'ACACA'	'5utr'	ebv	0.0015384615
6	'ACACA'	'frame_shift_ins'	ebv	0.0015384615
7	'ACACA'	'intron'	ebv	0.035384615
8	'ACACA'	'missense_mutation'	ebv	0.027692308
9	'ACACA'	'nonsense_mutation'	ebv	0.0030769231
10	'ACACA'	'silent'	ebv	0.0076923077000000005
11	'AHNAK2'	'3utr'	ebv	0.0032502709000000003
12	'AHNAK2'	'5flank'	ebv	0.0032502709000000003
13	'AHNAK2'	'frame_shift_del'	ebv	0.013001083
14	'AHNAK2'	'intron'	ebv	0.0065005417
15	'AHNAK2'	'missense_mutation'	ebv	0.12676056
16	'AHNAK2'	'nonsense_mutation'	ebv	0.0032502709000000003
17	'AHNAK2'	'silent'	ebv	0.07475623
18	'ANK1'	'3utr'	ebv	0.0014513788
19	'ANK1'	'5utr'	ebv	0.0014513788
20	'ANK1'	'frame_shift_del'	ebv	0.0029027576
21	'ANK1'	'intron'	ebv	0.018867925
22	'ANK1'	'missense_mutation'	ebv	0.030478955
23	'ANK1'	'nonsense_mutation'	ebv	0.0014513788
24	'ANK1'	'rna'	ebv	0.0014513788
25	'ANK1'	'silent'	ebv	0.018867925
26	'ANK2'	'3utr'	ebv	0.0030769231
27	'ANK2'	'5flank'	ebv	0.0015384615
28	'ANK2'	'frame_shift_del'	ebv	0.013846154
29	'ANK2'	'intron'	ebv	0.027692308
30	'ANK2'	'missense_mutation'	ebv	0.04
31	'ANK2'	'nonsense_mutation'	ebv	0.0061538462
32	'ANK2'	'silent'	ebv	0.023076923
33	'ANK3'	'5utr'	ebv	0.0009496676199999999
34	'ANK3'	'frame_shift_del'	ebv	0.0028490028000000005
35	'ANK3'	'frame_shift_ins'	ebv	0.0009496676199999999
36	'ANK3'	'intron'	ebv	0.024691357999999997
37	'ANK3'	'missense_mutation'	ebv	0.031339031

Figure 21 - Query (3) results from approach III of case study I.
This figure is only a portion of the total query output.



	Hugo_Symbol	Variant_Classification	Cancer_Subtype	Probability
0	'ABCA12'	'intron'	gs	0.010449321
1	'ABCA12'	'missense_mutation'	gs	0.010971787
2	'ABCA12'	'silent'	gs	0.0083594566
3	'ABCA12'	'splice_region'	gs	0.0015673981
4	'ABCA12'	'splice_site'	gs	0.0015673981
5	'ACACA'	'5utr'	gs	0.00034482759
6	'ACACA'	'frame_shift_ins'	gs	0.00034482759
7	'ACACA'	'intron'	gs	0.0079310345
8	'ACACA'	'missense_mutation'	gs	0.0062068966
9	'ACACA'	'nonsense_mutation'	gs	0.00068965517
10	'ACACA'	'silent'	gs	0.0017241379
11	'AHNAK2'	'3utr'	gs	0.00072850898
12	'AHNAK2'	'5flank'	gs	0.00072850898
13	'AHNAK2'	'frame_shift_del'	gs	0.0029140359000000005
14	'AHNAK2'	'intron'	gs	0.0014570179999999998
15	'AHNAK2'	'missense_mutation'	gs	0.02841185
16	'AHNAK2'	'nonsense_mutation'	gs	0.00072850898
17	'AHNAK2'	'silent'	gs	0.016755706999999998
18	'ANK1'	'3utr'	gs	0.00065061809
19	'ANK1'	'5utr'	gs	0.00065061809
20	'ANK1'	'frame_shift_del'	gs	0.0013012362
21	'ANK1'	'intron'	gs	0.0084580351
22	'ANK1'	'missense_mutation'	gs	0.01366298
23	'ANK1'	'nonsense_mutation'	gs	0.00065061809
24	'ANK1'	'rna'	gs	0.00065061809
25	'ANK1'	'silent'	gs	0.0084580351
26	'ANK2'	'3utr'	gs	0.0022988506
27	'ANK2'	'5flank'	gs	0.0011494253
28	'ANK2'	'frame_shift_del'	gs	0.010344828
29	'ANK2'	'intron'	gs	0.020689655
30	'ANK2'	'missense_mutation'	gs	0.029885057000000003
31	'ANK2'	'nonsense_mutation'	gs	0.0045977011
32	'ANK2'	'silent'	gs	0.017241378999999998
33	'ANK3'	'5utr'	gs	0.00085142614
34	'ANK3'	'frame_shift_del'	gs	0.0025542784
35	'ANK3'	'frame_shift_ins'	gs	0.00085142614
36	'ANK3'	'intron'	gs	0.02213708
37	'ANK3'	'missense_mutation'	gs	0.028097063

