ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
UNIVERSITY OF CRETE

FORTH
INSTITUTE OF COMPUTER SCIENCE

University of Crete
Foundation of Research and Technology Hellas
Institute of Computer Science

A graph representation of the individual exome variation with evidence from biomedical text corpora

Ioannis Giannoulakis

June, 2022

Three member committee:

Dr. Ioannis Iliopoulos

Dr. George Potamias

Dr. Alexandros Kanterakis

Thesis submitted within the Master's Program Bioinformatics

A graph representation of the individual exome variation with evidence from biomedical text corpora

Abstract

One of the most crucial steps in clinical genetics pipelines is variant annotation and prioritization. This step usually includes the consultancy of other databases that can shed light on the importance of the identified genomic variation. One of the genomic data sources with a valuable wealth of information is online BioMedical publication databases such as PubMed. Today is debatable as to which extend modern clinical genetics pipelines involved in Next Generation Sequencing exploit this information.
Despite the plethora of available methods for information extraction from biomedical text, they rarely take part in the annotation/prioritization step of typical Next Generation Sequencing pipelines. This is because existing methods are not suited for mass querying the complete genome variation of an individual. Here we present an open tool that builds a graph from the BioC corpus consisting of all open and extensively pre-annotated PubMed articles in less than 10 hours. In this graph, nodes represent Articles (n=27M), Chemicals (n=350K), Diseases (n=12K), Genes (n=37K), Mutations (n=1.1M) interconnected through 190 million edges.The graph can be queried and explored through the Cypher language that is served and visualized through the Neo4j graph database engine. Through this engine we can query the entirety of variants (~50K) identified in NGS experiments in a practical timescale. The result of this query is the intersection of the graph's mutations with those of the file that have been given as input. The articles that contain these mutations are used for topic modeling through Top2Vec.Through the results of topic modeling, a user can easily and flexibly investigate all existing bibliographic evidence linking the genetic profile of the individual with known diseases and chemical/drug interactions.

Acknowledgments

I would like to thank my thesis advisor Dr. Alexandros Kanterakis who guided me from the beginning to the end of this thesis. His inspirational ideas in science, his trusting in me and his willingness to help me, played a key role in completing my master thesis. I would also like to thank my supervisor, Dr. Ioannis Iliopoulos for the valuable advice and ideas when I needed them.
I would also like to thank all of my fellow students for the interesting, educational and supportive conversations.
Finally, I must express my very profound gratitude to my parents, my brother and my sister for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Contents

List of Figures

List of Tables

1.Introduction

1.1 Exome sequencing

In recent years, genome sequencing is a term that we come across quite often and more and more people are moving towards this test. The benefits of this experiment are many as it is possible to detect mutations that cause diseases, genes that cause a disease or a phenotype. Exome sequencing now tends to be a routine test. Since the beginning of the millennium, the cost of genome sequencing has dropped dramatically. Specifically in 2001 the cost per genome sequencing was $ 100,000,000, until 2007 this price was reduced to $ 10,000,000 while today it costs less than $ 1,000[1] for a quality level acceptable in clinical genetics. (Figure 1.1)

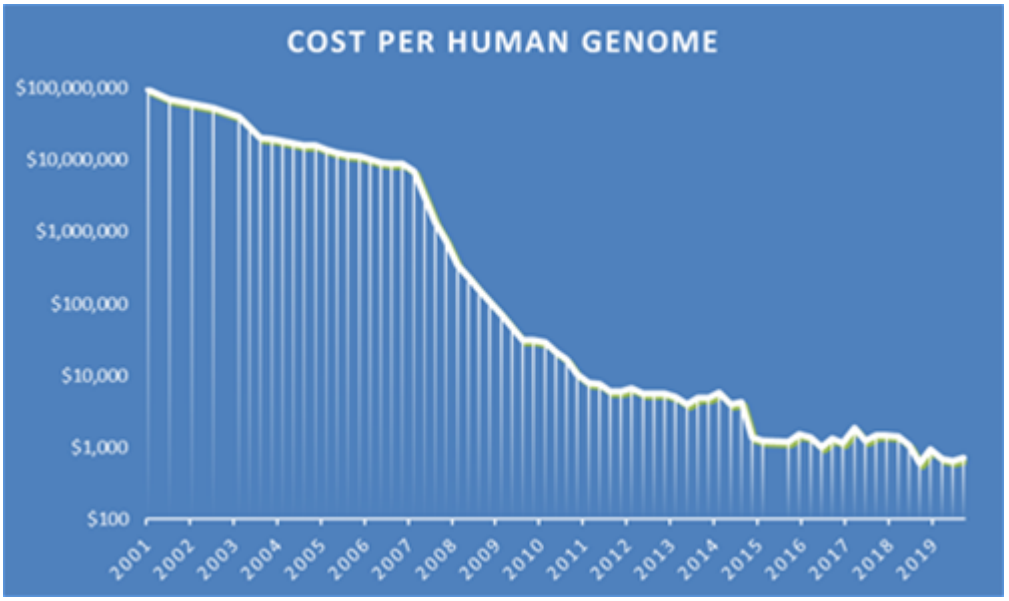

Figure 1.1 : "Evolution of the cost of sequencing a human genome[2]"

1.1.1 Whole exome sequencing vs whole genome sequencing

Whole exome sequencing (WES) is divided into two steps. The first is the isolation of DNA regions that encode proteins. The second step is to sequence these areas using any high-throughput DNA sequencing technology. The above procedure aims to detect genetic mutations that alter protein sequences at a much lower cost than whole genome sequencing. Although exons occupy only 1.1% of the total genome (Venter et al. 2001) or about 30 megabases of DNA, it contains approximately 85% of known disease-related variants (Choi et al. 2009). On the other hand, whole genome sequencing is more powerful and more sensitive than whole-exome sequencing in detecting potentially disease-causing mutations within the exome (Belkadi et al. 2015). One must also keep in mind that non-coding regions

[1] https://en.wikipedia.org/wiki/$1,000_genome
[2] https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data

can be involved in the regulation of the exons that make up the exome, and so whole-exome sequencing may not be complete in showing all the sequences at play in forming the exome. Nevertheless, since the exome is a very small part of the total genome, this process is more cost efficient and fast as it involves sequencing around 40 million bases rather than the 3 billion base pairs that make up the genome (Nagele 2013). For these reasons WES is preferred for many research activities that require the study of the coding regions of the genome.

1.1.2 Next Generation Sequencing

Next Generation Sequencing (NGS) is a relatively new technology used for DNA and RNA sequencing and mutation detection, starting a few years after the completion of the Human Genome Project (2001). NGS can sequence thousands of genes or an entire genome in a short period of time. Mutations that have been detected by NGS are widely used for disease diagnosis, prognosis, treatment decision, and patient monitoring. The development and improvement of the next generation sequencing has led to the increasing application of cancer genomic research over the last decade.

1.1.3 Next generation sequencing diagnostic pipeline

A common diagnostic pipeline that uses NGS methods has the following scheme:

1) Sample collection
2) Sequencing and generation of FASTQ raw reads
3) Alignment to a reference genome
4) Variant calling
5) Variant annotation
6) Variant prioritization
7) Reporting

The steps from “FASTQ raw reads generation” to “variant annotation” have been fairly automated through well tested and streamlined NGS pipelines. Some examples are SIMPLEX (Fischer et al. 2012), combination of MuTect and GATK (Valle et al. 2016), OpEx (Ruark et al. 2016) and WEP (D’Antonio et al. 2013). Moreover these pipelines have been imported in Workflow Management Systems making the installation, deployment and comparison a relatively easy process even for researchers with limited IT knowledge. Some examples are SeqMule (D’Antonio et al. 2013; Guo et al. 2015) that uses the Galaxy system[3],Sarek (Garcia et al. 2020) that uses Nextflow (Di Tommaso et al. 2017) and NGS-pipe (Singer et al. 2018) that uses Snakemake (Singer et al. 2018; Köster and Rahmann 2018). Additionally, testing and benchmarking these approaches is possible via the introduction of the Genome In A Bottle dataset (Zook et al. 2014). This dataset consists of a Trio (mother, father, child) of Askenazi descent, a trio of Chinese descent and a male sample (NA12878) of European descent. For a complete list of samples, sequencing techniques and applied pipelines see[4]. All these samples have been analyzed in a plethora of sequencing and genotyping platforms and processed in almost all common alignment and variant calling software packages (Cornish and Guda 2015), (Zook et al., n.d.), (Linderman et al. 2014).

[3] https://galaxyproject.github.io/training-material/topics/variant-analysis/tutorials/exome-seq/tutorial.html

[4] https://github.com/genome-in-a-bottle/giab_data_indexes

1.1.4 Variant pathogenicity

Despite the plethora, availability and automation of exome sequencing pipelines, the task of variant detection with potential clinical interest is still a very challenging task. Towards this direction many frameworks have been introduced that employ in-silico methods for the prediction of the deleterious status of the identified variants. Some examples are VAST (Flygare et al. 2018) and Cpipe (Stark et al. 2017). These frameworks usually use three types of information in order to classify a variant: rarity (allele frequency), conservation status and protein effect. Moreover they do so by employing known metrics of pathogenicity such as the CADD (Rentzsch et al. 2019), SIFT (Sim et al. 2012) and Polyphen-2 scores (Adzhubei, Jordan, and Sunyaev 2013). Over the last years more complex types of information have been used for this task. For example DeepPVP ((Adzhubei, Jordan, and Sunyaev 2013; Boudellioua et al. 2019) incorporates information from the OMIM (Hamosh et al. 2000) and the ClinVar database (Landrum and Kattman 2018). Other tools like Phevor (Singleton et al. 2014), Exomiser (Singleton et al. 2014; Smedley et al. 2015) and Phenolyzer (Yang, Robinson, and Wang 2015) use structured information from known ontologies like the Gene Ontology (GO) ("Gene Ontology (GO)," n.d.) , the Human Phenotype Ontology (HPO) (Köhler et al. 2017) and the Disease Ontology (DO) (Schriml et al. 2012).

1.1.5 Sequence Variant Nomenclature

As shown in figure 1.2 the number of publications, or else articles that contain genomic variants in PubMed is constantly increasing. Every year from 2016 onwards more than 50,000 articles contain genomic variants. These articles deal with the discovery of new mutations or the investigating of existing ones. Already known mutations exist in various databases such as e.g. dbSNP (Sherry, Ward, and Sirotkin 1999), GWAS (Sherry, Ward, and Sirotkin 1999; Pearson 2008), HapMap (Consortium and †The International HapMap Consortium 2003), JSNP (Hirakawa et al. 2002). An issue that was more acute in previous years was that some databases used different nomenclatures to report a mutation. Table 1.1 shows the different ways of reporting a mutation (Poo, Cai, and Mah 2011).

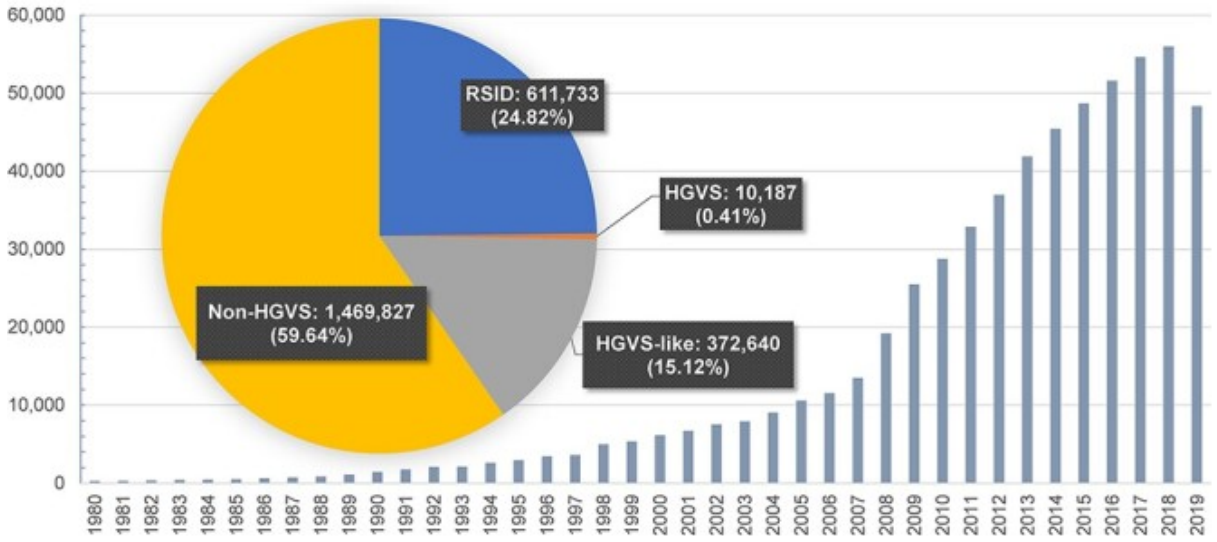

Figure 1.2: "Number of publications that contain genomic variants in PubMed and PMC". From 2016 onwards, more than 50,000 articles per year contain mutations (Lee, Wei, and Lu 2021)

Database	SNP names
dbSNP	rs3737965 ss4923964, ss69366921
HGVBaseG2P	HGVM2256489
HGVS	NM_001286.2:c.87+45G>A, NM_021735.2:c.87+45G>A NM_021736.2:c.87+45G>A, NM_021737.2:c.87+45G>A NT_021937.19:g.7871183G>A
JSNP	IMS-JST083663
PharmGKB	rs3737965@chrl: 11789038
HapMap	rs3737965

Table 1.1: "Alternative names of a SNP"

Given the plethora of different naming schemes, it was imperative to use a common nomenclature for mutation reporting that would allow communication between different databases. Although the most common mutation reporting method was rs-id from dbSNP, in recent years there has been a trend towards the use of the HGVS format (Human Genome Variation Society) (Antonarakis and Nomenclature Working Group 1998). There are specific and strict guidelines for the HGVS form of a mutation. The official HGVS guidelines were introduced in 2001 and the last update was published in 2016 (den Dunnen et al. 2016).
The complete form of a HGVS variant is "reference:description".
According to the oficial HGVS guidelines the part of "reference" must be one of the following[5]:

- RefSeq sequences with the prefixes NC_, NT_, NW_,NG_, NM_, NR_ or NP_
    - chromosome - NC_000023.11
    - genomic contigs or scaffolds - NT_010718.17, NW_003315950.2
    - gene/genomic region - NG_012232.1
    - coding transcript - NM_004006.2
    - non-coding transcript - NR_004430.2
    - protein - NP_003997.1
- Ensembl transcript (ENST) and protein (ENSP) which are not identified by Ensembl as being incomplete, e.g. CDS 5' incomplete (cds_start_NF), CDS 3' incomplete (cds_end_NF)
    - gene/genomic region - ENSG00000198947.15

[5] http://varnomen.hgvs.org/bg-material/refseq/

    - coding transcript - ENST00000357033.8
    - non-coding transcript - ENST00000383925.1
    - protein - ENSP00000354923.3
- LRG sequences with the prefixes LRG_#, LRG_#t#, LRG_#p# (see examples below)
    - gene/genomic region - LRG_199
    - coding transcript (or non-coding transcript) - LRG_199t1
    - protein - LRG_199p1

The part of “description” must be one of the following:

- DNA
    - g. = linear genomic reference sequence
    - o. = circular genomic reference sequence
    - m. = mitochondrial reference (special case of a circular genomic reference sequence)
    - c. = coding DNA reference sequence (based on a protein coding transcript)
    - n. = non-coding DNA reference sequence (based on a transcript not coding for a protein)
- RNA
    - r. = RNA reference sequence
- Protein
    - p. = protein reference sequence

Thus, some examples of complete HGVS forms of mutations are the following:

- DNA - coding variant: NM_004006.1:c.5690G>A
- Protein variant: NP_003997.1:p.(Trp24Cys)

More details for HGVS nomenclature recommendations are located in https://varnomen.hgvs.org/bg-material/simple/

1.1.6 HGVS variant validation

There are some tools that can check if a variant is valid and if it has the correct nomenclature. Some of them are Mutalyzer (Wildeman et al. 2008; Lefter et al., n.d.) , VariantValidator (Freeman et al. 2018) and the hgvs Python package (M. Wang et al. 2018) . Although Mutalyzer is a widely used tool for validating sequence variant descriptions, it is not able to validate intronic variants, for example NM_206933.2:c.6317C>G.
Hgvs Python package has an important and useful function which allows the conversion between coding position to protein position for a variant as well as the conversion from a genomic position to coding position.
VariantValidator validates coding and genomic HGVS sequence variation descriptions, accurately mapping between transcript and genomic variants. It also utilizes the hgvs Python package.

1.2 Biomedical text mining

PubMed is a database and a search engine for biomedical publications. Pubmed provides access to more than 33 million biomedical articles. Thousands of new articles are added to Pubmed daily. This exponential increase in information volume at Pubmed can be addressed with the help of text mining. This fact led to the integration of text mining in biomedicine. The field of biomedical text mining is becoming an integral part of biomedical workflows.
There are various categories of application of text mining in biomedicine. Some of the most known are the following:

- Document clustering and classification
- Information extraction (IE)
- Information retrieval (IR)
- Name entity recognition (NER)
- Natural language processing (NLP)
- Question-answering (QA)
- Visualization

1.2.1 Genotype-Phenotype relation extraction

Both the research and the clinical community are interested in identification of genotype-phenotype relationships. Some known databases such as OMIM (Hamosh et al. 2000), HGMD (Griffith and Griffith 2004), Comparative Toxicogenomics Database (CTD) (Mattingly et al. 2003), employ manual curation of biomedical literature to provide data that can help at the detection of genotype-phenotype relationships.

However, the huge volume of biomedical information published daily makes it difficult to update the above databases manually. To this end, extraction tools for biomedical entities have been developed in the last 15 years.

Concerning the mutations there are a lot of tools which can detect a mutation in raw text. Some examples are TmVar (Wei et al. 2013, 2018), SETH (Thomas et al. 2016) , AVADA (Birgmeier et al., n.d.), EMU (Extractor of Mutation) (Doughty et al. 2011), MutationFinder (Caporaso et al. 2007).

Concerning disease terms, there are tools like TaggerOne (Leaman and Lu 2016), DNorm (Leaman and Lu 2016, 2014), PhenoTagger (Luo et al. 2021), Metamaplite (Luo et al. 2021; Demner-Fushman, Rogers, and Aronson 2017). All of the above tools can detect disease terms in a text and normalize those terms into different ontologies.

Despite having so many entity extraction tools, to our knowledge, there are no tools that focus on extracting mutation-disease relationships.

1.2.2 Biomedical corpus - PubMed

Pubmed, trying to make the information it contains accessible to anyone, has made the PMC (Pubmed Central) corpus which contains 3 million full text articles. The number of full text

articles offered by pubmed is constantly increasing. For the rest of the articles, pubmed gives access to 30 million abstracts.
All of these abstracts contain a lot of information which is difficult to manually manage.

Thus, some corpus have been created such as the BioC-PMC which contains all the PMC articles in BioC format. Bioc is an XML-based format for embedding text, annotations and relations. The aim of this corpus is to provide machine readable, easy and flexible access to all PMC texts.
All these articles are provided via the FTP site of NCBI in compressed files with a total size of 50 gb.

