Alma Mater Studiorum · Università di Bologna

Physics and Astronomy Department
PhD Thesis in Applied Physics

Implementation and optimization of algorithms in Biomedical Big Data Analytics

Supervisor:

Prof. Daniel Remondini

Correlator:

Prof. Gastone Castellani
Prof. Armando Bazzani

Presented by:

Nico Curti

Session 2019/2020

Chapter 1

Biomedical Big Data - CHIMeRA project

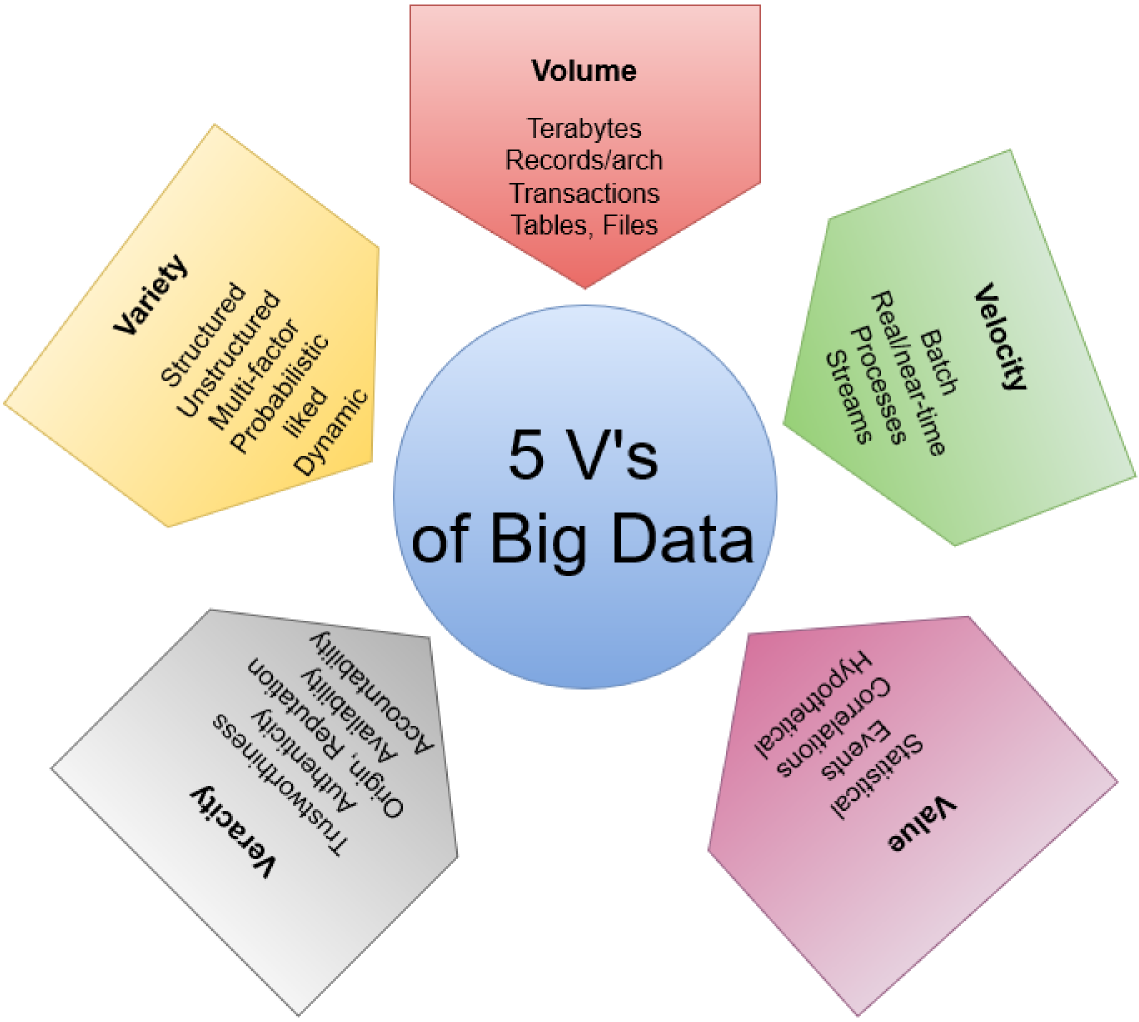


Figure 1.1: Big Data 5 V's

Every second a large quantity of data are produced and shared along the Internet and web-pages. Data are collected by social networks, chat messages, video streaming and images. Everyone, in fact, can easily create new data sources and share or put them in

Internet pages. The growth of these data is not limited to multimedia data but it involves many different fields. This is one of the most important features of the contemporary time, the so-called Big Data era: this huge volume of data has created a new field in data processing which is called Big Data Analytics that nowadays has positioned among the top ten strategic technologies (Gartner Research, 2012).