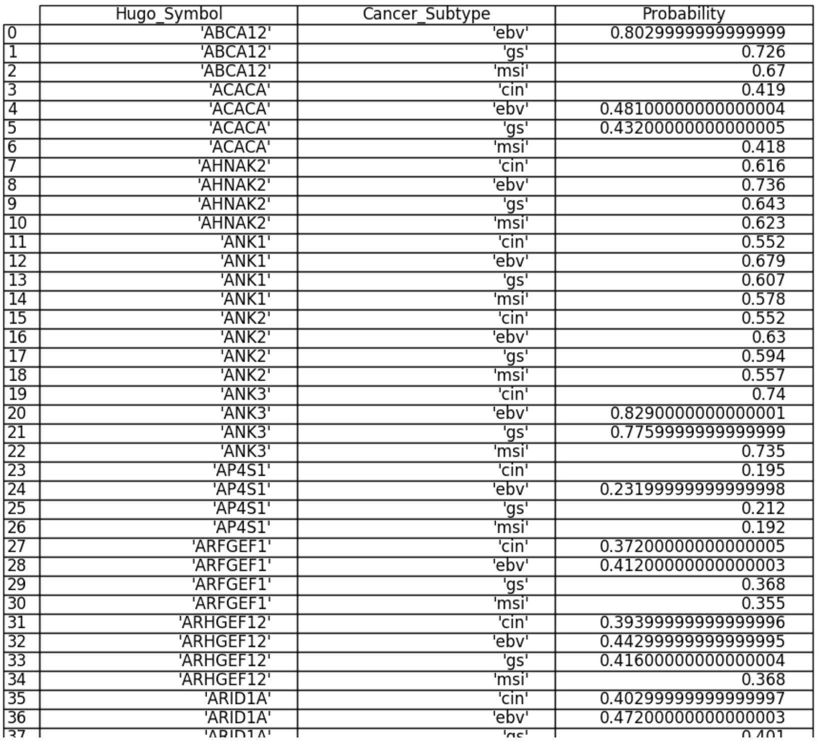
Figure 22 - Query (4) results from approach III of case study I.
This figure is only a portion of the total query output.

Case Study II

Approach I

Query

- query(metgene_cs_prob(GENE,CS)). (1)