Another web service from NCBI is the PubTator Central which enables the retrieval of biomedical annotations in biomedical articles. PubTator Central's articles contain annotations from various tools for identifying genes, mutations, diseases, and chemicals. These tools are TmVar for variants, GNormPlus for genes, TaggerOne for disease and chemicals terms. PubTator Central annotates 30 million abstracts of PubMed and the 3 million full text articles of PMC Open Access Subset and enables users to download all these articles along with bio-entity annotations through the FTP site. All these files of PubTator Central's FTP site occupy about 0.5 terabyte and are updated monthly. These files are available in three formats:

1) PubTator
2) BioC-XML
3) BioC-JSON

Figure 1.3 represents the pipeline of making the PubTator Central's corpus.

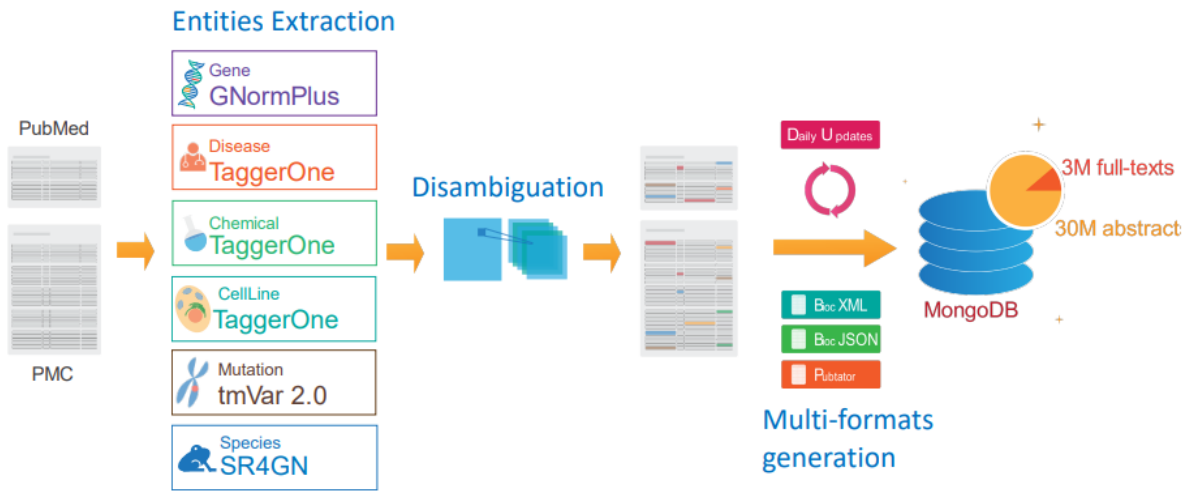

Figure 1.3: "Pipeline of making the PubTator Central's corpus" PubMed abstracts and Full text articles from PubMed Central are annotated from entities extraction tools. The disambiguation module resolves annotation conflicts and finally the annotated articles are stored in a database in multiple formats.

1.3 Graph Databases

Everyone interacts with databases on a daily basis without even realizing it. The history of databases dates back to 1960 when Charles Bachman designed the first one. Since then, there has been significant progress made in this field.
There are a lot of database types. Some of these are relational databases, NoSQL, Cloud, Hierarchical, Document, Graph and Time-series.
Graph databases are used when there is a need to create relationships between data that need to be quickly queried. A graph database is a NoSQL database that stores data as a network graph. What differentiates graph databases from relational database engines (i.e. MySQL, Postgresql) is that the main data point is the node and its relations as opposed to the “table” and its rows..
In graph theory nodes are also referred to as vertices and relationships are also referred to as edges or links. In graph databases nodes are also referred to as entities. Both nodes and edges can be annotated with meta-information. Nodes can have labels. Labels characterize a node with a simple attribute. For example, a node can have the label “Person”. Both nodes and edges have “Properties” which are key value pairs. For example a node can have the label “Person” and an attribute: “Name=John”. Although labels can be only strings, attribute values can have multiple types, even complex types such as lists[6].
Overall graph databases offer a very verbose schema for storing complex semantic information. Also, they are designed to be scalable and offer flexibility that's hard to find in other databases.
One of the main reasons developers are choosing graph databases is performance. For certain types of big data problems–particularly those that involve analyzing the relationships among millions or billions of entities–a graph database will outperform nearly every other type of out-of-the-box database in existence.
Graph databases are commonly selected against other types of databases for performance[7].
In particular, when it comes to big data problems that involve the analysis of complex relationships between entities (Vicknair et al. 2010) (Sahatqija et al. 2018).

1.4 Topic modeling

Topic modeling is an unsupervised machine learning technique which is used for the characterizing of a set of documents. This is achieved by detecting words and phrases within the texts and by automatically clustering word groups and similar expressions. The goal of topic modeling is to discover topics in a collection of articles.
There are two categories of topic models, Statistical and Deep learning models. Some examples of Statistical models are LSA (Dumais 2005), pLSA (Hofmann 2017), LDA (Blei et al. 2003) . Each of them uses a different way of finding topics.

1.4.1 LDA

To date, the most common way of topic modeling is the LDA (Latent Dirichlet Allocation) method. LDA is a probabilistic model which is applied over the words of a set of documents.

---

[6] https://neo4j.com/docs/getting-started/current/graphdb-concepts/
[7] https://www.tigergraph.com/blog/what-are-the-major-advantages-of-using-a-graph-database/

Its outputs are a list of topics from a collection of documents and a probability distribution over the topics that were identified for each document.
The LDA algorithm requires some pre-processing steps. There are:

- Text conversation into lowercase
- Split text into words
- Remove the stop loss words
- Remove symbols and special characters
- Lemmatization

Except for the many steps required for the implementation of the LDA algorithm, there are some parameters that need to be configured. These are[8]:

- Number of Topics
- Number of Iterations
- Chunksize. The number of documents to load into memory at once
- Alpha: the document-topic density. The higher the alpha the more topics will be found within the document.
- Beta: the topic word density. the higher the beta, the topics consist of a large number of words in the corpus

Configuring these parameters and steps is not a trivial process and requires excessive experimentation. This is one of the reasons for the recent advent of algorithms that perform topic analysis in a more data-agnostic manner, that do not require configuring an excessive amount of parameters. One of these is Top2Vec.

1.4.2 Top2Vec

A new, up-and-coming and promising algorithm is Top2Vec (Angelov 2020) which is used for topic modeling and semantic search.
As referred to https://github.com/ddangelov/Top2Vec some of the benefits of Top2Vec over other algorithms such as LDA are the following:

- Automatically finds the number of topics.
- No stop word lists required.
- No need for stemming/lemmatization.
- Creates jointly embedded topic, document, and word vectors.
- Has search functions built in.

Top2vec enables the user for multiple results such as:

- Get hierarchical topics from a set of documents.
- Search topics by keywords.
- Search documents by topic or keywords.
- Find similar documents.

The three main steps of Top2Vec are:

1) Transform documents to numeric representations through Doc2Vec (Quoc et al. 2014) or Universal Sentence Encoder (Cer et al. 2018) or BERT Sentence Transformer (Devlin et al. 2018).

[8] https://radimrehurek.com/gensim/models/ldamodel.html

2) Dimensionality reduction using UMAP (McInnes et al. 2018) algorithm.
3) Clustering of documents to find topics using HDBSCAN (McInnes, Healy, and Astels 2017) algorithm.

1.5 Bioinformatics workflows

In recent years, with the discovery of many biological and medical data, and the advancements in sequencing technology, a plethora of bioinformatics tools have being developed. It is very common for these tools to require a complex amount of steps in order to configure, install and use them. Additionally chaining multiple tools into pipelines often requires both tools to share input/output formats and to also be able to coexist on the same computation environment. Both these prerequisites are not commonly met, which complicates this process and renders a seemingly simple pipeline, a programming task that requires above average IT skills. For this reason, bioinformatics workflow management systems have been developed in order to simplify tool chaining, offer execution in a variety of computation environments and streamline the complete analysis. Some of them are Galaxy, Taverna[9], Nextflow (Di Tommaso et al. 2017) and OpenBio (Kanterakis et al. 2021).

Directed Acyclic Graphs (DAG)

A common way of representing workflows is Directed Acyclic Graphs (DAG). In a DAG nodes represent the processing steps and edges represent step dependence. Figure 1.4 represents an example of a DAG (Jain and Kumari 2017).

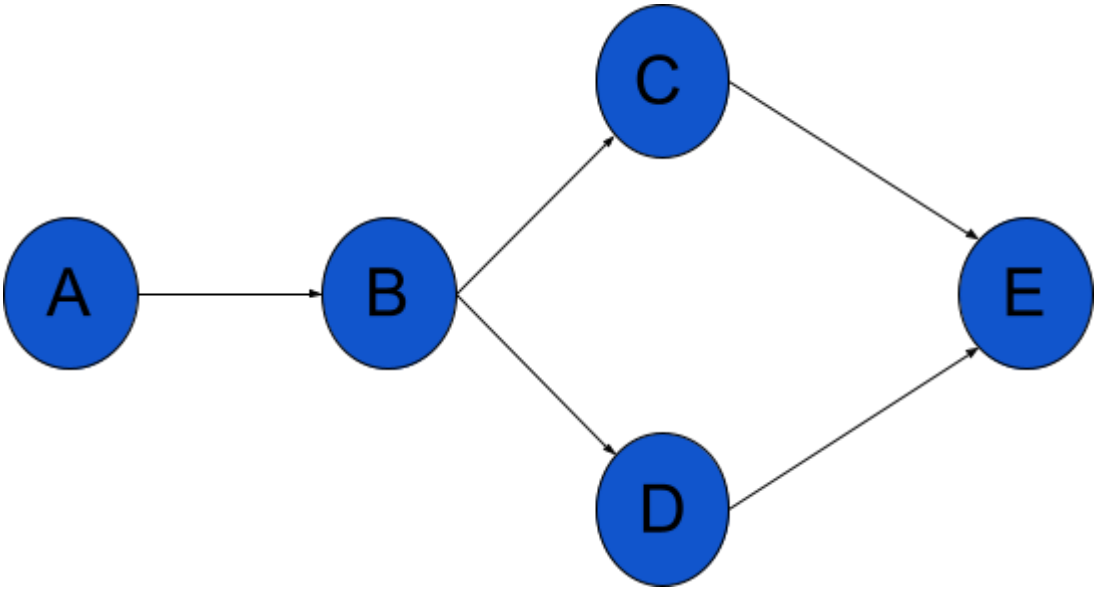

Figure 1.4: "Example of a Directed Acyclic Graph". Step A runs first. Step B depends on step A. Both steps C and D depend on B. Step E depends on C and D.

Reproducibility crisis

With the exponential increase of publications in recent years, the new tools and technologies provided by researchers with the aim of their free use by the community have also increased.

[9] http://www.taverna.org.uk/download/workbench/2-5/bioinformatics/

Nevertheless, a major problem that has arisen is the reproducibility crisis (Baker 2016). This term expresses the difficulty of reproducing the analysis and the discrepancies in validating the results from published works. Non-reproductive science has no practical usage for the community. However, there are several papers that suggest solutions to address this crisis by setting rules for publishing new tools and technologies.(Sandve et al. 2013; Ligozat et al. 2020); (Kulkarni et al. 2018)

1.6 Research Purpose

This work lies in the intersection between exome sequencing and biomedical text mining. Our goal is to integrate biomedical text mining into an exome sequencing pipeline. The purpose of this thesis is to enable anyone with a set of mutations to easily and quickly search for all the information available on PubMed about this set. Essentially, the aim of this project is to flexibly find the projection of a set of mutations in the existing literature. In this work, an attempt is made to extract information from database differences and combine it with the information extracted from Pubmed.
Overall we expect that this thesis will result in a framework that will help clinical genetics take more informed decisions when reporting variants for diagnostic purposes.

2. Methods

2.1 A pipeline for manual identification of biomedical entities

2.1.1 Pipeline

In order to implement the central idea of this thesis we made a pipeline (figure 1.5). The first step of this pipeline is to collect data from PubMed. The second one is the parsing of these articles of PubMed with some tools which can detect mutations, genes and diseases. The third one is the organization of the entities which have been retrieved from the second step into a database making the navigating to this information easier. The fourth step of this pipeline is the massive querying for a set of mutations and the last step is report generation.

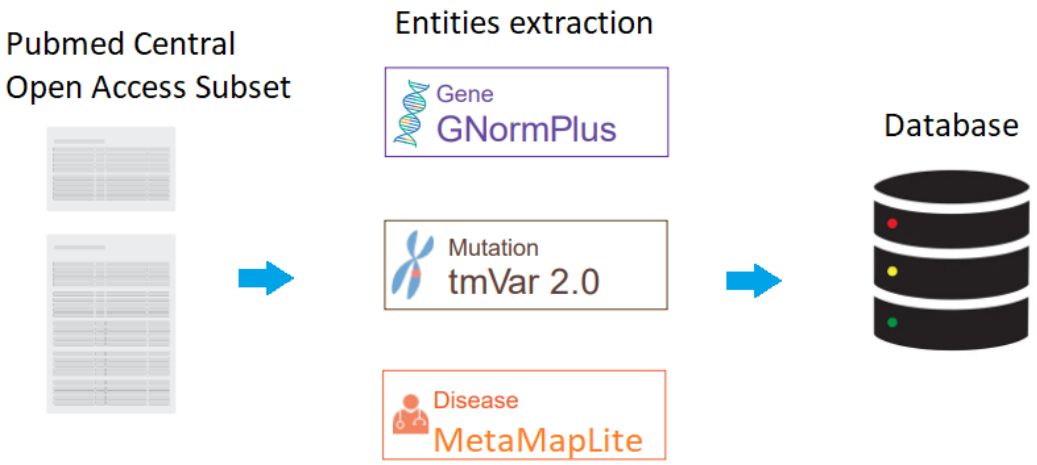

Figure 1.5: "Pipeline for manual identification of biomedical entities". PMC full text articles are annotated by multiple concept taggers and stored in a database.

2.1.2 Corpus

In our first attempt to implement the above pipeline we downloaded the BioC-PMC dataset which is described in section 1.2.2. The articles in the BioC-PMC corpus are available in both the original Unicode characters as well as an ASCII encoding. BioC files are available in both XML and JSON. This dataset consists of about 55GB of compressed files.

2.1.3 Tools

In order to parse the above data we installed a set of text mining tools. Our purpose was to detect variants, genes and diseases in every article. We tried to install and use all these tools that are mentioned in section 1.2.1 to decide which one we would use. One prominent exception was AVADA (Birgmeier et al., n.d.) which had dependencies in libraries that could not be found. We skipped this tool after our requests for help from the corresponding authors were left unanswered.
We finally chose TmVar 2.0 which had the highest efficiency in extracting genomic variant information from biomedical literature according to a recent review (Lee, Wei, and Lu 2021). As described in section "1.6 Research Purpose" our aim was to detect variants in any form and normalize them to HGVS format and TmVar fits perfectly in this task.

TmVar recognizes not only variants but also gene names that appear near the variants in the text. It links the gene names to variants so that the tool can find gene-variant pairs to specify the correct variant information. This feature of TmVar is achieved in combination with GNormPlus (Wei, Kao, and Lu 2015). GNormPlus is a tool that detects genes and proteins in raw text. In order to work properly with these two tools, GNormPlus needs to run first for a text and then the output of this process should be the input for TmVar.
Concerning disease terms, among the tools that are referred to in section 1.2.1, we chose MetaMapLite.

Continuing on our pipeline, after we downloaded the papers and installed the tools, we started the process of parsing papers in order to detect variants and genes. Trying to parse 3 Million articles 3 times (genes-mutations-diseases), we realized that the required time was about 150 days for each tool! Even when we tried to run GNormPlus or TmVar in parallel the estimated time was prohibitive.

2.2 A pipeline for retrieving biomedical entities from pre-annotated corpus and importing them to a database

2.2.1 Corpus

The prohibitive estimated time mentioned above led us to PubTator Central, which is described in section 1.2.2. This corpus fits perfectly with our goal as it is made in a similar way to the one described in the section of Pipeline.

Regarding the FTP site of PubTator Central, we encountered some issues. Specifically, although there were 10 files with the suffix ".gz", they were not files that were zipped through gunzip but were tar files. Additionally, a file of them was not downloadable. We contacted PubTator's communications manager in order to address the above issues and they were resolved.

2.2.2 Parsing

After we downloaded all these compressed files (10 tar files), we realized that each tar file contained a list of bioc.xml files. Each one of them was a collection of articles in BioC format. Then, we wrote a python script in order to parse these files. The aim of this script was to keep only genes/mutations/diseases/chemicals that exist in each article and the relationships between them (in which articles the entities are contained). There were 38.000 collections of articles which should be parsed and at the end of this process we should keep only biomedical entities along with the articles' ID that they appear.
At the beginning of the algorithm CSV files are made for each category of nodes and edges. Then for each article in each of the 38,000 collections, all the entities we are interested in are identified and added to the respective CSV file.
Although this process finished successfully after 5 hours, we just had CSV files with the information mentioned above. Next step was the import of this information into the graph database.
In our case we had over 20 million entities (Articles' id, Mutations, Genes, Diseases, Chemicals) and over 150 million relationships between them. These are the main entities that we imported to the graph database.

2.2.3 Import to the Graph Database

A database for the nature of this data needs to have the following criteria:

1) Easy and quickly data entry
2) Easy data browsing and visualization

3) Relationships creation
4) Quickly queries
5) Accessible to people without programming knowledge

Graph databases fit quite well with the above criteria. Apart from this and due to all of the features of graph databases that are described in section 1.3, we chose to organize our data to a graph database. There are a lot of open source graph databases such as AllegroGraph[10], ArangoDB[11], InfiniteGraph[12], Neo4j[13]. Taking into account the paper of Fernandes and Bernandino (Fernandes and Bernardino 2018) which compares the above graph databases we chose Neo4j.
Neo4j is an open-source graph database which uses Cypher, a declarative SQL-like language. Through Cypher the user is able to import data to the graph and to query the graph. Neo4j also allows the user to browse the graph database via Neo4j Browser. Neo4j Browser is a developer-focused tool that allows users to execute Cypher queries and visualize the results.
In addition to the Neo4j Browser the user is able to connect to the database and send queries via the Neo4j Python driver.
Neo4j includes a web server that can host different HTTP modules. In the default configuration, the web server will be started and host the BROWSER module (Neo4j Browser) at port 7474.