It is still difficult to provide a definition of what exactly are the Big Data and we can find many slight different nomenclatures and categories which aim to formulate it. Moreover, Big Data does not define a particular data type but more than we normally think sources can be labeled as it. The *International Journal of Computer Applications* defined them as "[$\cdots$] a collection of large and complex datasets that cannot be processed and analyzed using traditional computing techniques. It is not a single technique or a tool, rather it involves many areas of business and technology". This definition certainly involves many aspects of Big Data processing but it does not provide any definition about their nature. Moreover, it is easy to identify them as "big" and thus difficult to analyze, but they are all around us every day and just using the Internet connection every smart-phone or laptop can extract and visualize our web queries so could be not properly correct to define them in this way. However, it is sure that standard computing techniques have to be reviewed to face on this vast amount of data and a even more important attention has to be payed on the implementation of algorithms.

While a global definition of them is evidently difficult, we can provide a description of them using some of their "essential" features. One of the most common and used set of labels for this purpose is given by the so-called 5 V's of Big Data: volume, velocity, variety, veracity and value. Despite the first twos are quite obvious (the Big Data are certainly *big* in volume and they are produced very *fast*), the remaining three need a particular attention. We have already treated problems about the volume of data (ref. Chapter**??**) and the need of very fast processing and algorithmic optimizations (ref. Chapter**??**) so in this chapter we want to focus on the remaining three characteristics of Big Data Analytics.

As pre-announced there are many different sources able to provide data and this feature describes the extreme heterogeneity and variety of them. We can however broadly classify this variety into three global classes: *structured*, *semi-structured* and *unstructured* data. A dataset is *structured* if we can easily manage the information in it or, in other words, if it is described using a standard data formats and thus if it can be "quearable". On the other side we have the completely *unstructured* datasets in which data are disorganized and we need one or multiple pre-processing steps before use them or we have to develop different techniques to handle them. The intermediate format is given by the *semi-structured* data in which only a part of them could be handled with standard techniques or if they can be easily brought to their structured version. The organization of data was a crucial task in this work of thesis and we will return on this topic in the next sections.

The fourth essential characteristic of Big Data is their *veracity* due to data inconsistency and incompleteness. The data are shared very fast using Internet and we can find some ambiguities and/or deceptions between different data sources. If we want to merge and aggregate different kinds of information we have also to face this kind of problems. The final task of every Big Data Analytic application is, in fact, to process these large quantities of data and obtain a unique answer to a problem which can not vary in relation to the portion of samples or datasets used.

The last and probably most important feature is certainly their *value*: it is good to have access to large amount of data but unless we can turn them into value they are useless. In this vast amount of data only a small part of them can be considered as informative and it is always harder to extract this core of information from them. Moreover, we have to take into account also the difficulties about the management of these data and their more or less complex structure. However, also in this case, it is hard to generalize this property to all the amount of data contained in Internet: every day we see a large quantity of useless information in the Web and it is hard to figure out that some of them can be useful for research applications. A key role is played by the *questions* which we ask: for every data source there is always an appropriated question which can be answer using it and that can give a value to it and vice versa. In this way also the seemingly useless datasets can acquire importance for an appropriated research project.

In this chapter we are going to discuss about one of the latest project developed during my PhD and which is still in work in progress: the CHIMera (*Complex Human Interactions in MEdical Records and Atlases*) project. The project is founded on Big Data applications and it is accidentally born as separated branch of the INFN FiloBlu project (ref. next sections and Appendix E for further information about the FiloBlu project) which financed my last PhD year. CHIMeRA aims to create a unified database of biomedical records using Natural Language processing techniques. Its final purpose is to merge multiple data sources available on-line into a single network structure which highlights the relevant interactions between biomedical information, i.e starting from diseases to the biological agents and compounds involved into their causes and consequences. The realization of the first version of CHIMeRA required a lot of time and the development of novel pipelines of data processing. The project does not still achieve its conclusions but in this chapter we are going to cross through the main key points which allowed its construction.

1.1 The CHIMeRA project

The increasing availability of large-scale biomedical literature under the form of public on-line databases, has opened the door to a whole new understanding of multi-level associations between genomics, protein interactions and metabolic pathways for human diseases. Many structures and resources aiming at such type of analyses have been built, with the purpose of disentangling the complex relationships between various aspects of the human system relating to diseases [12, 4, 5]. Such structures, while allowing to study disease-to-some-other-omic associations, may not suffice when trying to bridge the gap of interpreting results and concept proofing clinical studies, when many types of data are involved. Looking for causation of diseases across different omics has also become a major challenge, with the aim of expanding etiology and obtaining insights on pathogenesis [6]. This task may prove to be particularly hard when dealing with medical ontology strings coming from different sources. Information of this type are usually provided by brief sentences and periphrases, while synonyms may occur to describe the same concepts thus causing different data source to provide different relationships for similar instances. Text mining and string processing, thus becomes a required step when trying to exploit medical ontology as a bridge to diffuse information.