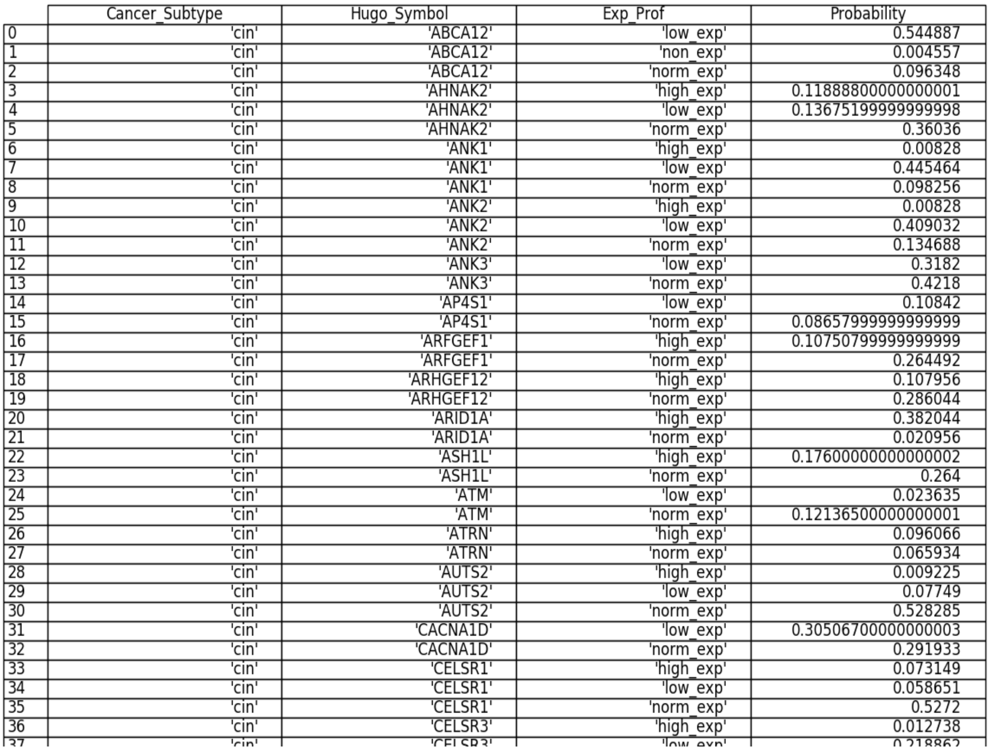
	Hugo_Symbol	Cancer_Subtype	Probability
0	'ABCA12'	'ebv'	0.8029999999999999
1	'ABCA12'	'gs'	0.726
2	'ABCA12'	'msi'	0.67
3	'ACACA'	'cin'	0.419
4	'ACACA'	'ebv'	0.48100000000000004
5	'ACACA'	'gs'	0.43200000000000005
6	'ACACA'	'msi'	0.418
7	'AHNAK2'	'cin'	0.616
8	'AHNAK2'	'ebv'	0.736
9	'AHNAK2'	'gs'	0.643
10	'AHNAK2'	'msi'	0.623
11	'ANK1'	'cin'	0.552
12	'ANK1'	'ebv'	0.679
13	'ANK1'	'gs'	0.607
14	'ANK1'	'msi'	0.578
15	'ANK2'	'cin'	0.552
16	'ANK2'	'ebv'	0.63
17	'ANK2'	'gs'	0.594
18	'ANK2'	'msi'	0.557
19	'ANK3'	'cin'	0.74
20	'ANK3'	'ebv'	0.8290000000000001
21	'ANK3'	'gs'	0.7759999999999999
22	'ANK3'	'msi'	0.735
23	'AP4S1'	'cin'	0.195
24	'AP4S1'	'ebv'	0.23199999999999998
25	'AP4S1'	'gs'	0.212
26	'AP4S1'	'msi'	0.192
27	'ARFGEF1'	'cin'	0.37200000000000005
28	'ARFGEF1'	'ebv'	0.41200000000000003
29	'ARFGEF1'	'gs'	0.368
30	'ARFGEF1'	'msi'	0.355
31	'ARHGEF12'	'cin'	0.39399999999999996
32	'ARHGEF12'	'ebv'	0.44299999999999995
33	'ARHGEF12'	'gs'	0.41600000000000004
34	'ARHGEF12'	'msi'	0.368
35	'ARID1A'	'cin'	0.40299999999999997
36	'ARID1A'	'ebv'	0.47200000000000003
37	'ARID1A'	'gs'	0.401

Figure 23 - Query (1) results from approach I of case study II.
This figure is only a portion of the total query output.