2.3 Graph database construction

After the installation and configuration of the Neo4j, we redesigned the python script so as to import data to the database whilst parsing the PubMed articles. As mentioned in section 1.3, graph databases are made up of nodes and edges. In our case the labels of nodes are “Articles”, “Genes”, “Mutations”, “Diseases”, “Chemicals” and the labels of edges are “Articles_Genes”, “Articles_Mutations”, “Articles_Diseases”, “Articles_Chemicals” and “Mutations_Genes”.
The nodes of articles have the PubMed id or the PMC id as properties, the nodes of genes have the name of the gene and the ncbi id of the gene as property. The nodes of diseases and chemicals have the MESH id as property. Concerning the mutations, their nodes have an attribute which is mutation’s type, RS-id or HGVS.
Concerning edges, they are undirected and they do not have any other properties except for the label of the edge.
The script for the Neo4j database building is described below:
For each article in each bioc.xml collection, the PubMed id is identified and the entities contained within it are located and stored in dictionaries by label. The nodes are then inserted into the graph. The edges between the entities' nodes and the articles’ nodes that contain them are inserted into the graph. Each node or edge addition to the graph requires a separate connection to the Neo4j database.
It is noteworthy that the TmVar 2.0 which has been used to detect mutations in these articles returns some additional information besides mutation’s name. In some cases, it locates the

[10] https://allegrograph.com/
[11] https://www.arangodb.com/
[12] https://infinitegraph.com/
[13] https://neo4j.com/

gene in which the mutation takes place, with the help of GNormPlus and the NCBI taxonomy id which is an identifier for a taxon in the Taxonomy Database of NCBI . When the corresponding gene is detected, an edge is added between the mutation and the gene nodes with the label “Mutation_Gene”. We keep only mutations with NCBI Taxonomy id = 9606 which is the id for homo sapiens.
Ideally, the above process would be completed for each of the 38,000 collections, and the graph would have been successfully constructed containing all the Pubmed entities that concern us.
The above process lasted longer than we expected. In addition to the large amount of data that had to be parsed, a large role in this delay seems to have been played by the multiple connections to the database within each loop. Each connection to the base required about 0.5 second. In our case it required about 150 million connections which means that the days required for only these connections are 600, 2 years. Even when we tried to speed up the process by using Python's Parallel library, there was no improvement.
This estimated time was prohibitive and we had to find other solutions.
We proceeded with some actions that would speed up the construction of the graph.
Below are some of the acceleration methods we used.

2.3.1 Speeding up graph database construction

- **Genes/Diseases/Chemicals separate import**

In another attempt to improve the construction time of the graph we tried to import Genes, Diseases and Chemicals before the parsing of PubMed corpus.
We got gene information from NCBI and Diseases and Chemicals from MESH.
Existing these categories in the graph, during the parsing these entities should not be detected and imported as nodes. Thus, instead of making the nodes in each loop, since they already exist in the graph, they are used only for the construction of the edges.

- **importing via csv files**

Neo4j enables the acceleration of dataimport by using csv files for nodes and edges. During the parsing of bioc.xml files we created some csv files of nodes and edges which were complemented from the entities (Articles, Mutations, Genes, Diseases, Chemicals) and the relationships between them (Article-Mutation, Article-Gene, Article-Disease, Article-Chemical, Gene-Mutation). After the end of parsing there were 5 csv files of nodes, one of each entity category and 5 csv files of edges. These files could be imported to the graph using simple Cypher commands. In this way, the connections required to the database are 10, the same number as the sum of nodes and edges categories.

- **splitting csv files or USING PERIODIC COMMIT LOAD CSV WITH HEADERS**

Importing CSV files that had over 10,000 lines proved to be unstables and resulted in many system crashes inhibiting significantly the data import process.
To address this, we added an extra step before importing csv files in which each file was split per 1000 lines.

- **Constraints**

When a node is added in the graph, a search is made on all the existing nodes of the specific label and if this node does not already exist, then it is added to the graph.This process is very time consuming especially in the case where the graph has a lot of nodes. This issue

can be resolved by creating constraints. The constraints automatically create a schema index in the graph database and the search described above requires logarithmic time complexity.

The user is able to specify unique constraints of a property on nodes with a specific label. Thus, we specified constraints for each node label. The setting up of constraints significantly accelerated the construction of the graph. Unfortunately, node key constraints, node property existence constraints and relationship property existence constraints are only available in Neo4j Enterprise Edition.

The time required for parsing and making csv files was 2 hours and for importing was 4 hours. After the completion of the construction of the graph it finally contained 22 Million nodes and 120 million edges. Specifically, the numbers of each category are shown in Table 2.1.

Nodes		**Relationships**	
Articles	27.208.776	Articles-Genes	33.202.889
Genes	37.743	Articles-Mutations	3.429.989
Mutations	1.138.097	Articles-Chemicals	93.595.435
Chemicals	348.017	Articles-Diseases	60.171.400
Diseases	11.952	Gene-Mutations	244.922
Total	28.744.585	**Total**	190.624.635

Table 2.1: "The numbers of Nodes and Edges of the graph"

2.4 Mutations

2.4.1 HGVS format

The main target of this thesis is the matching of a set of mutations with the mutations which exist in PubMed. This presupposes that the mutations are in the same format on both sides. Our first thought was to use the HGVS form of mutations because of what is mentioned in section 1.1.5.
TmVar, which was used to build the PubTator dataset, can detect SNP and HGVS variants in a text. Concerning HGVS variants, TmVar detects only the part of "description" as descripted in section 1.1.5. (For the mutation NM_134241.1:c.1234A>G TmVar detects only the c.1234A>T)
Thus, we had an incomplete form of mutations within a lot of articles.
Concerning the coding hgvs variants (mutations using the "c." positioning system), there were 71.686 in the graph. What was missing from these variants was the corresponding transcript. Finding all transcripts of all genes that coexist with these variants in articles and making the cartesian product between transcripts and coding hgvs variants was a way to

deal with the problem of incomplete form of hgvs coding variants. As mentioned in section 1.1.6, VariantValidator could do this work. VariantValidator requires the installation of MySQL, SQLite 3.8.0 and Postgres 9.5. After the installation of the above software and the configuration of VariantValidator we tried to derive the transcripts of genes. This process was time consuming as each of the 33,000 genes required approximately 2-3 minutes.

In order to speed up this process we bypassed the VariantValidator and we downloaded from RefSeq all transcripts of all genes and we made a csv file with transcripts as nodes and a csv file with the relationships between genes and transcripts. These csv files were imported to the graph which was enriched with the label “Transcripts” for nodes and the label “Gene-Transcript” for edges. This process was completed in less than 1 hour. The exact number of transcripts that entered the graph was 126.657 and the number of Gene_Transcripts edges was 126.741.

Thus, we combined the transcripts of genes which co-existed in articles with mutations with those mutations into a cartesian product. In the end we had mutations in complete hgvs format and we should find out which combination of all of them was the right one.
For this purpose we used another mode of VariantValidator which accepts as input an hgvs variant (e.g. NM_206933.2:c.6317C>G) and returns if the variant is valid or not. So, passing all variants that resulted from the cartesian product mentioned above through VariantValidator, we would know which ones are correct.
The times required were disappointing. Although we ran the VariantValidator for all the mutations in 10 different screens, the time required exceeded 100 days.
Another problem that we encountered was the following:
TmVar detects all the types of hgvs mutations. In our graph we find the following mutation types:

- 154.154 protein variants
- 71.868 coding variants
- 12.324 genomic variants
- 582 rna variants
- 1.866 m variants
- 56 n variants
- 130.014 dbSNP rs-id variants

protein (p.)	154.154
coding (c.)	71.868
genomic (g.)	12.324
mitochondrial (m.)	1.866
rna (r.)	582
non-coding (n.)	56

Table 2.2: “The number of each type of HGVS variants of the graph”

Concerning the HGVS variants of the graph (Table 2.2) we focused on protein, coding and genomic variants because these variants are related to a disease/phenotype.
As mentioned in section 1.1.6 VariantValidator accepts only genomic or coding hgvs variants for validation. Moreover, hgvs Python package allows the convention of a coding to protein variant (c. to p.) or a genomic to coding variant (g. to c.). These two facts in combination with the fact that neither the Mutalyzer accepts protein variants for validation, led us to place the hgvs protein in our future work.
These two problems mentioned in this section led us to leave the challenge of normalization of hgvs mutations and approach the goal of this work in a different way.
So, we were content with dbSNP mutations (RS-id). Many of the hgvs mutations detected by TmVar have been normalized to RS-id. This process is done as follows: TmVar detects a mutation, (eg c.1234A>T) then the genes that exist in the same sentence-paragraph are detected and only the gene that initially has the position where the mutation is reported is kept (it is within the boundaries of the gene) and at the specific site of the genome there is the specific nucleotide referred to in the mutation.Then, once the location of the genome, the gene and the nucleotide change are known, the mutation is normalized to RS-id. (figure 2.1)

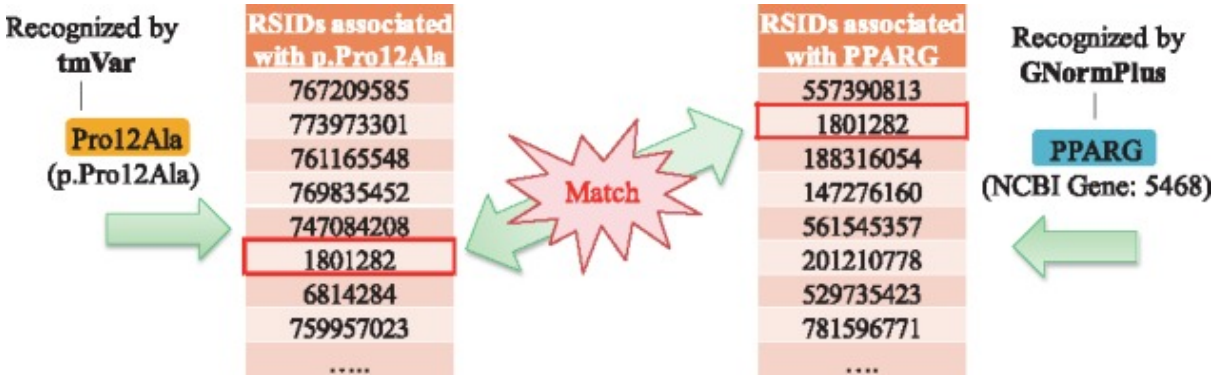

Figure 2.1: "Example of variant normalization process by TmVar 2.0" For each gene it is obtained its list of associated RSIDSs from dbSNP. Another one list is obtained with RSIDs associated with the target mutation. Any RSID found in both lists is returned as a candidate for the normalized form of the mutation.(Wei et al. 2018)

2.5 Graph enrichment

Once the construction of the graph is completed there is the possibility of searching for entities and relationships through Neo4j Browser via queries in the Cypher language. The user has the ability to navigate the graph through Neo4j Browser with its graphical environment. Additionally the user is able to send a query for a specific entity or for a specific category of entities associated with it.
In the above cases the user can manage the results manually and browse the graph to find various information that is interested in.

The problem of managing the results returned by the graph occurs in the case of massive queries or more generally of multiple results.
To deal with this we had to enrich the graph with more information that would help filtering the results.

A mutation that exists in an article does not mean that there is any important information about it. It could be in a table with other 50 mutations or a mutation could be referenced once in an article. On the contrary, there are articles that deal with a specific mutation and provide a complete analysis related to its significance and effect.
In our graph we had only the information of which mutation exists in which articles. It would be useful if there was a metric that would determine the significance of a mutation within an article.

TF-IDF (Term Frequency - Inverse Document Frequency) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents.
Therefore, we calculated tf-idf for every mutation and it was added to nodes as an attribute. This attribute enables the user to determine the importance of a mutation in an article.
We also downloaded from dbSNP's ftp site the "GCF_000001405.39.gz'' file[14] which contains all known RS-ids with the relevant information. This information concerns the pathogenicity of the mutations (mutations that exist within the ClinVar), the allele frequency of a mutation according to the 1000 Genomes Project and the information regarding the regulatory effect of the mutation, namely if it is nonsense, missense or silent.
All this information was added as an attribute to mutations, easing the navigation of a user on the graph.

2.6 Topic modeling

In general this work belongs to the field of mass querying large corpora of biomedical documents in order to extract a subset of articles of interest. After submitting a query, our pipeline does not export a flat set of related articles. A flat set would be difficult to use in a downstream analysis. In contrast it generates an easy to navigate graph, containing all the resulting articles as nodes along with all the rich information that has been described either as nodes or as properties. Nevertheless even when the resulting graph contains as few as 100 articles, it is difficult for a user to locate interesting patterns just by visually inspecting it. For this purpose we applied topic modeling on the text of the resulting articles.
Towards this we tried to retrieve abstracts from PubMed via the Biopython package and the module of Entrez. Given a list of PubMed ids, these articles are returned though API. Thus, trying to retrieve the abstracts of some articles that were resulted from a query to our database, we realized that the estimated time for up to 20 articles was tolerable. When the number of articles exceeded 20 the waiting time was not practical for ordinary use. This was expected since the API method is not suitable for downloading large volumes of data.

[14] https://ftp.ncbi.nih.gov/snp//latest_release/VCF/

2.7 Random file access

The Pubmed corpus which we parsed as mentioned in section 1.2.2 includes not only the entities we saved in our graph database but also the text of the articles. Therefore, we parsed again the Pubmed corpus to keep only the text. In order to have quick access to these texts we used the Random File Access method. This method enables quick access to specific records rather than having to read the file sequentially. According to this method, in a file there should be all the texts of the articles and in a second file should contain the Pubmed of each article and the offset that starts and ends in the first file.
Thus, we made a file which included the texts of all articles and another file with the indexing of each article in the first file. (figure 2.2)

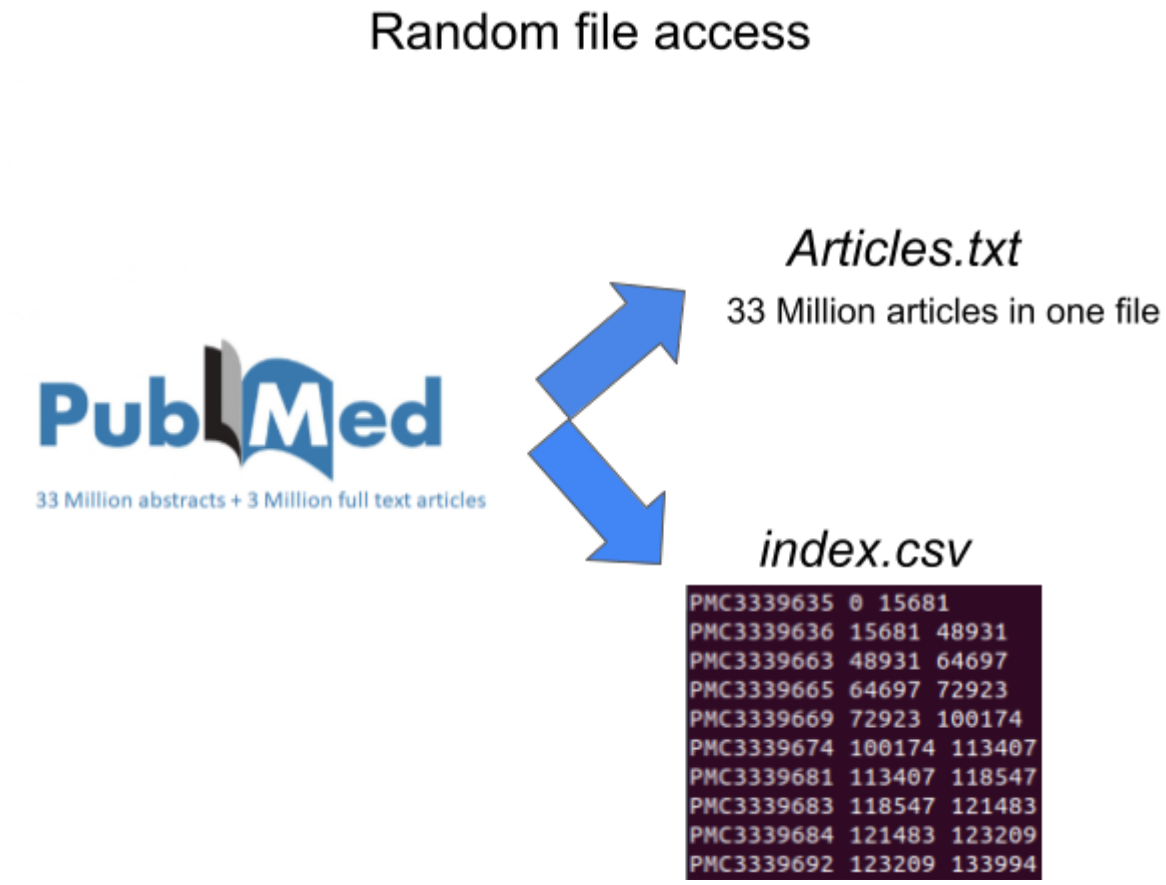

Figure 2.2: "Pre-processing steps for Random File Access method" . We create a file (index.csv) that contains the offsets and length (in bytes) of each Pubmed or PMC ID. This file is loaded in a hashtable in memory where the ID is the key and the tuple offset,length is the value. Then when we want to access the text of a given article we first acquire this tuple and we make a random read access in the file containing the articles. This process takes logarithmic time (binary search) for accessing a single article over the complete corpus.

In this way we had easy and flexible local access to all articles' text. For example acquiring the complete text of 100 random pubmed IDs required less than a minute of computation time. One way of analyzing a set of texts, as mentioned in section 1.4, is topic modeling.

2.8 Topic Modeling

2.8.1 LDA

Initially, we used the LDA algorithm for a set of articles which resulted from a query to our graph database. There are many ways to implement this algorithm. We performed LDA with Gensim (Rehurek and Sojka, 2011) which is an open source Python library for document representation as semantic vectors.
The LDA algorithm requires some pre-processing steps. So, we should perform the text preprocessing steps that are described in section 1.4.1. In addition to these steps, before the final building of the LDA model, the user must choose the number of topics that will be returned. Finding the optimum number of topics is a known issue in the LDA algorithm. A solution to this problem is to build many LDA models with different values of the "Number of Topics"t parameter or by checking if the same keywords are repeated in different topics. Therefore, the above is a practical problem in implementing an LDA model.
The purpose of this thesis is to build a tool that will not necessarily be used by people who will have programming knowledge or great familiarity with specific areas such as topic modeling. Taking into account that, we decided that the LDA algorithm does not fit the purpose of this project.

2.8.2 Top2Vec

As mentioned in section 1.4.2, Top2Vec does not require neither preprocessing steps nor configuration of parameters like the "Number of Topics". So, we configured the Top2Vec algorithm so as to accept as input the texts from the results of a query in our graph database. Then the pipeline proceeds as follows:

1) Create jointly embedded document and word vectors using Doc2Vec.
2) Create lower dimensional embedding of document vectors using UMAP.
3) Find dense areas of documents using HDBSCAN.
4) For each dense area calculate the centroid of document vectors in the original dimension, this is the topic vector.
5) Find n-closest word vectors to the resulting topic vector.

After these steps, a model is built and ready for use. Top2Vec has some modules that return information about the model. Some of these are:

- **"Get Number of Topics"** which returns the number of topics that Top2Vec has found in the corpus.
- **"Get Topic Sizes"** which return the number of documents most similar to each topic.
- **"Search Topics"** which enables the user to search for topics most similar to a keyword.

```
>>>topic_words,word_scores,topic_scores,topic_nums=model.search_topics(keywords=["medicine"], num_topics=5)
>>> topic_nums
[21, 29, 9, 61, 48]
>>> topic_scores
[0.4468, 0.381, 0.2779, 0.2566, 0.2515]
```

In the above example, topic 21 was the most similar topic to “medicine” with a cosine similarity of 0.4468. (Values can be from least similar 0, to most similar 1)

- **“Generate Word Clouds”** which returns to the user via word clouds representation a grouping of topics
- **“Search Documents by Topic”** which searches for a specific topic.
- **“Semantic Search Documents by Keywords”** which searches documents for content semantically similar to keywords.
- **“Similar Keywords”** which search for similar words to a keyword.