All these data come from different kind of studies performed by independent research groups who want to prove their theory about a particular aspect of biological agent interactions. Modern biological analyses perform very capillary studies on biological compounds: in this way we can easily study the deeper relationships between them but we loose information about what there are around them. This approach is extremely efficient for the detection of the minimal causal agents of a problem but it tends to loose the global and complex[1] environment and prospective in which the process occurs. This is also the main problem of complex systems, i.e a system composed of multiple components with a mutual interactions between them. The study of an individual aspect, in fact, could give us only a partial overview of the system but we have to take into account the interactions between the multiple components for a global description.

[1] From a physical point-of-view.

The network structures are acquiring even more importance on this kind of studies. Complex System and System Biology researches have proposed multiple models about the dynamical and evolutionary interactions of the human system agents aiming to study the hidden relationships between them using network models. A network model, in fact, is able to highlight and quantify the non-trivial correlations between the system components. The mathematical definition of network structures is given by the Graph Theory and we define them as a pair $G = (V, E)$, where V is a set of elements called nodes (or vertices) and E is the set of their pairwise associations (links or edges). The total number of graph nodes (or cardinality of the graph) is denoted as N and it defines the order of the graph. The graph dimension is given by the number of its edges (m). We define a network as *complete graph* if it has all possible edges ($m = N \times N$). A network model could be described via its adjacency matrix, i.e a matrix ($N \times N$) in which each row/column identifies a node of the underlying problem and each link e_{ij} quantifies the importance of the interaction between the node v_i and the node v_j. We can define the importance of a node into the graph using the number of its connections: this is a classical *centrality measure* of a node and it is called the *degree* centrality. Starting from these definitions we can enrich our model combining multiple network structure: given two graphs $G(V, E)$ and $G'(V', E')$ we define the combination of them as a new graph in which vertices are given by the intersection of $V \cap V'$ and edges given by $E \cap E'$. If $V \cap V' = \emptyset$ the two graphs are *disjointed*; in contrary, if $V' \subseteq V$ and $E' \subseteq E$ then G' is a subgraph of G. Combining multiple graphs together we obtain a network-of-networks structure able to map a wide range of interactions from multiple sets of elements.

In real data applications we can often reasonably assume that a wide amount of the matrix entries are null, i.e the interaction between the involved agents is quite sparse, and thus we can use the important properties related to the sparse matrices to manage our network. However, when the amount of data increases also the management of a such sparse matrix could be difficult. More efficient solutions are provided by the modern Database formats and languages (e.g MySQL, SQLite, InfluxDB, $\cdots$) which store all the information into a binary format and they allow to submit queries to extract the desired portion of data. A global visualization of these huge amount of data is, in fact, without practical-sense and none valuable information can be extracted from the global representation of the system. The most important feature of network model is, in fact, the definition of a hierarchy of the interactions: the relationship between two nodes is given by the amount of connections which link them or, in other words, by their paths. Starting from a node, its nearest neighbors are given by the set of nodes connected to it: re-iterating this concept we can explore all the network structure[2]. In this way we can study the interactions of each node at different precision orders and causalities.

In light of these considerations we started to develop the CHIMeRA project (*Complex Human Interactions in MEdical Records and Atlases*) in which we aim to merge the state-of-art studies and databases about biomedical researches into a unified network structure. A key role on our network structure is played by the diseases: the major part of biomedical researches are focused on causes and consequences of a given disease and thus the corresponding databases involve the interactions between them and other biological factors. The diseases are also the most bigger manifestations of biological malfunctions and a large part of biomedical researches are financed on their study looking for their fine grain causes. Thus, a disease could be a valid “bridge” between multiple data sources: merging the disease nodes derived from different datasets we can provide a unique structure which host all the information.

The crucial point of this project was, in fact, the merging of different kind of information provided by multiple distinct data structured. As told above, the major part of researches

[2] We assume that our network structure has not isolated nodes and it has only undirected connections.

have focused on a partial aspect of the problem and they provide an independent result from the others, reducing the possibility of interactions between the outputs. Moreover, a lot of time is always spent for the creation of a practical visualization of the results using web pages and on-line services which drastically affect the real usage of these information when we want to use them in scientific research. The CHIMeRA project started from these independent sources and it aims to maximize their overlap and thus the communications between them.