Approach II

Query

- query(cs_metgene_expprof_prob(CS,GENE,EXP)). (1)



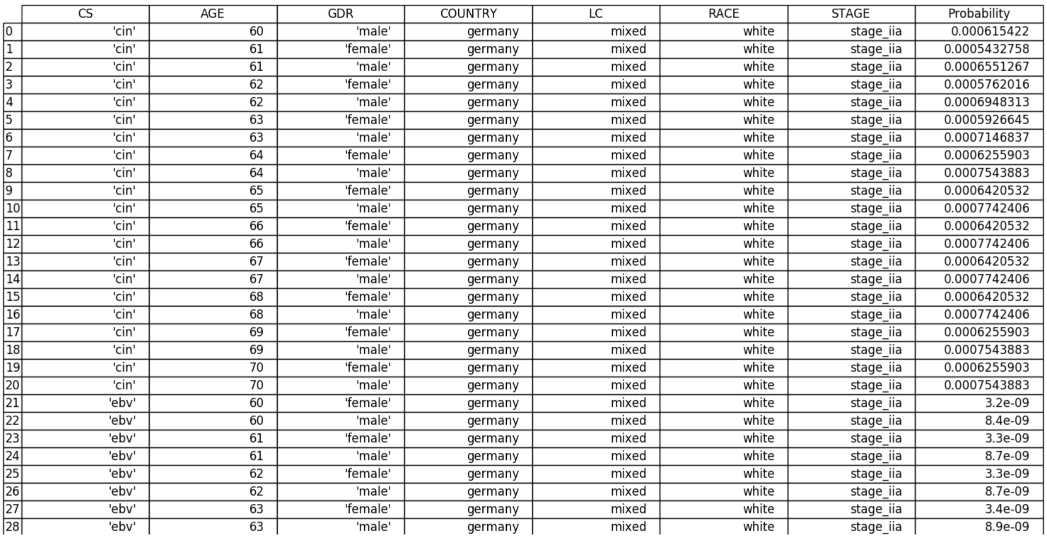
	Cancer_Subtype	Hugo_Symbol	Exp_Prof	Probability
0	'cin'	'ABCA12'	'low_exp'	0.544887
1	'cin'	'ABCA12'	'non_exp'	0.004557
2	'cin'	'ABCA12'	'norm_exp'	0.096348
3	'cin'	'AHNAK2'	'high_exp'	0.11888800000000001
4	'cin'	'AHNAK2'	'low_exp'	0.13675199999999998
5	'cin'	'AHNAK2'	'norm_exp'	0.36036
6	'cin'	'ANK1'	'high_exp'	0.00828
7	'cin'	'ANK1'	'low_exp'	0.445464
8	'cin'	'ANK1'	'norm_exp'	0.098256
9	'cin'	'ANK2'	'high_exp'	0.00828
10	'cin'	'ANK2'	'low_exp'	0.409032
11	'cin'	'ANK2'	'norm_exp'	0.134688
12	'cin'	'ANK3'	'low_exp'	0.3182
13	'cin'	'ANK3'	'norm_exp'	0.4218
14	'cin'	'AP4S1'	'low_exp'	0.10842
15	'cin'	'AP4S1'	'norm_exp'	0.08657999999999999
16	'cin'	'ARFGEF1'	'high_exp'	0.10750799999999999
17	'cin'	'ARFGEF1'	'norm_exp'	0.264492
18	'cin'	'ARHGEF12'	'high_exp'	0.107956
19	'cin'	'ARHGEF12'	'norm_exp'	0.286044
20	'cin'	'ARID1A'	'high_exp'	0.382044
21	'cin'	'ARID1A'	'norm_exp'	0.020956
22	'cin'	'ASH1L'	'high_exp'	0.17600000000000002
23	'cin'	'ASH1L'	'norm_exp'	0.264
24	'cin'	'ATM'	'low_exp'	0.023635
25	'cin'	'ATM'	'norm_exp'	0.12136500000000001
26	'cin'	'ATRN'	'high_exp'	0.096066
27	'cin'	'ATRN'	'norm_exp'	0.065934
28	'cin'	'AUTS2'	'high_exp'	0.009225
29	'cin'	'AUTS2'	'low_exp'	0.07749
30	'cin'	'AUTS2'	'norm_exp'	0.528285
31	'cin'	'CACNA1D'	'low_exp'	0.30506700000000003
32	'cin'	'CACNA1D'	'norm_exp'	0.291933
33	'cin'	'CELSR1'	'high_exp'	0.073149
34	'cin'	'CELSR1'	'low_exp'	0.058651
35	'cin'	'CELSR1'	'norm_exp'	0.5272
36	'cin'	'CELSR3'	'high_exp'	0.012738
37	'cin'	'CELSR3'	'low_exp'	0.218862