Therefore, the user can run the modules described above to explore the data generated from a query in the database. Additionally, the dimensionality reduction and the clustering graphs are returned to the user.

With the addition of Top2Vec for topic modeling, the pipeline scheme completed. (Figure 2.3)

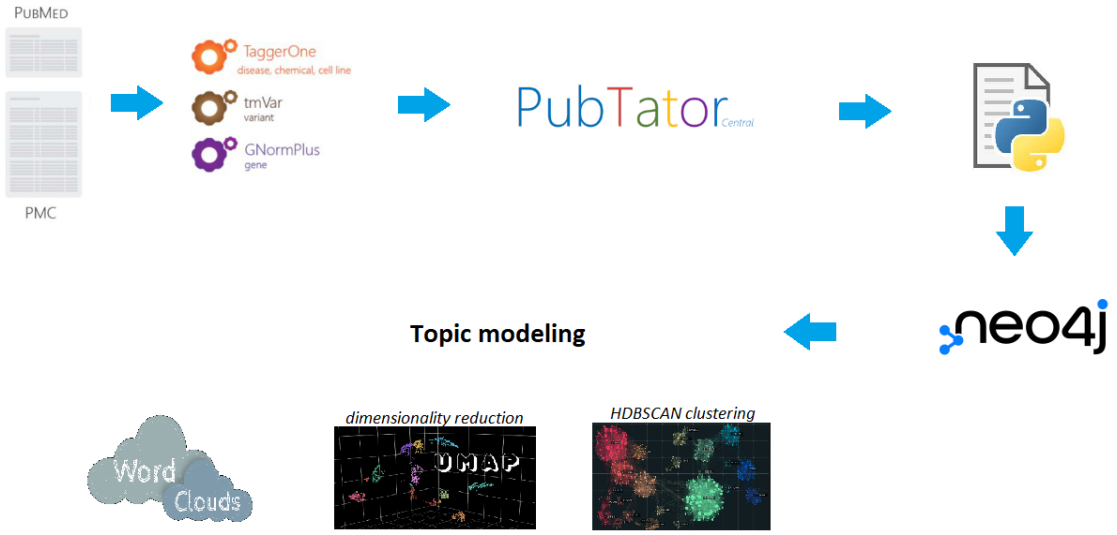

Figure 2.3: “Our final pipeline”. Pubmed abstracts and full-text PMC articles were commented on by TmVar, GNorm, and Tagger one to identify mutations, genes, diseases, and chemicals, creating the PubTator BioCXML files. We wrote a python script in order to parse the BioCXML files and to keep only the entities. These entities are inserted into the Neo4j graph database. Finally the results of the queries in neo4j are given as inputs for topic modeling through Top2Vec

2.9 Argo Workflow

As described in the Graph database construction section, the construction of our graph requires a lot of steps. Specifically, until the completion of the construction of the graph, the following steps are required:

1) Download data from PubTator
2) Uncompress the data
3) Parse the data and create csv files with nodes and edges
4) Setup Neo4j database
5) Import nodes
6) Import edges

Additionally, the above steps should be repeated for each graph update. For this reason, we used the Argo workflow[15] with the aim of automating and speeding up the graph construction process even more. Argo Workflow enables the user to run jobs in parallel and through a directed acyclic graph (DAG) it manages the dependencies between the tasks.
It is important to note that Argo Workflows are Research Objects that encapsulate the complete analysis which can be reproduced with a minimal effort (Nikolov, 2021).

The six layers of the DAG in figure 2.4 correspond to the six steps mentioned above.
As also shown in the DAG, the steps 1,2 and 5 run in parallel. Especially the fifth step (Import of nodes), which was the most time consuming, by using the Argo the importing of nodes was greatly accelerated. The same parallelism can not occur when importing the edges as multiple threads cannot write information to the same node at the same time.

[15] https://argoproj.github.io/argo-workflows/

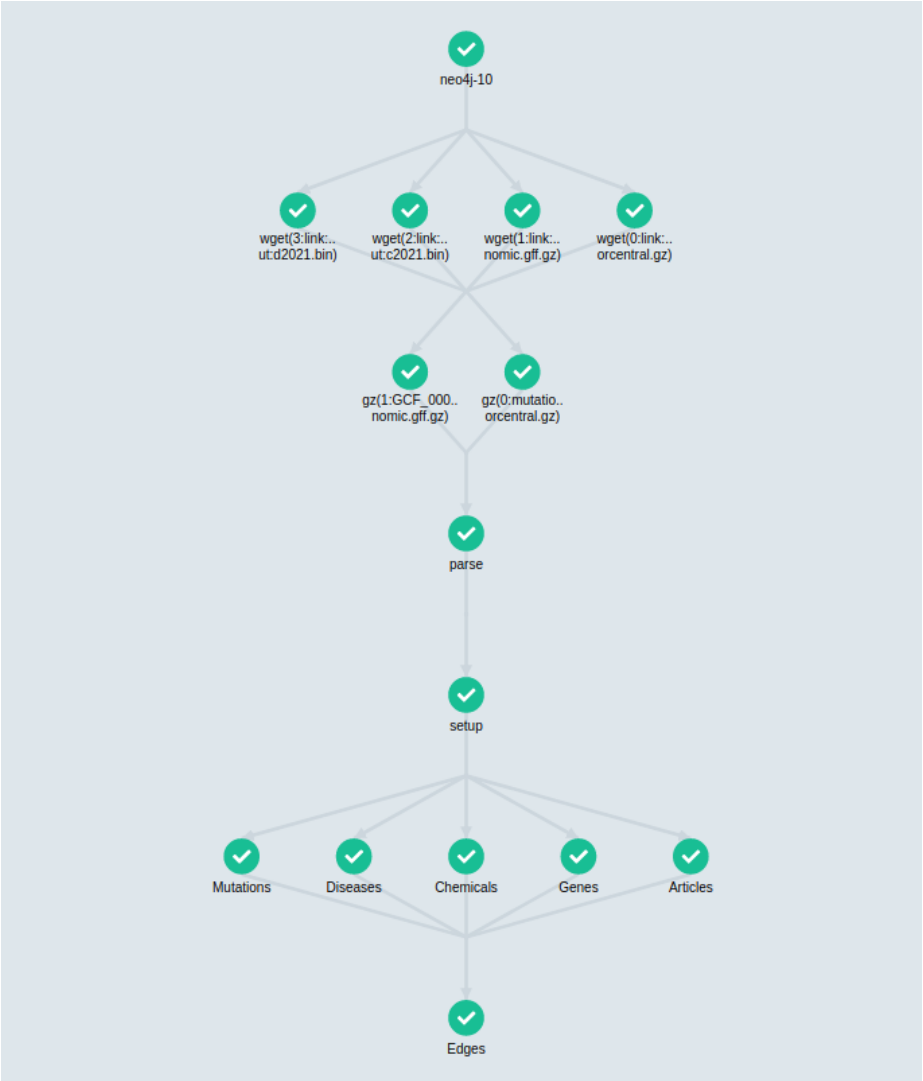

Figure 2.4: "The steps of the building of the graph in Argo workflow". The first step is the setting up of Neo4j, the second one is the downloading of the required files. The third one is the uncompressing of the files. Fourth step is the parsing of BioCXML files. The fifth one is the preparation of the database for its construction. The last two steps are the importing of nodes in parallel and the importing of edges.

3. Results-Use cases

3.1 Exploring the graph

After a user logs in to the address and port that neo4j has installed, the interface of Neo4j Server that appears is shown in the figure 3.1.

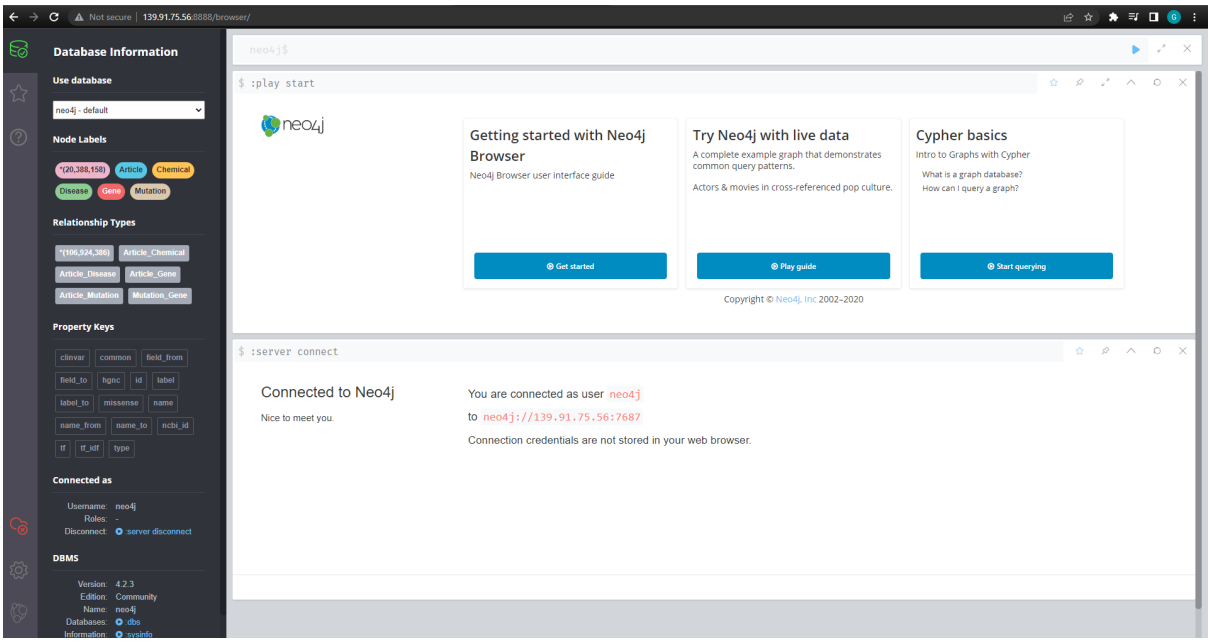

Figure 3.1: "Neo4j Server interface"

Once users connect to the Neo4j server they have the ability to browse the graph. There are several options for navigating the graph.
There is a possibility by selecting a category from the Nodes on the left side, to display some of the results of this category. The figure 3.2 shows the result after selecting the category of nodes "Articles".

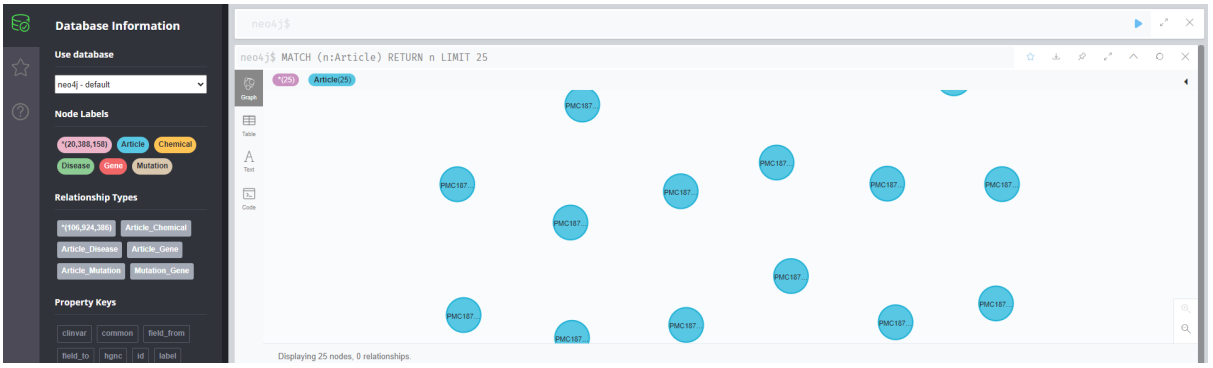

Figure 3.2: " Graphic representation of articles"

If users are interested in a specific entity of a certain category they can do a specific search. (Figure 3.3)

Figure 3.3: "Query for a specific Article"

In the above case we searched for the article with PMC-id 3789669 and the corresponding node was returned (Figure 3.3). By clicking on the node we can expand it and display all its relationships (Figure 3.4).

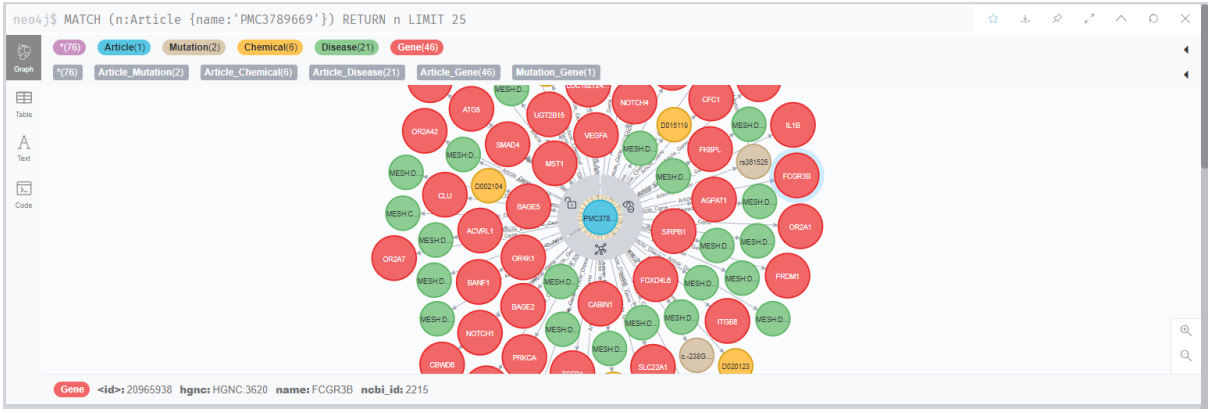

Figure 3.4: "Article's contents"

Then we can isolate the unique rs-id mutation that exists within the article (rs391525) and observe that it is also linked to the TNF gene which is the gene in which this mutation occurs (Figure 3.5).

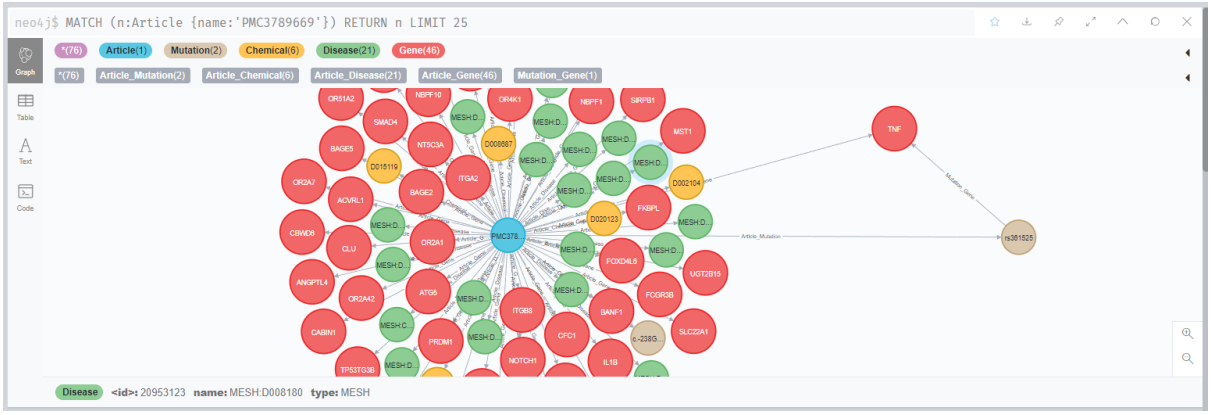

Figure 3.5: "Connections between the contents of the article"

If we are interested in this mutation we can expand its node to see the rest of its relationships (Figure 3.6). We notice that this mutation exists in many articles, specifically in 533.

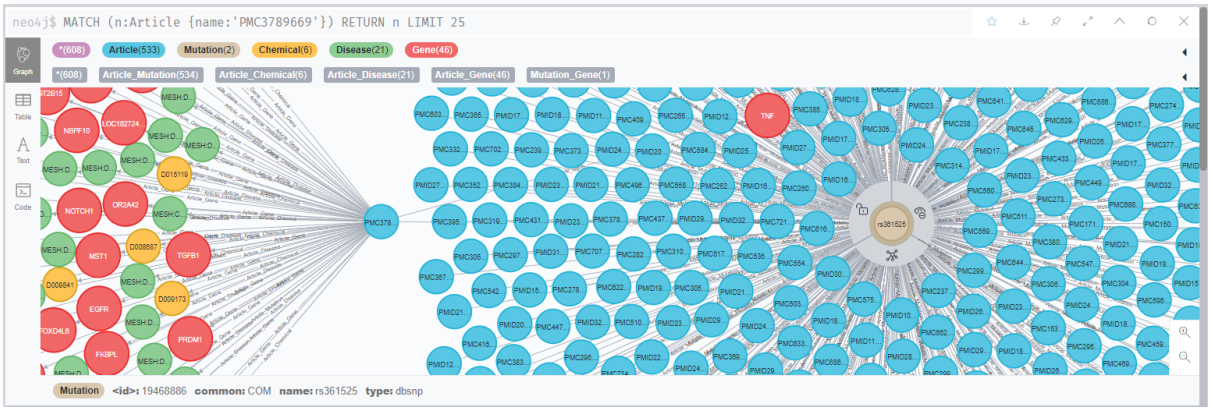

Figure 3.6: "Articles (blue nodes) in which the mutation (brown node) appears"

Also as we notice at the bottom of the figure, this particular mutation has the attribute "common" which indicates that it is a common mutation. As a common mutation we mean that this mutation occurs in more than 2% of the population in the 1000genome project. Queries can be even more complicated. We can include more than one relationship in a query. In the following example we ask for the return of all the first 100 diseases that exist in articles that contain a dbSNP mutation with tf-idf>0.5.

```
MATCH p=(m:Mutation)-[r:Article_Mutation]-(a:Article)-[rr:Article_Disease]-(d:Disease)
WHERE r.tf_idf>0.5 AND m.type='dbsnp'
RETURN p LIMIT 100
```

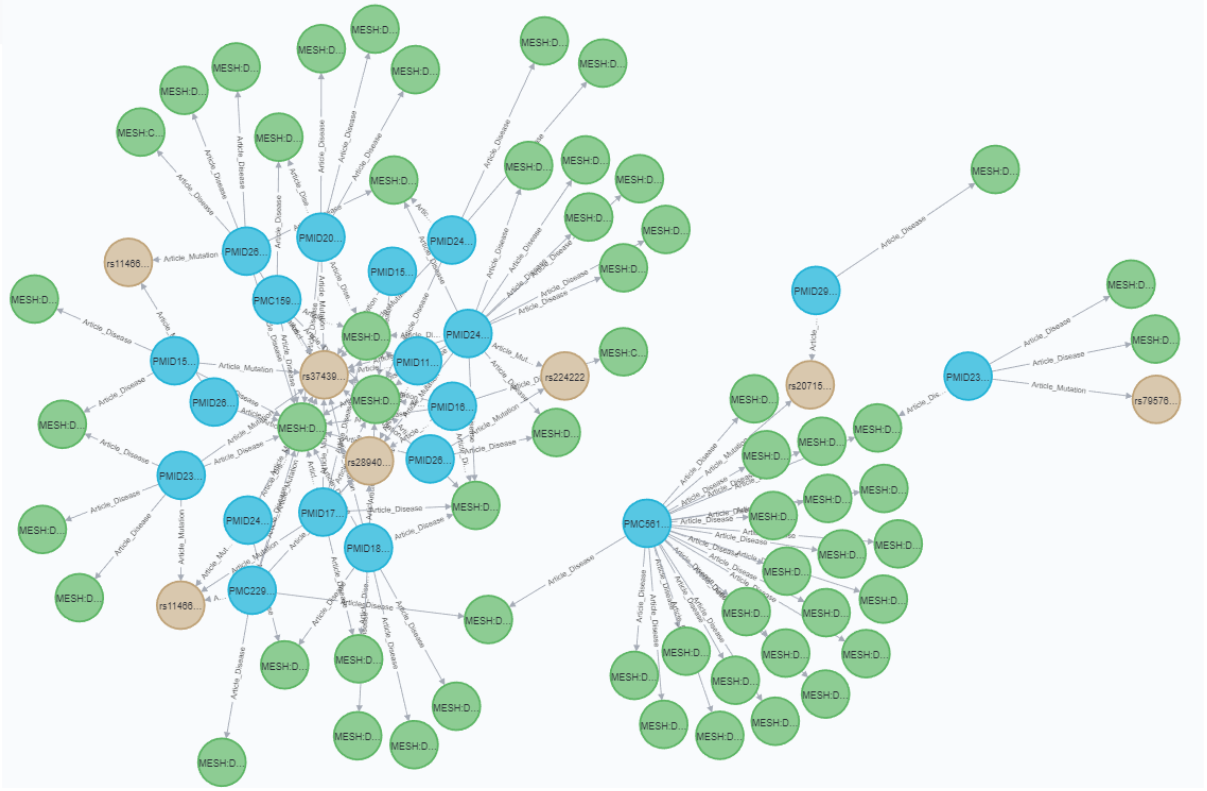

Figure 3.7: "Results of a query with multiple relationships"

Through a query like the one above (Figure 3.7), relationships between two categories can be extracted within the graph.