Figure 24 - Query (1) results from approach II of case study II.
This figure is only a portion of the total query output.

Case Study IV

Query

- query(match_clinical(CS,AGE,GDR,germany,mixed,white,stage_iia)) :-
  between(60, 70, AGE). (1)



	CS	AGE	GDR	COUNTRY	LC	RACE	STAGE	Probability
0	'cin'	60	'male'	germany	mixed	white	stage_iia	0.000615422
1	'cin'	61	'female'	germany	mixed	white	stage_iia	0.0005432758
2	'cin'	61	'male'	germany	mixed	white	stage_iia	0.0006551267
3	'cin'	62	'female'	germany	mixed	white	stage_iia	0.0005762016
4	'cin'	62	'male'	germany	mixed	white	stage_iia	0.0006948313
5	'cin'	63	'female'	germany	mixed	white	stage_iia	0.0005926645
6	'cin'	63	'male'	germany	mixed	white	stage_iia	0.0007146837
7	'cin'	64	'female'	germany	mixed	white	stage_iia	0.0006255903
8	'cin'	64	'male'	germany	mixed	white	stage_iia	0.0007543883
9	'cin'	65	'female'	germany	mixed	white	stage_iia	0.0006420532
10	'cin'	65	'male'	germany	mixed	white	stage_iia	0.0007742406
11	'cin'	66	'female'	germany	mixed	white	stage_iia	0.0006420532
12	'cin'	66	'male'	germany	mixed	white	stage_iia	0.0007742406
13	'cin'	67	'female'	germany	mixed	white	stage_iia	0.0006420532
14	'cin'	67	'male'	germany	mixed	white	stage_iia	0.0007742406
15	'cin'	68	'female'	germany	mixed	white	stage_iia	0.0006420532
16	'cin'	68	'male'	germany	mixed	white	stage_iia	0.0007742406
17	'cin'	69	'female'	germany	mixed	white	stage_iia	0.0006255903
18	'cin'	69	'male'	germany	mixed	white	stage_iia	0.0007543883
19	'cin'	70	'female'	germany	mixed	white	stage_iia	0.0006255903
20	'cin'	70	'male'	germany	mixed	white	stage_iia	0.0007543883
21	'ebv'	60	'female'	germany	mixed	white	stage_iia	3.2e-09
22	'ebv'	60	'male'	germany	mixed	white	stage_iia	8.4e-09
23	'ebv'	61	'female'	germany	mixed	white	stage_iia	3.3e-09
24	'ebv'	61	'male'	germany	mixed	white	stage_iia	8.7e-09
25	'ebv'	62	'female'	germany	mixed	white	stage_iia	3.3e-09
26	'ebv'	62	'male'	germany	mixed	white	stage_iia	8.7e-09
27	'ebv'	63	'female'	germany	mixed	white	stage_iia	3.4e-09
28	'ebv'	63	'male'	germany	mixed	white	stage_iia	8.9e-09

Figure 25 - Query (1) results from of case study IV.
This figure is only a portion of the total query output.