3.2 Export relationships

Just navigating through the Neo4j Browser makes it difficult for the user to reach a conclusion. We have created a script that allows the user to give a mutation (with RS-id) and return the diseases that coexist most often in articles with this mutation. This application may be useful to anyone looking for possible relationships between categories of entities. In the following example (Figure 3.8) we have given as input the mutation rs165599 and the script returns the 5 most frequently disease terms (with Mesh id) which coexist with this mutation in bibliography. As shown in the following figure the first disease term is this with mesh id D012559. This id corresponds to the schizophrenia term. Searching for the same mutation in Clinvar we observe that there is no report of association with schizophrenia. On the contrary, there are several articles in Pubmed that study the association of the rs165599 mutation with schizophrenia (Figure 3.9). Overall this script explores possible relationships between a mutation and a disease.

python /private/use_case_2.py rs165599

```
Variant: rs165599 ---> 160 articles
In these articles exist 679 disease terms
1 --> ['D012559'] 68.75 %
2 --> ['D001523'] 65.0 %
3 --> ['D000275'] 48.12 %
4 --> ['D001714'] 34.38 %
5 --> ['D003072'] 33.12 %
```

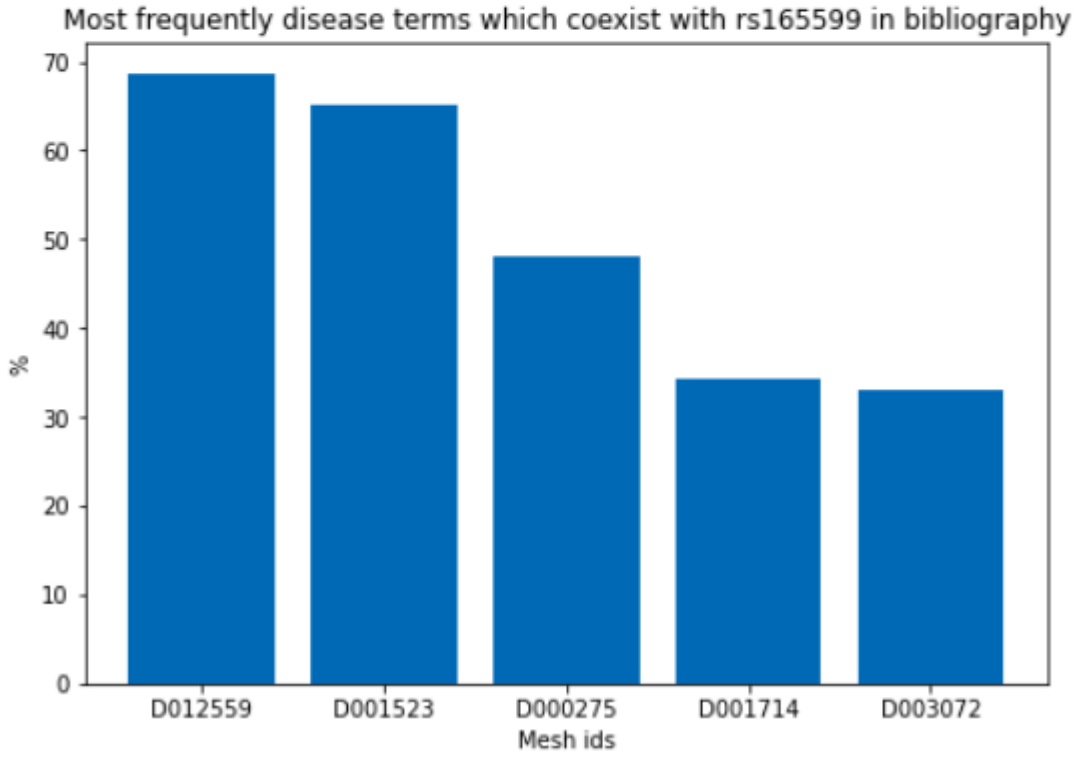

Figure 3.8: "Most frequently disease terms which coexist with rs165599 mutation in PubMed"

```
python /private/use_case_2.py rs1042713

Variant: rs1042713 ---> 170 articles
867 disease terms exist in these articles
1 --> ['D001249'] 42.94 %
2 --> ['D006973'] 34.71 %
3 --> ['D009765'] 31.76 %
4 --> ['D003920'] 30.59 %
5 --> ['D002318'] 30.0 %
```

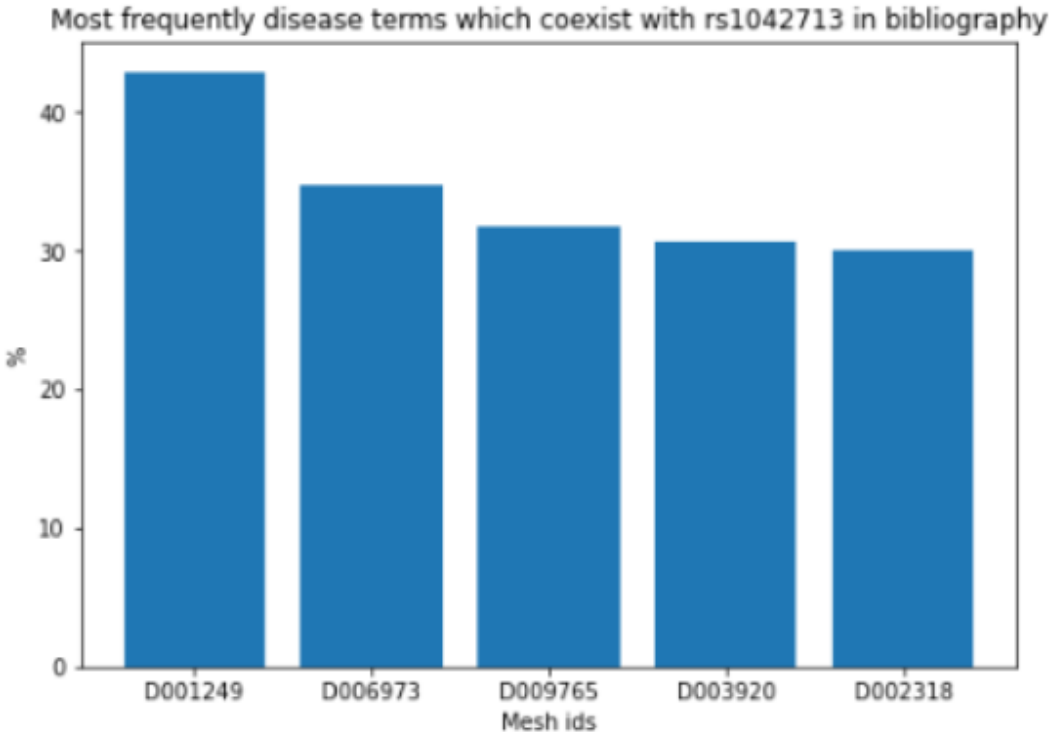

Figure 3.9: "Most frequently disease terms which coexist with rs1042713 mutation in PubMed"

3.3 Massive queries via a csv file

Neo4j enables the user to send massive queries via csv files. The csv file that is loaded to Neo4j must have a specific format. Specifically, there should be a header that is the attribute that we will look for in the graph. This csv file must have one entity per line. A csv file with a mutation per line can be loaded and sent to the graph as a query. The result of this query could be the number of mutations that exist in the graph, the articles that are connected with these mutations or the diseases that coexist in articles with these mutations. The user has many options as to what the query will return. The impressive thing about this process is the time it takes. For example a query of 35.000 rs-ids, requires 2 seconds (Figure 3.10).

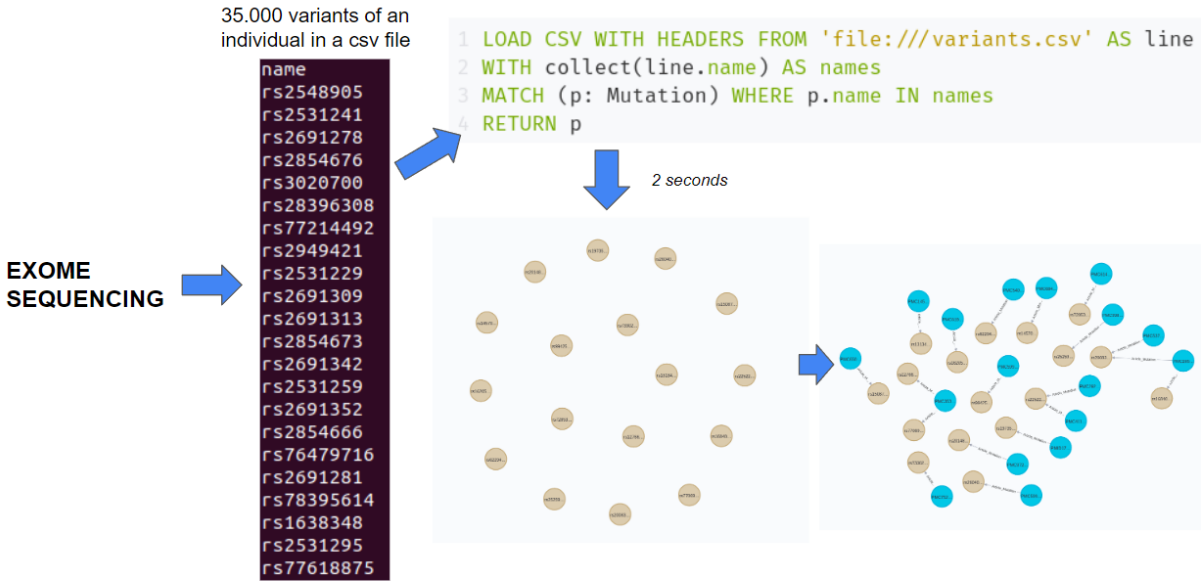

Figure 3.10: "From exome sequencing to the results of the graph"

When we send a query of 35.000 mutations to the graph, the average number of results is 5000-6000 mutations. If we select to also include the articles that mention these mutations, then it also returns approximately 15.000 articles.

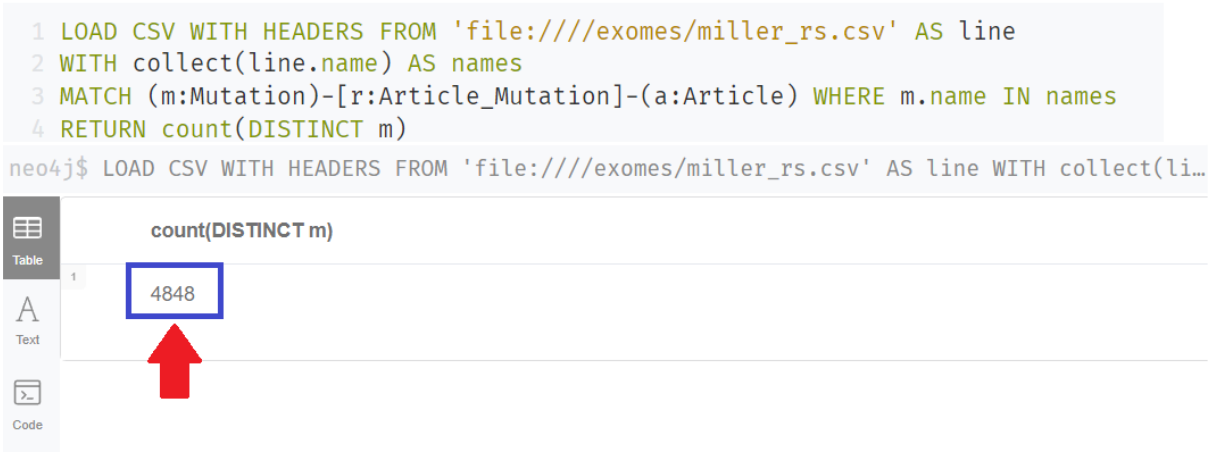

Figure 3.11: "The number of returned articles of a query without filters"

Managing the graph and the contained information from 15.000 articles is practically impossible. With the help of the filters that were added (section Graph enrichment) we can further filter the results. As we can see below, having a csv file of 35.000 mutations of a person, we can query the complete file and get the 4848 mutations that also exist in the graph (Figure 3.11). Using the TF-IDF filter for the same file the number of returned articles is significantly reduced. The higher the TF-IDF that we use as a filter, the lower the number of returned articles. (table 3.1)

TF-IDF	Number of returned articles
without TF-IDF filter	4343
>0.5	565
>2	121

Table 3.1: "The effect of TF-IDF filter on returned articles"

Filters	Number of returned articles
without filters	4343
with ClinVar filter	1374
With ClinVar filter and TF-IDF>2	69

Table 3.2: "The effect of ClinVar filter on returned articles"

The user has the ability to further limit the results by using the Clinvar filter. As it shown in Table 3.2 the results that are returned are significantly reduced with the ClinVar filter. As also shown in Table 3.2 when the TF-IDF and ClinVar filters are applied in combination the number of returned items is manageable.

After some articles have been returned from a query the user can use them for topic modeling as mentioned in the chapter Topic modeling. Initially the articles resulting from the query are extracted by the method of random file access as mentioned in the corresponding chapter. Then the texts of these articles are given as input to Top2Vec for topic modeling. Figure 21 includes the steps from the loading of a file with mutations to topic modeling with Top2Vec.

```
LOAD CSV WITH HEADERS FROM 'file:////exomes/miller_rs.csv' AS line
WITH collect(line.name) AS names
MATCH (a:Article)-[r:Article_Mutation]-(m:Mutation) WHERE m.name IN names AND r.tf_idf>1
RETURN collect(distinct(a.name)) AS article
```

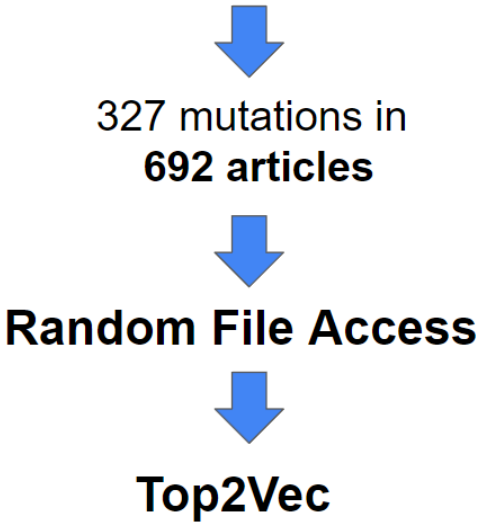

Figure 3.12: "The steps from the loading of a file with mutations to topic modeling with Top2Vec"

The file uploaded to the graph (figure 3.12) contains 35,000 mutations of one individual. Setting the tf-idf filter to greater than 1.327 mutations are returned that are contained in 692 articles.
The texts of 692 articles are available to be given as input to Top2Vec within 10 seconds due to the Random File Access method. Top2Vec provides various ways of representing the results of topic modeling. One of them is word clouds. Figure 3.13represents the 7 topics that emerged from Top2Vec.

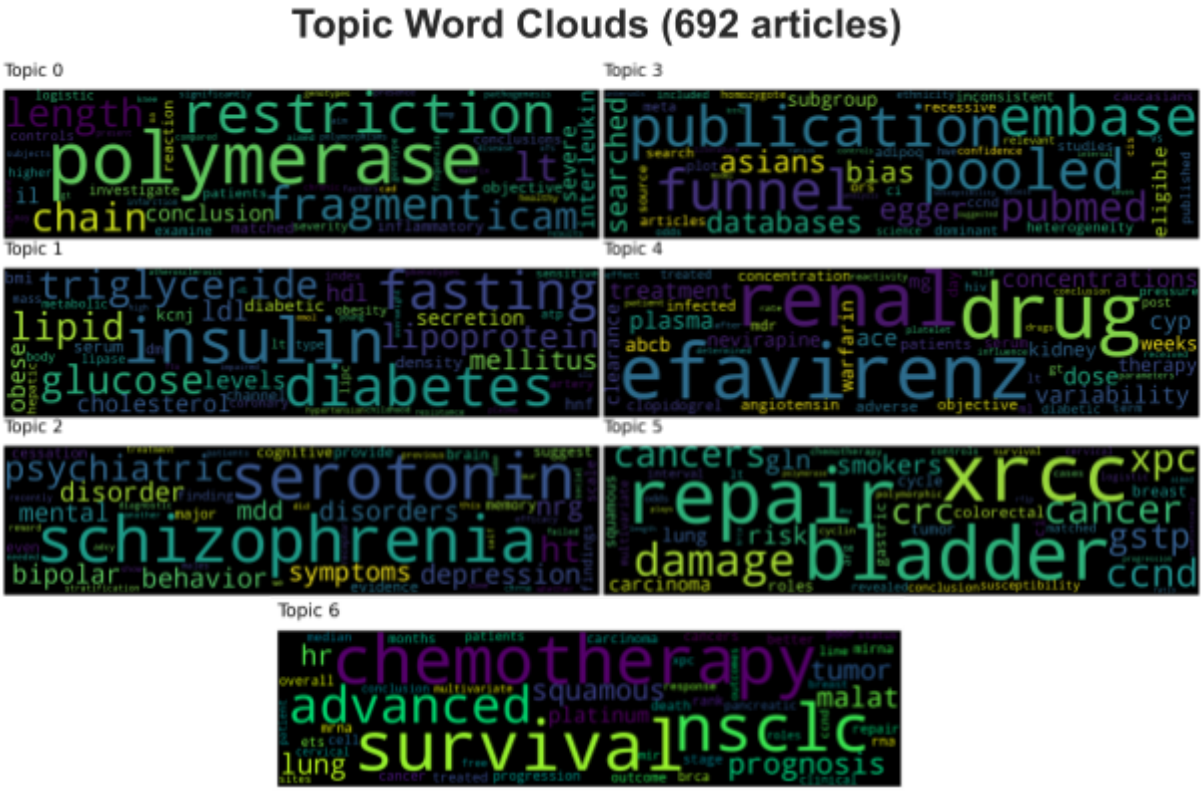

Figure 3.13: "Topic Word Clouds of 692 articles"

The user is able to select one of the topics and to search for the documents related to the specific topic. By selecting a topic, the most similar documents to it could be returned with a cosine similarity of the document. For example if a user choose the topic 2 of figure 22 which is related to schizophrenia, the documents of Figure 23 are returned:

Document: 68, Score: 0.8039693832397461

Negative association between T102C polymorphism of the 5-HT2a receptor gene and schizophrenia in Japan.Serotonin (5-hydroxytryptamine, 5-HT) may play an important role in the pathogenesis of schizophrenia. Previous studies suggested that the efficacy of atypical neuroleptic drugs (e.g., risperidone and clozapine) on negative symptoms may be related to the 5-HT2a receptor. Although association studies between MspI polymorphism (T102C) and the 5-HT2a receptor gene and schizophrenia have been reported, their results are still controversial. The aim of this study was to examine the association between T102C polymorphism of the 5-HT2a receptor gene and schizophrenia as well as the association between the polymorphism and negative symptoms in a Japanese population (106 patients with schizophrenia and 109 healthy controls). No significant positive associations were observed. Our results suggest that the 5-HT2a receptor gene is not involved in the pathogenesis of schizophrenia or negative symptoms.

Document: 72, Score: 0.7858105897903442

Lack of allelic association between 102T/C polymorphism of serotonin receptor type 2A gene and schizophrenia in Chinese.Recent studies have reported an association between a 102T/C polymorphism of serotonin receptor type 2A gene (5-HT2A) and schizophrenia. In addition, an association was detected between a 102T/C polymorphism of the 5-HT2A receptor gene and drug response to clozapine in the treatment of schizophrenic patients. These studies suggest an important role of the 5-HT2A gene in schizophrenia. To study the possible involvement of the 5-HT2A gene in the pathogenesis of schizophrenia, a case-control association study was carried out in a Chinese population from Taiwan. No significant differences of genotype distributions, allele frequencies and homozygosity were detected between schizophrenic patients (n = 177) and nonpsychiatric controls (n = 98). When subjects were divided into subgroups according to sex, still no differences of allele frequencies or genotype distributions were noted between patients and controls. Our data do not support an allelic association between the 102T/C polymorphism of the 5-HT2A receptor gene and schizophrenia in Chinese population.

Document: 66, Score: 0.7960748672485352

Differential expression and parent-of-origin effect of the 5-HT2A receptor gene C102T polymorphism: analysis of suicidality in schizophrenia and bipolar disorder.The serotonin 2A (5-HT2A) receptor gene has been implicated in the pathogenesis of suicidal behavior by a genetic association between the 5-HT2A C102T silent polymorphism and suicidality in patients with major depression. However, a recent meta-analysis failed to confirm this association. We developed an improved quantitative assay for the measurement of allele-specific expression of the 5-HT2A gene, and find that the ratio of C/T allele expression in the pre-frontal cortex of heterozygous suicide victims (n = 10) was significantly decreased in comparison with the non-suicide group (n = 10) (P = 0.049). Because the 5-HT2A gene is subject to imprinting, the parent-of-origin may affect the inheritance of suicidal behavior. Thus we examined the parental origin of specific alleles for genetic association in a genetic family-based sample of major psychoses in which information on suicidal behavior was available. No association between the 5-HT2A C102T polymorphism and suicidal behavior in major psychoses was detected with the transmission/disequilibrium test (TDT).

Document: 74, Score: 0.7664878964424133

Variations in 5-HT2A influence spatial cognitive abilities and working memory.BACKGROUND: 5-hydroxytryptamine receptor 2A (5-HT2A) participates in diverse psychiatric disorders by regulating the activity of serotonin. Some previous studies have also suggested that the receptor is involved in cognitive abilities of disease groups. We hypothesize that some functional genetic variants in 5-HT2A have certain specific influences on cognitive abilities in a normal population. METHOD: To confirm this hypothesis, two polymorphisms (rs6313 and rs4941573) in 5-HT2A were selected, and a population-based study was performed in a young healthy Chinese Han cohort. RESULTS: The results indicated that the rs6313 and rs4941573 were associated with touching blocks and mental rotation-3D error ratio in males, and the rs4941573 was associated with visuo-spatial working memory in the whole cohort. CONCLUSION: All the findings suggest that 5-HT2A participates in human spatial cognitive abilities and spatial working memory.

Figure 3.14: "the closest documents to topic 2 along with the cosine similarity score"

The 4 above documents (Figure 3.14) are the closest to topic 2 (schizophrenia) and they are easy to read and can lead the user to useful conclusions.

The user can search for documents for content semantically similar to keywords. For example if the user is interested in searching for the top 5 documents closest to the word "cancer", the documents in figure 3.15 are returned:

Document: 142, Score: 0.47784173488616943
Association between the FAS rs2234767G/A polymorphism and cancer risk: a systematic review and meta-analysis.Abnormal regulation of apoptosis can lead to carcinogenesis. Single nucleotide polymorphisms in apoptotic genes have been associated with cancer risk, such as the FAS rs2234767G/A polymorphism, which alters transcription of the FAS promoter. Downregulation of FAS, with resultant cellular resistance to death signals, has been found in many cancers. However, the association between the FAS rs2234767G/A polymorphism and cancer risk is still controversial. Here, we performed a meta-analysis including 41 articles (44 case-control studies, 17,814 cases and 24,307 controls) identified from PubMed and Chinese language (CNKI and WanFang) databases related to cancer susceptibility and the FAS rs2234767G/A polymorphism. We used odds ratios (ORs) and 95% confidence intervals (CIs) to assess the strength of the associations. We found that the rs2234767 G-allele was a protective factor for cancer risk (GG vs. AA: OR=0.88, 95% CI=0.79-0.98; GG+GA vs. AA: OR=0.87, 95% CI=0.79-0.96). Similar associations were detected in the "source of control", ethnicity, and cancer type subgroups. Further studies on a larger sample size and considering gene-environment interactions should be conducted to confirm the role of FAS polymorphisms, especially rs2234767G/A, in cancer risk.
Document: 305, Score: 0.4495234787464142
[Association between CCND1 G870A polymorphism and radiotherapy response in high-risk human papillomavirus-related cervical cancer].OBJECTIVE: To investigate the correlation of cyclin D1 (CCND1) G870A single nucleotide polymorphism (SNP) with radiotherapy response in patients with high risk human papillomavirus (HR-HPV) related cervical cancer. : METHODS: A total of 273 patients with cervical cancer, who were confirmed by histopathology and hybrid capture 2 (HC-2) assay and treated by radiotherapy, were enrolled for this study. The correlation of CCND1 G870A polymorphism with tumor response in patients was assessed. : RESULTS: Compared with patients with AA genotype, the patients with GG genotype and AA genotype showed lower sensitivity to radio-therapy treatment (adjusted ORGA=2.69, 95% CI 1.28-5.67 and adjusted ORGG=3.28, 95% CI 1.47-7.29, respectively), an increase in risks of recurrence/metastasis (adjusted ORGA=2.52, 95% CI 1.12-5.63 and adjusted ORGG=3.95, 95% CI 1.68-9.26, respectively), and shorter recurrence/metastasis-free survival (PGA=0.010 and PGG=0.045). : CONCLUSION: G870A polymorphism is a frequent variation that could be used for evaluate the radio-sensitivity and prognosis for patients with HR-HPV related cervical cancer.
Document: 165, Score: 0.4649355709552765
Association between lncRNA H19 (rs217727, rs2735971 and rs3024270) polymorphisms and the risk of bladder cancer in Chinese population.BACKGROUND: The Long non-coding RNA (lncRNA) H19 is involved in carcinogenesis, progression, and metastasis. However, the association between genetic variants in H19 and bladder cancer susceptibility has not been reported. This case-control study assessed the association between H19 genetic variants and susceptibility to bladder cancer in a Chinese Han population. METHODS: In this study, 200 patients with bladder cancer and 200 healthy controls were surveyed and compared for frequencies of the genotypes of the H19 gene. Logistic regression analysis and the chi-square test were employed as statistical methods. Odds ratio (OR) and its corresponding 95% confidence interval (95% CI) were calculated to estimate the strength of association between H19 polymorphisms and risk of bladder cancer. RESULTS: We analyzed the frequency of three lncRNA H19 SNPs in 200 bladder cancer patients and 200 healthy controls. Carriers of variant rs217727 heterozygous genotype showed increased bladder cancer risk (P=0.008). Further stratified analyses revealed that the association between bladder cancer risk and variant genotypes of rs217727 was more profound in smokers. Furthermore, decreased risk of invasive bladder cancer was found in carrying rs3024270 CC patients. CONCLUSIONS: Our results provided the evidence that rs217727 in H19 was associated with elevated risk of bladder cancer, which may be a potential biomarker for predicting bladder cancer susceptibility. Furthermore, invasive bladder cancer in carrying rs3024270 CC genotype maybe have a good prognosis.
Document: 314, Score: 0.43818914890289307
CCND1 rs9344 polymorphisms are associated with the genetic susceptibility to cervical cancer in Chinese population.Cyclin D1, with a common G/A polymorphism in rs9344, is an essential regulator of the G1 phase in cell cycles and plays an important role in several tumor types, and the homology of cyclin D1 with human papillomavirus (HPV)-16 E7 brought our attention to CCND1 gene in cervical cancer. A total of 738 native Chinese subjects consist of 327 cases and 411 controls were enrolled in this study. CCND1 genotyping was analyzed by polymerase chain reaction-restriction fragment length polymorphism (PCR-RFLP) and partially verified by sequencing of genomic DNA and cDNA. The transcription of cyclin D1 mRNA isoforms was analyzed by quantitative PCR; expression of protein isoforms by immunohistochemistry and Western blotting. We observed that the AA genotype had decreased risk of developing cervical cancer (odds ratio [OR] = 0.332; 95% confidence interval [CI] = 0.113-0.978; P = 0.045). The two mRNA isoforms were both transcripted from A and G allele. Transcript b decreased in squamous cell carcinoma of the uterine cervix (SCCUC) group (P = 0.004), especially poorly differentiated group (P = 0.004), and in G allele group of normal subjects (P = 0.001). In immunohistochemistry analysis, cyclins D1, D1a, and D1b failed to correlate with cervical cancer (P = 0.808, 0.445, and 0.095). However, cyclin D1b was downregulated in SCCUC group analyzed by Western blotting (P = 0.039). This study indicates that CCND1 rs9344 polymorphisms confer host susceptibility to cervical cancer. A allele possesses a relative protective effect probably through the cyclin D1b's inhibition on HPV carcinogenesis.

Figure 3.15: “Top 5 documents closest to the word "cancer"

In conclusion, in the above example we uploaded a file with 35,000 mutations to the graph. Using a tf-idf filter, 327 of them were returned. These 327 mutations are contained in 692 texts which were loaded as input to Top2Vec. Top2vec identified 6 topics. Selecting the topic that is related to schizophrenia, the most relevant articles of this topic returned along with a cosine metric that states how relevant the article is to the topic of the cluster. These documents contain information about some of the mutations that exist in both the graph and the file with the 35.000 mutations. We then searched the entire set of documents (692 documents) for content semantically similar to the word "cancer". As shown in Figure 24, the articles that are returned, study the association of some mutations with cancer risk.

Top2Vec enables the user to understand how close the above topics are to each other through UMAP plots. In figure 3.16 the nodes represent the articles and their colors represent the different clusters (topics). The figure also represents the connectivity between the nodes.

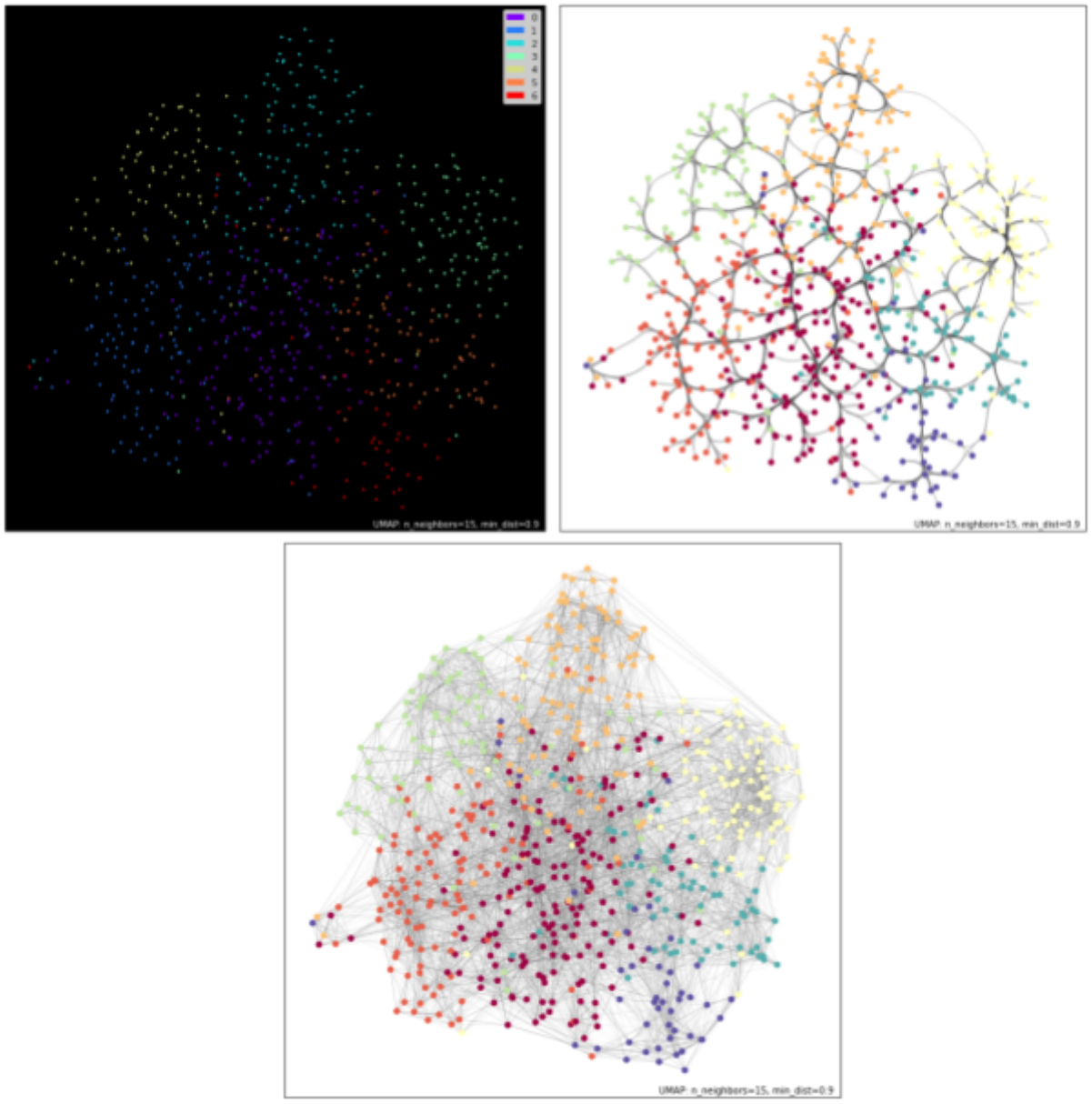

Figure 3.16: "UMAP plots"

Although the above graphs (Figure 3.17) may look impressive, especially in cases where there is a large number of nodes and topics, they do not help the user to extract detailed information from them. For this reason we made a python script on a jupyter (Kluyver et al. 2016) notebook which produces interactive UMAP plots. As shown in Figure 3.18the user is able to hover over the data points and get more information about the nodes of the graph.

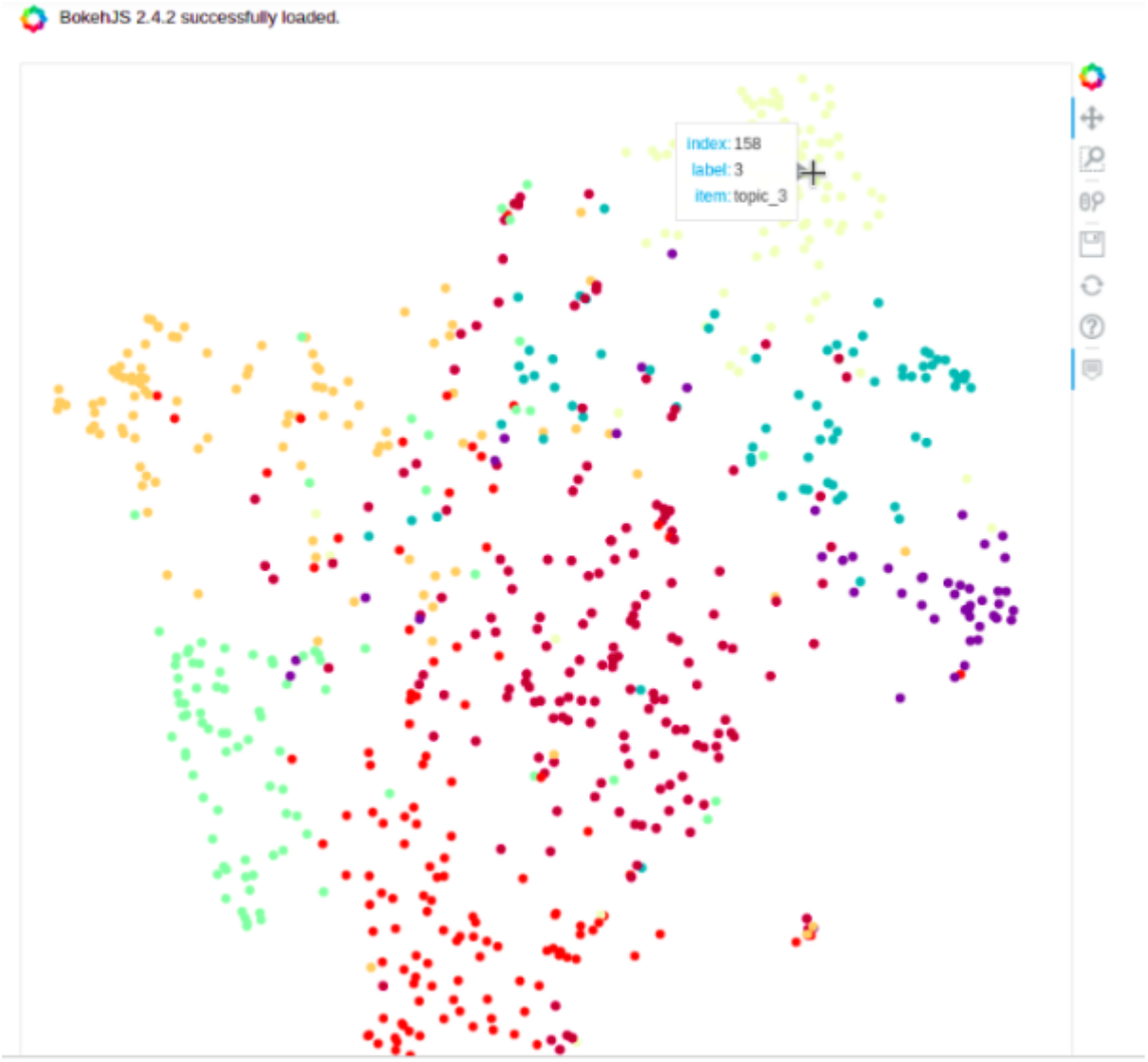

Figure 3.17: "Interactive UMAP plot"

3.4 Validation with PRS

3.4.1 Polygenic risk score

The genetic basis of common diseases is complex since the biologic factors regulating most of these diseases are polygenic (Iles 2008). For the development of complex diseases there are a huge number of genetic variants that have been associated, most with very small effects (Momozawa and Mizukami 2021). Nevertheless the cumulative effect of these variants turns out to be substantial both for the prognosis and characterization of these diseases. This cumulative score from a set of disease associated variants is called Polygenic

Risk Score (PRS) (Khera et al. 2018). PRS is calculated as the weighted sum of risk alleles with the weights specified by genome-wide association studies (GWAS). The need for an open resource for research and validation of polygenic scores led to the creation of the PSG Catalog (Lambert et al. 2021). All published PGS and related metadata are available in PSG Catalog site[16] along with the disease/phenotype that they characterize.

3.4.2 Validation

PRS is a valuable resource for the validation of our pipeline. PRS are commonly applied in exome sequencing in order to generate a predisposition score for a certain common disease. Therefore it is expected that an individual with a genetic common disease should have a subset of the variants that are present in a relevant PRS. Similarly, in our pipeline we expect that a subset of the ~40.000 variants identified from the exome sequencing screening of an individual, to be present in our graph. Querying the graph with the complete set of variants for a given PRS, extracting the resulting text from papers and applying Top2Vec, should reveal topics that are relevant to the disease. For example, giving as input to our graph a file with mutations that have effect for Alzheimer disease, topic modeling should find a topic related to Alzheimer disease. Interestingly our process disregards the effect information that is present in PRS. Therefore our process is different from the typical PRS pipeline which assigns a score for a given common disease. In general our pipeline answers to “which” common disease an individual has predisposition on, whereas PRS answers to “how much” predisposition exists.

To validate our approach, we downloaded two variant collections from the FTP site of PGS, one for osteoporosis (Tanigawa et al. 2022) and one for thrombosis (Tanigawa et al. 2022). For each file we retrieved the rs-ids in order to mass query the graph.

Concerning the thrombosis file, there were 634 rs-ids and the corresponding effect weights. We removed the effect weights and the resulting file was sent to the graph as a query. In total 27 mutations were returned which were contained in 72 articles. Then these articles were given to Top2Vec for topic modeling and two topics were returned (figure 3.19).

As the words clouds of figure 3.19 indicate, topic 0 concerns osteoporosis and topic 1 concerns fasting and lipid metabolism. This finding poses the question of whether there is a correlation between these two conditions. A simple search in PubMed revealed articles that relate osteoporosis and fasting (Hisatomi and Kugino 2019; Barnosky et al. 2017; Veronese and Reginster 2019) and articles that relate osteoporosis and lipid metabolism (Chen et al. 2017; Tian and Yu 2015; B. Wang et al. 2022).

In brief, osteoarthritis and fasting are correlated since weight loss leads to improved bone health and consequently in battling age related osteoporosis.

[16] thttps://www.pgscatalog.org/

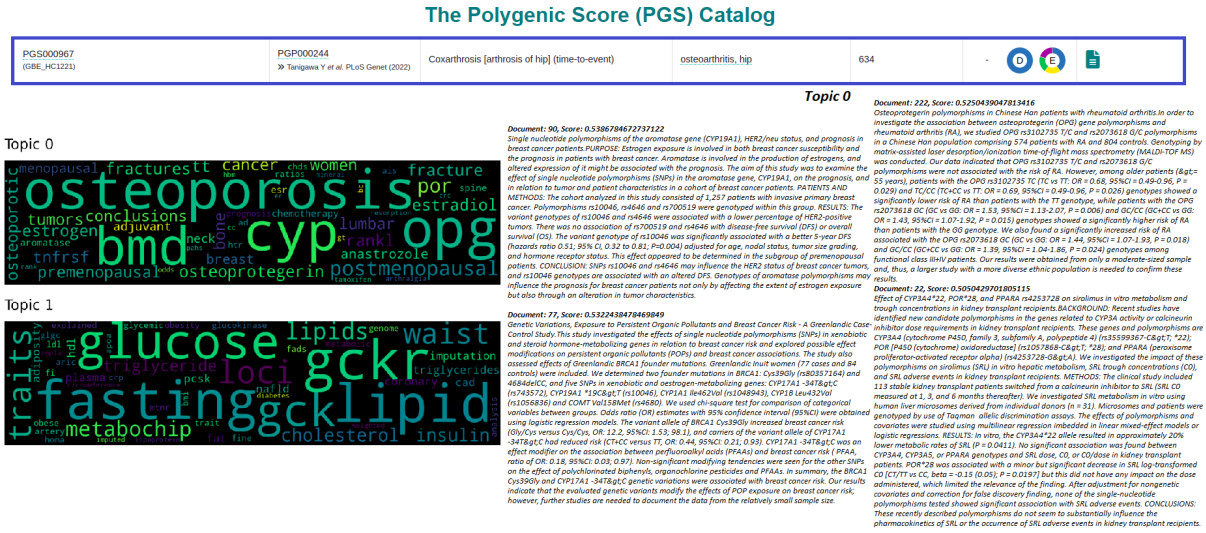

Figure 3.18: "Results of topic modeling of a PGS Catalogue file with 634 mutations associated with osteoarthritis"

Concerning the thrombosis PRS, there were 839 corresponding mutations which were sent to the graph as a query and 81 mutations of them were returned. These mutations were contained in 425 articles which were given for topic modeling. As shown in words clouds of figure 3.20, there are three topics, topic 0 concerns common genomic concepts without any clinical information, topic 1 concerns thrombosis and topic 2 is about schizophrenia. Topic 1 is what we expected and confirms that our tool works properly. The words of topic 2 are indicative of an interesting potential association between thrombosis and schizophrenia.

There are several articles that search for possible correlation between schizophrenia and thrombosis (Hsu et al. 2015); (Lin et al. 2019; De Hert et al. 2010). Some articles refer that antipsychotic drug use increases this risk of thrombosis. Thus, topic modeling led us to the conclusion that there is a possible association between schizophrenia and thrombosis.

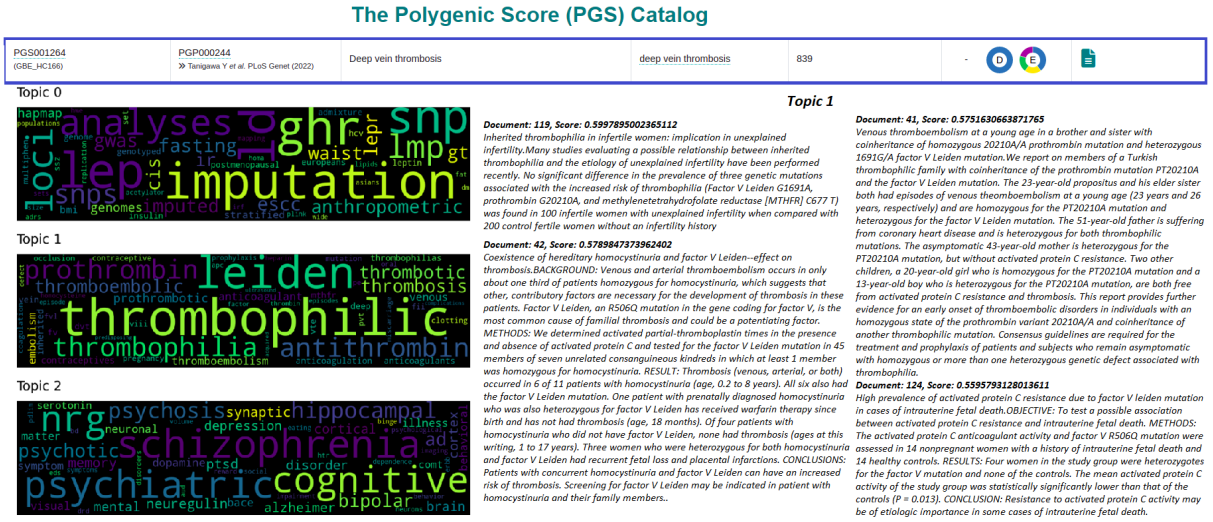

Figure 3.19: "Results of topic modeling of a PGS Catalogue file with 839 mutations associated with thrombosis"

In conclusion, through our pipeline, new relationships can be explored between a huge set of mutations and possible diseases / phenotypes. The topics that are identified that are not the target ones might provide insights about novel relationships between diseases.

4. Discussion

Next generation sequencing (NGS) tends to become a routine test in clinical practice through the rise of its accessibility, the increase of its trustworthiness and the reduction of its cost. .One last criterion that is still missing from NGS before “reaching the clinic” is interpretability. Findings from NGS should easily and prominently stick out so that clinicians and medical professionals could make a reliable informed decision. Despite the great progress that has been made in the field of variant prioritization, Geneticists have not benefited much from the information contained in PubMed regarding genetic entities such as genes and mutations.

Both the immense size and the exponential growth of the information contained within PubMed makes it difficult to exploit this knowledge. Although many tools have been suggested for the mass exploration of this knowledge base, most are based on pre-defining a minimum set of keywords and relationships in which the exploration will be based. In the era of Next Generation Sequencing and in particular in exome sequencing, we cannot define such a set. Or else we cannot simply search the entirety of PubMed about the complete set of mutations that have been identified. Interestingly the problem is not about how to perform the query. Existing query techniques and database designs can execute complex queries in massive databases in a timely manner. The problem lies in interpreting and visually inspecting the results.

The purpose of this master thesis is to integrate text mining into the evaluation of a set of mutations. We implemented a pipeline to locate biomedical entities (mutations, genes, diseases, chemicals) in Pubmed and import them into a graph database. Through the graph a user has the ability to send multiple queries to easily and flexibly search for information both for individual entities and for a set of mutations.

Variant nomenclature is an issue that has been of considerable concern to the scientific community (Poo, Cai, and Mah 2011). Our initial goal was to convert as many mutations as possible into hgvs format. This attempt failed because we created all possible HGVS mutations in the articles and the estimated time for their validation was too long. Nevertheless the validation of HGVS mutations in a short period of time is still a challenging task. The enrichment of the graph we created with the HGVS mutations would be a significant addition that belongs to our immediate future work

After finally using SNP mutations (RS-id), we constructed the graph with the biomedical entities of the articles. When a set of mutations is given as a query in the graph, the entities are mere words and can not help to draw safe and clear conclusions. A solution for rapidly

accessing the text of an article would be to store the contents of the papers in the database. This would practically create a "clone" of PubMed and it would require an excessive amount of disk space. Given the allocated resources to this project this was not possible, but this option should remain under consideration given sufficient amounts of resources.
To solve this issue, we used the Random File Access (Peterson 1957) method, so that we have very fast access to the texts of the articles. These texts were used for topic modeling with Top2Vec so that the different topics of a set of articles could be distinguished.

To further validate our approach we used sets of disease associated variants from the PRS Catalogue. The hypothesis was that the resulting topics from Top2Vec should include topics related to the given disease. Experiments with both osteoporosis and thrombosis (section 3.4.1) validated this hypothesis. In addition, we also showed that topic modeling generates additional topics, allowing the research to explore existing or potential novel interesting relationships between a condition of interest and other biomedical concepts. In contrast to other text mining tools that explore relationships through semantically proximal concepts (i.e. diseases with diseases or genes with genes), topic modeling generates clouds with words that transcend the semantic space. Although this might be vague, it can also help researchers to perform a broader investigation beyond the definition of certains concepts.

It is important to note that a definite validation of this pipeline is missing since we do not have in our disposal a comprehensive set of exome sequencing data from individuals with known diseases, rare or common. As a future work we intend to test this pipeline either in simulated exome sequencing data or in real data that are part of a research initiative. This could also help to fine-tune our pipeline, especially the Top2Vec part which we might be prone to noise.

This work is intended to be used by people who are not necessarily familiar with the command line or Cypher language. For this reason, another future work is the construction of a webpage that could ease the data access, the graph exploration and the extraction of topic models.

Finally, the complete pipeline has been made available as an Argo workflow. This workflow can be installed and deployed with minimal effort. Additionally it can be easily scaled in a Kubernetes cluster and meet the needs of a more demanding setup. Describing and providing complex bioinformatics pipelines in widely accepted formats and easily deployable workflow management systems such as Argo can not be very convenient but also help to battle the reproducibility crisis which, especially in a clinical genetics setting, could have detrimental effects.

References

Adzhubei, Ivan, Daniel M. Jordan, and Shamil R. Sunyaev. 2013. “Predicting Functional Effect of Human Missense Mutations Using PolyPhen-2.” *Current Protocols in Human Genetics / Editorial Board, Jonathan L. Haines ... [et Al.]* Chapter 7 (January): Unit7.20.

Antonarakis, Stylianos E., and Nomenclature Working Group. 1998. “Recommendations for a Nomenclature System for Human Gene Mutations.” *Human Mutation*. https://doi.org/10.1002/(sici)1098-1004(1998)11:1<1::aid-humu1>3.0.co;2-o.

Baker, Monya. 2016. “1,500 Scientists Lift the Lid on Reproducibility.” *Nature* 533 (7604): 452–54.

Barnosky, Adrienne, Cynthia M. Kroeger, John F. Trepanowski, Monica C. Klempel, Surabhi Bhutani, Kristin K. Hoddy, Kelsey Gabel, Sue A. Shapses, and Krista A. Varady. 2017. “Effect of Alternate Day Fasting on Markers of Bone Metabolism: An Exploratory Analysis of a 6-Month Randomized Controlled Trial.” *Nutrition and Healthy Aging* 4 (3): 255–63.

Belkadi, Aziz, Alexandre Bolze, Yuval Itan, Aurélie Cobat, Quentin B. Vincent, Alexander Antipenko, Lei Shang, Bertrand Boisson, Jean-Laurent Casanova, and Laurent Abel. 2015. “Whole-Genome Sequencing Is More Powerful than Whole-Exome Sequencing for Detecting Exome Variants.” *Proceedings of the National Academy of Sciences of the United States of America* 112 (17): 5473–78.

Birgmeier, Johannes, Andrew P. Tierno, Peter D. Stenson, Cole A. Deisseroth, Karthik A. Jagadeesh, David N. Cooper, Jonathan A. Bernstein, Maximilian Haeussler, and Gill Bejerano. n.d. “AVADA Enables Automated Genetic Variant Curation Directly from the Full Text Literature.” https://doi.org/10.1101/461269.

Boudellioua, Imane, Maxat Kulmanov, Paul N. Schofield, Georgios V. Gkoutos, and Robert Hoehndorf. 2019. “DeepPVP: Phenotype-Based Prioritization of Causative Variants Using Deep Learning.” *BMC Bioinformatics* 20 (1): 65.

Caporaso, J. Gregory, William A. Baumgartner Jr, David A. Randolph, K. Bretonnel Cohen, and Lawrence Hunter. 2007. “MutationFinder: A High-Performance System for Extracting Point Mutation Mentions from Text.” *Bioinformatics* 23 (14): 1862–65.

Chen, Z., G-H Zhao, Y-K Zhang, G-S Shen, Y-J Xu, and N-W Xu. 2017. “Research on the Correlation of Diabetes Mellitus Complicated with Osteoporosis with Lipid Metabolism, Adipokines and Inflammatory Factors and Its Regression Analysis.” *European Review for Medical and Pharmacological Sciences* 21 (17): 3900–3905.

Choi, Murim, Ute I. Scholl, Weizhen Ji, Tiewen Liu, Irina R. Tikhonova, Paul Zumbo, Ahmet Nayir, et al. 2009. “Genetic Diagnosis by Whole Exome Capture and Massively Parallel DNA Sequencing.” *Proceedings of the National Academy of Sciences*. https://doi.org/10.1073/pnas.0910672106.

Consortium, †the International Hapmap, and †The International HapMap Consortium. 2003. “The International HapMap Project.” *Nature*. https://doi.org/10.1038/nature02168.

Cornish, Adam, and Chittibabu Guda. 2015. “A Comparison of Variant Calling Pipelines Using Genome in a Bottle as a Reference.” *BioMed Research International*. https://doi.org/10.1155/2015/456479.

D’Antonio, Mattia, Paolo D’onorio De Meo, Daniele Paoletti, Berardino Elmi, Matteo Pallocca, Nico Sanna, Ernesto Picardi, Graziano Pesole, and Tiziana Castrignanò. 2013. “WEP: A High-Performance Analysis Pipeline for Whole-Exome Data.” *BMC Bioinformatics*. https://doi.org/10.1186/1471-2105-14-s7-s11.

De Hert, M., G. Einfinger, E. Scherpenberg, M. Wampers, and J. Peuskens. 2010. “The Prevention of Deep Venous Thrombosis in Physically Restrained Patients with Schizophrenia.” *International Journal of Clinical Practice* 64 (8). https://doi.org/10.1111/j.1742-1241.2010.02380.x.

Demner-Fushman, Dina, Willie J. Rogers, and Alan R. Aronson. 2017. “MetaMap Lite: An Evaluation of a New Java Implementation of MetaMap.” *Journal of the American Medical Informatics Association: JAMIA* 24 (4): 841–44.

Di Tommaso, Paolo, Maria Chatzou, Evan W. Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. 2017. “Nextflow Enables Reproducible Computational Workflows.” *Nature Biotechnology* 35 (4): 316–19.

Doughty, Emily, Attila Kertesz-Farkas, Olivier Bodenreider, Gary Thompson, Asa Adadey, Thomas Peterson, and Maricel G. Kann. 2011. “Toward an Automatic Method for Extracting Cancer- and Other Disease-Related Point Mutations from the Biomedical Literature.” *Bioinformatics* 27 (3): 408–15.

Dumais, Susan T. 2005. “Latent Semantic Analysis.” *Annual Review of Information Science and Technology* 38 (1): 188–230.

Dunnen, Johan T. den, Raymond Dalgleish, Donna R. Maglott, Reece K. Hart, Marc S. Greenblatt, Jean McGowan-Jordan, Anne-Francoise Roux, Timothy Smith, Stylianos E. Antonarakis, and Peter E. M. Taschner. 2016. “HGVS Recommendations for the Description of Sequence Variants: 2016 Update.” *Human Mutation* 37 (6): 564–69.

Fernandes, Diogo, and Jorge Bernardino. 2018. “Graph Databases Comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB.” *Proceedings of the 7th International Conference on Data Science, Technology and Applications*. https://doi.org/10.5220/0006910203730380.

Fischer, Maria, Rene Snajder, Stephan Pabinger, Andreas Dander, Anna Schossig, Johannes Zschocke, Zlatko Trajanoski, and Gernot Stocker. 2012. “SIMPLEX: Cloud-Enabled Pipeline for the Comprehensive Analysis of Exome Sequencing Data.” *PLoS ONE*. https://doi.org/10.1371/journal.pone.0041948.

Flygare, Steven, Edgar Javier Hernandez, Lon Phan, Barry Moore, Man Li, Anthony Fejes, Hao Hu, et al. 2018. “The VAAST Variant Prioritizer (VVP): Ultrafast, Easy to Use Whole Genome Variant Prioritization Tool.” *BMC Bioinformatics*. https://doi.org/10.1186/s12859-018-2056-y.

Freeman, Peter J., Reece K. Hart, Liam J. Gretton, Anthony J. Brookes, and Raymond Dalgleish. 2018. “VariantValidator: Accurate Validation, Mapping, and Formatting of Sequence Variation Descriptions.” *Human Mutation* 39 (1): 61–68.

Garcia, Maxime, Szilveszter Juhos, Malin Larsson, Pall I. Olason, Marcel Martin, Jesper Eisfeldt, Sebastian DiLorenzo, et al. 2020. “Sarek: A Portable Workflow for Whole-Genome Sequencing Analysis of Germline and Somatic Variants.” *F1000Research* 9 (January): 63.

“Gene Ontology (GO).” n.d. *SpringerReference*. https://doi.org/10.1007/springerreference_96994.

Griffith, Malachi, and Obi L. Griffith. 2004. “HGMD (Human Gene Mutation Database).” *Dictionary of Bioinformatics and Computational Biology*. https://doi.org/10.1002/9780471650126.dob0942.

Guo, Yunfei, Xiaolei Ding, Yufeng Shen, Gholson J. Lyon, and Kai Wang. 2015. “SeqMule: Automated Pipeline for Analysis of Human Exome/genome Sequencing Data.” *Scientific Reports* 5 (September): 14283.

Hamosh, Ada, Alan F. Scott, Joanna Amberger, David Valle, and Victor A. McKusick. 2000. “Online Mendelian Inheritance In Man (OMIM).” *Human Mutation*. https://doi.org/10.1002/(sici)1098-1004(200001)15:1<57::aid-humu12>3.0.co;2-g.

Hirakawa, Mika, Toshihiro Tanaka, Yoichi Hashimoto, Masako Kuroda, Toshihisa Takagi, and Yusuke Nakamura. 2002. “JSNP: A Database of Common Gene Variations in the Japanese Population.” *Nucleic Acids Research* 30 (1): 158–62.

Hisatomi, Yuko, and Kenji Kugino. 2019. “Changes in Bone Density and Bone Quality Caused by Single Fasting for 96 Hours in Rats.” *PeerJ* 6. https://doi.org/10.7717/peerj.6161.

Hofmann, Thomas. 2017. “Probabilistic Latent Semantic Indexing.” *ACM SIGIR Forum*. https://doi.org/10.1145/3130348.3130370.

Hsu, W. Y., H. Y. Lane, C. L. Lin, and C. H. Kao. 2015. “A Population-Based Cohort Study on Deep Vein Thrombosis and Pulmonary Embolism among Schizophrenia Patients.” *Schizophrenia Research* 162 (1-3). https://doi.org/10.1016/j.schres.2015.01.012.

Iles, Mark M. 2008. “What Can Genome-Wide Association Studies Tell Us about the Genetics of Common Disease?” *PLoS Genetics* 4 (2): e33.

Jain, Aditi, and Raj Kumari. 2017. “A Review on Comparison of Workflow Scheduling Algorithms with Scientific Workflows.” In *Advances in Intelligent Systems and Computing*, 613–22. Advances in Intelligent Systems and Computing. Singapore: Springer Singapore.

Khera, Amit V., Mark Chaffin, Krishna G. Aragam, Mary E. Haas, Carolina Roselli, Seung Hoan Choi, Pradeep Natarajan, et al. 2018. “Genome-Wide Polygenic Scores for Common Diseases Identify Individuals with Risk Equivalent to Monogenic Mutations.” *Nature Genetics* 50 (9): 1219–24.

Köhler, Sebastian, Nicole A. Vasilevsky, Mark Engelstad, Erin Foster, Julie McMurry, Ségolène Aymé, Gareth Baynam, et al. 2017. “The Human Phenotype Ontology in 2017.” *Nucleic Acids Research* 45 (D1): D865–76.

Köster, Johannes, and Sven Rahmann. 2018. “Snakemake—a Scalable Bioinformatics Workflow Engine.” *Bioinformatics*. https://doi.org/10.1093/bioinformatics/bty350.

Kulkarni, Neha, Luca Alessandrì, Riccardo Panero, Maddalena Arigoni, Martina Olivero, Giulio Ferrero, Francesca Cordero, Marco Beccuti, and Raffaele A. Calogero. 2018. “Reproducible Bioinformatics Project: A Community for Reproducible Bioinformatics Analysis Pipelines.” *BMC Bioinformatics*. https://doi.org/10.1186/s12859-018-2296-x.

Lambert, Samuel A., Laurent Gil, Simon Jupp, Scott C. Ritchie, Yu Xu, Annalisa Buniello, Aoife McMahon, et al. 2021. “The Polygenic Score Catalog as an Open Database for Reproducibility and Systematic Evaluation.” *Nature Genetics* 53 (4): 420–25.

Landrum, Melissa J., and Brandi L. Kattman. 2018. “ClinVar at Five Years: Delivering on the Promise.” *Human Mutation* 39 (11): 1623–30.

Leaman, Robert, and Zhiyong Lu. 2014. “Disease Named Entity Recognition and Normalization with DNorm.” *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*. https://doi.org/10.1145/2649387.2660780.

———. 2016. “TaggerOne: Joint Named Entity Recognition and Normalization with Semi-Markov Models.” *Bioinformatics* 32 (18): 2839–46.

Lee, Kyubum, Chih-Hsuan Wei, and Zhiyong Lu. 2021. “Recent Advances of Automated Methods for Searching and Extracting Genomic Variant Information from Biomedical Literature.” *Briefings in Bioinformatics*. https://doi.org/10.1093/bib/bbaa142.

Lefter, Mihai, Jonathan K. Vis, Martijn Vermaat, Johan T. den Dunnen, Peter E. M. Taschner, and Jeroen F. J. Laros. n.d. “Mutalyzer 2: Next Generation HGVS Nomenclature Checker.” https://doi.org/10.1101/2020.06.24.168583.

Ligozat, Anne-Laure, Aurélie Névéol, Bénédicte Daly, and Emmanuelle Frenoux. 2020. “Ten Simple Rules to Make Your Research More Sustainable.” *PLoS Computational Biology* 16 (9): e1008148.

Lin, C. E., C. H. Chung, L. F. Chen, and W. C. Chien. 2019. “Increased Risk for Venous Thromboembolism among Patients with Concurrent Depressive, Bipolar, and Schizophrenic Disorders.” *General Hospital Psychiatry* 61. https://doi.org/10.1016/j.genhosppsych.2019.10.003.

Linderman, Michael D., Tracy Brandt, Lisa Edelmann, Omar Jabado, Yumi Kasai, Ruth Kornreich, Milind Mahajan, Hardik Shah, Andrew Kasarskis, and Eric E. Schadt. 2014. “Analytical Validation of Whole Exome and Whole Genome Sequencing for Clinical Applications.” *BMC Medical Genomics*. https://doi.org/10.1186/1755-8794-7-20.

Luo, Ling, Shankai Yan, Po-Ting Lai, Daniel Veltri, Andrew Oler, Sandhya Xirasagar, Rajarshi Ghosh, Morgan Similuk, Peter N. Robinson, and Zhiyong Lu. 2021. “PhenoTagger: A Hybrid Method for Phenotype Concept Recognition Using Human Phenotype Ontology.” *Bioinformatics* , January. https://doi.org/10.1093/bioinformatics/btab019.

Mattingly, Carolyn J., Glenn T. Colby, John N. Forrest, and James L. Boyer. 2003. “The Comparative Toxicogenomics Database (CTD).” *Environmental Health Perspectives*. https://doi.org/10.1289/txg.6028.

McInnes, Leland, John Healy, and Steve Astels. 2017. “Hdbscan: Hierarchical Density Based Clustering.” *The Journal of Open Source Software*. https://doi.org/10.21105/joss.00205.

McInnes, Leland, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. “UMAP: Uniform Manifold Approximation and Projection.” *Journal of Open Source Software*. https://doi.org/10.21105/joss.00861.

Momozawa, Yukihide, and Keijiro Mizukami. 2021. “Unique Roles of Rare Variants in the Genetics of Complex Diseases in Humans.” *Journal of Human Genetics* 66 (1): 11–23.

Nagele, Peter. 2013. “Exome Sequencing: One Small Step for Malignant Hyperthermia, One Giant Step for Our Specialty--Why Exome Sequencing Matters to All of Us, Not Just the Experts.” *Anesthesiology*.

Pearson, Thomas A. 2008. “How to Interpret a Genome-Wide Association Study.” *JAMA*.

https://doi.org/10.1001/jama.299.11.1335.
Peterson, W. W. 1957. “Addressing for Random-Access Storage.” *IBM Journal of Research and Development*. https://doi.org/10.1147/rd.12.0130.
Poo, Danny C. C., Shaojiang Cai, and James T. L. Mah. 2011. “UASIS: Universal Automatic SNP Identification System.” *BMC Genomics* 12 Suppl 3 (November): S9.
Rentzsch, Philipp, Daniela Witten, Gregory M. Cooper, Jay Shendure, and Martin Kircher. 2019. “CADD: Predicting the Deleteriousness of Variants throughout the Human Genome.” *Nucleic Acids Research* 47 (D1): D886–94.
Ruark, Elise, Márton Münz, Matthew Clarke, Anthony Renwick, Emma Ramsay, Anna Elliott, Sheila Seal, Gerton Lunter, and Nazneen Rahman. 2016. “OpEx - a Validated, Automated Pipeline Optimised for Clinical Exome Sequence Analysis.” *Scientific Reports* 6 (August): 31029.
Sahatqija, Kosovare, Jaumin Ajdari, Xhemal Zenuni, Bujar Raufi, and Florije Ismaili. 2018. “Comparison between Relational and NOSQL Databases.” *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. https://doi.org/10.23919/mipro.2018.8400041.
Sandve, Geir Kjetil, Anton Nekrutenko, James Taylor, and Eivind Hovig. 2013. “Ten Simple Rules for Reproducible Computational Research.” *PLoS Computational Biology* 9 (10): e1003285.
Schriml, L. M., C. Arze, S. Nadendla, Y-W W. Chang, M. Mazaitis, V. Felix, G. Feng, and W. A. Kibbe. 2012. “Disease Ontology: A Backbone for Disease Semantic Integration.” *Nucleic Acids Research*. https://doi.org/10.1093/nar/gkr972.
Sherry, S. T., M. Ward, and K. Sirotkin. 1999. “dbSNP-Database for Single Nucleotide Polymorphisms and Other Classes of Minor Genetic Variation.” *Genome Research* 9 (8): 677–79.
Sim, Ngak-Leng, Prateek Kumar, Jing Hu, Steven Henikoff, Georg Schneider, and Pauline C. Ng. 2012. “SIFT Web Server: Predicting Effects of Amino Acid Substitutions on Proteins.” *Nucleic Acids Research* 40 (Web Server issue): W452–57.
Singer, Jochen, Hans-Joachim Ruscheweyh, Ariane L. Hofmann, Thomas Thurnherr, Franziska Singer, Nora C. Toussaint, Charlotte K. Y. Ng, et al. 2018. “NGS-Pipe: A Flexible, Easily Extendable and Highly Configurable Framework for NGS Analysis.” *Bioinformatics* 34 (1): 107–8.
Singleton, Marc V., Stephen L. Guthery, Karl V. Voelkerding, Karin Chen, Brett Kennedy, Rebecca L. Margraf, Jacob Durtschi, et al. 2014. “Phevor Combines Multiple Biomedical Ontologies for Accurate Identification of Disease-Causing Alleles in Single Individuals and Small Nuclear Families.” *American Journal of Human Genetics* 94 (4): 599–610.
Smedley, Damian, Julius O. B. Jacobsen, Marten Jäger, Sebastian Köhler, Manuel Holtgrewe, Max Schubach, Enrico Siragusa, et al. 2015. “Next-Generation Diagnostics and Disease-Gene Discovery with the Exomiser.” *Nature Protocols* 10 (12): 2004–15.
Stark, Zornitza, Melbourne Genomics Health Alliance, Harriet Dashnow, Sebastian Lunke, Tiong Y. Tan, Alison Yeung, Simon Sadedin, et al. 2017. “A Clinically Driven Variant Prioritization Framework Outperforms Purely Computational Approaches for the Diagnostic Analysis of Singleton WES Data.” *European Journal of Human Genetics*. https://doi.org/10.1038/ejhg.2017.123.
Tanigawa, Yosuke, Junyang Qian, Guhan Venkataraman, Johanne Marie Justesen, Ruilin Li, Robert Tibshirani, Trevor Hastie, and Manuel A. Rivas. 2022. “Significant Sparse Polygenic Risk Scores across 813 Traits in UK Biobank.” *PLoS Genetics* 18 (3): e1010105.
Thomas, Philippe, Tim Rocktäschel, Jörg Hakenberg, Yvonne Lichtblau, and Ulf Leser. 2016. “SETH Detects and Normalizes Genetic Variants in Text.” *Bioinformatics* 32 (18): 2883–85.
Tian, Li, and Xijie Yu. 2015. “Lipid Metabolism Disorders and Bone Dysfunction--Interrelated and Mutually Regulated (review).” *Molecular Medicine Reports* 12 (1): 783–94.
Valle, Ítalo Faria do, Ítalo Faria do Valle, Enrico Giampieri, Giorgia Simonetti, Antonella Padella, Marco Manfrini, Anna Ferrari, et al. 2016. “Optimized Pipeline of MuTect and

GATK Tools to Improve the Detection of Somatic Single Nucleotide Polymorphisms in Whole-Exome Sequencing Data." *BMC Bioinformatics*. https://doi.org/10.1186/s12859-016-1190-7.
Venter, J. C., M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, et al. 2001. "The Sequence of the Human Genome." *Science* 291 (5507): 1304–51.
Veronese, N., and J. Y. Reginster. 2019. "The Effects of Calorie Restriction, Intermittent Fasting and Vegetarian Diets on Bone Health." *Aging Clinical and Experimental Research* 31 (6). https://doi.org/10.1007/s40520-019-01174-x.
Vicknair, Chad, Michael Macias, Zhendong Zhao, Xiaofei Nan, Yixin Chen, and Dawn Wilkins. 2010. "A Comparison of a Graph Database and a Relational Database." *Proceedings of the 48th Annual Southeast Regional Conference on - ACM SE '10*. https://doi.org/10.1145/1900008.1900067.
Wang, Bo, Heng Wang, Yuancheng Li, and Lei Song. 2022. "Lipid Metabolism within the Bone Micro-Environment Is Closely Associated with Bone Metabolism in Physiological and Pathophysiological Stages." *Lipids in Health and Disease* 21 (1): 1–14.
Wang, Meng, Keith M. Callenberg, Raymond Dalgleish, Alexandre Fedtsov, Naomi K. Fox, Peter J. Freeman, Kevin B. Jacobs, et al. 2018. "Hgvs: A Python Package for Manipulating Sequence Variants Using HGVS Nomenclature: 2018 Update." *Human Mutation* 39 (12): 1803–13.
Wei, Chih-Hsuan, Bethany R. Harris, Hung-Yu Kao, and Zhiyong Lu. 2013. "tmVar: A Text Mining Approach for Extracting Sequence Variants in Biomedical Literature." *Bioinformatics* 29 (11): 1433–39.
Wei, Chih-Hsuan, Hung-Yu Kao, and Zhiyong Lu. 2015. "GNormPlus: An Integrative Approach for Tagging Genes, Gene Families, and Protein Domains." *BioMed Research International* 2015 (August): 918710.
Wei, Chih-Hsuan, Lon Phan, Juliana Feltz, Rama Maiti, Tim Hefferon, and Zhiyong Lu. 2018. "tmVar 2.0: Integrating Genomic Variant Information from Literature with dbSNP and ClinVar for Precision Medicine." *Bioinformatics* 34 (1): 80–87.
Wildeman, Martin, Ernest van Ophuizen, Johan T. den Dunnen, and Peter E. M. Taschner. 2008. "Improving Sequence Variant Descriptions in Mutation Databases and Literature Using the Mutalyzer Sequence Variation Nomenclature Checker." *Human Mutation* 29 (1): 6–13.
Yang, Hui, Peter N. Robinson, and Kai Wang. 2015. "Phenolyzer: Phenotype-Based Prioritization of Candidate Genes for Human Diseases." *Nature Methods* 12 (9): 841–43.
Zook, Justin M., Brad Chapman, Jason Wang, David Mittelman, Oliver Hofmann, Winston Hide, and Marc Salit. 2014. "Integrating Human Sequence Data Sets Provides a Resource of Benchmark SNP and Indel Genotype Calls." *Nature Biotechnology*. https://doi.org/10.1038/nbt.2835.
Zook, Justin M., Jennifer McDaniel, Hemang Parikh, Haynes Heaton, Sean A. Irvine, Len Trigg, Rebecca Truty, et al. n.d. "Reproducible Integration of Multiple Sequencing Datasets to Form High-Confidence SNP, Indel, and Reference Calls for Five Human Genome Reference Materials." https://doi.org/10.1101/281006.

Angelov, Dimo. "Top2vec: Distributed representations of topics." arXiv preprint arXiv:2008.09470 (2020).
Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." In International conference on machine learning, pp. 1188-1196. PMLR, 2014.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research 3, no. Jan (2003): 993-1022.

Cer, Daniel, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant et al. "Universal sentence encoder." arXiv preprint arXiv:1803.11175 (2018).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of

deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

Rehurek, Radim, and Petr Sojka. "Gensim–python framework for vector space modelling." NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic 3, no. 2 (2011): 2.

Kluyver, Thomas, Benjamin Ragan-Kelley, Fernando Pérez, Brian E. Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley et al. Jupyter Notebooks-a publishing format for reproducible computational workflows. Vol. 2016. 2016.

Jain, Aditi, and Raj Kumari. 2017. “A Review on Comparison of Workflow Scheduling Algorithms with Scientific Workflows.” In Proceedings of International Conference on Communication and Networks, edited by Nilesh Modi, Pramode Verma, and Bhushan Trivedi, 508:613–22. Advances in Intelligent Systems and Computing. Singapore: Springer Singapore. https://doi.org/10.1007/978-981-10-2750-5_63.

Kanterakis, Alexandros, Nikos Kanakaris, Manos Koutoulakis, Konstantina Pitianou, Nikos Karacapilidis, Lefteris Koumakis, and George Potamias. 2021. "Converting Biomedical Text Annotated Resources into FAIR Research Objects with an Open Science Platform" Applied Sciences 11, no. 20: 9648. https://doi.org/10.3390/app11209648

Nikolov, Nikolay, Yared Dejene Dessalk, Akif Quddus Khan, Ahmet Soylu, Mihhail Matskin, Amir H. Payberah, and Dumitru Roman. "Conceptualization and scalable execution of big data workflows using domain-specific languages and software containers." Internet of Things 16 (2021): 100440.