We have to pay a final attention about the format of these data: in physics we are friendly with numerical data but in these context we have to work with words and text strings. The above told databases include only the research outputs and thus the interpretations of the analyzed numerical data. For example, if a numerical significant correlation was found between a disease and a gene we would find an association between them into a database (modeled as a link in our network). The only information available into this database is a link between two words, the disease name and the gene name, without any numeric value. While numbers have a unique representation (the number 42 is always 42) we can use multiple periphrases, i.e set of strings, to identify the same concept. The biomedical community, in fact, has not yet provided a unified standard for disease identification or, at least, it has not yet provided a rigid standard as for other kind of data as genes or SNPs. So, if the diseases could be an efficient way to link together multiple data sources since all the studies prove correlations between them and other biomedical information, they have the payback of an extreme variability in their nomenclature. The CHIMeRA project tried to overcome this issue using a Natural Language Processing (NLP) approach.

In the next sections we are going to discuss about the multiple steps which bring us to the formulation of our unified CHIMeRA database. We will start from the preliminary studies performed into the INFN FiloBlu project which allow the creation of the SymptomsNet structure, i.e a "smaller" network based on Italian words which connects diseases to related symptoms. Then we will briefly introduce the most common NLP techniques, also used into the CHIMeRA pipeline and finally we will show the main developed features of CHIMeRA network.

1.2 How to find the data - Web Scraping

The INFN FiloBlu project was developed by the collaboration between the Physics Department of the University of Bologna and the INFN group of the Sapienza University of Rome. The project aims to implement a NLP pipeline to process messages with medical theme using sentimental analyses. Domiciliary care for oncologic patients is preferred due to cheaper costs than hospitalization and a more comfortable living for the him. To successfully follow therapy during the domiciliary care the patient is in constant contact with health-care professionals and he is monitored frequently during his therapy. Patients are interested in actively collaborate to the management of their health and they are willing to use also ICT technologies. The FiloBlu project meets the citizens' needs developing a tool to optimize the efficiency and the effectiveness of care processes developing two APPs (patient and medical sides) to support doctor-patient communication. The final purpose of the project is to process doctor-patient chat messages (using an interface similar to "WhatsApp") computing from them a score related to the patient state. The APPs are equipped with features specifically designed for health-care applications and using a Natural Language Processing pipeline on the text messages we compute an "attention" score for each of them. The "attention" score is then used to rank the patients' messages on the medical side and thus prioritize potential critical situations.

The project was financed by Lazio region and it was developed to work at the Sant'Andrea

Hospital of Rome so the project involves only Italian words. This constrain drastically affects the data availability which are very hard to find on-line. The text messages analysis concerns the evaluation of critical keywords and medical terms so we faced on this problem generating a diseases ontology. In particular we were interesting in the relation between symptoms, diseases and their mutual interactions for the realization of our score function. More details about the pipeline used for the message processing are given in Appendix E.

The English is becoming the predominant language in the research community and it is really hard to find (enough) data in other languages: everyone who wants to share his data and information via Internet have to provide them in English if he wants to increase its visibility and availability. The Italian constrain posed by the project drastically limits the data sources and no public database was found. We would stress that as database we consider a public available set of structured data which can be downloaded and easily used.

Surfing on Internet many web pages can be found about diseases and their interactions with symptoms and causes, the so-called *on-line doctor*[3] (or Medical Services) pages. An on-line doctor is a querable Internet service which allows to provide user auto-diagnosis based on the information provided. The reliability of the information stored in these tools is only partially guaranteed by the service provider and thus it can not be considered as a scientific method for medical diagnosis. However, the amount of information collected by these applications is very interesting and it can be used to simulate reasonable medical queries, needed by our project. Also in this case, it is important to notice that, despite the availability of these public information, the data are structured according to the web page needs and moreover there is not an immediate download of the raw data.

So, how can we obtain these useful information and re-organize them into a structured data format? The answer is given by the web-scraping techniques. With the term web-scraping we identify the wide set of algorithms aim to extract the information from a website, or more in general from the Internet: while web-scraping can be done also manually, with this term we typically refer to automated processes. All the Internet pages are intrinsically pieces of codes written in different programming languages (HTML, PHP, ·). The major part of websites are written in HTML, an extreme verbose language, with more or less JavaScript supports. The way chosen to write a code and to reach the desired output is always left to the programmer: in these way we do not have a rigid standard (excepted by the programming language constrains) and under each website underlies a potential completely different ensemble of code lines. Thus, the realization of a web-scraper poses several issues to the programmer who has to find the underlying patterns inside the web page to get the information stored. In other words, the web-scraping technique is an emblematic example of Big Data Analytics algorithm since it aims to extract a *value* from a large amount of *unstructured* information (raw website code).

A web-scraping algorithm is made by a series of multiple steps which has to be performed automatically (without human overview). First of all, the algorithm has to recognized the unique website structures: we can broadly summarize this task as the parsing of the underlying HTML code. Inside the large amount of code lines[4] are stored the set of information useful for our application. So, the algorithm should be able to detect the relevant and interesting part and filter them. At this step we can easily reorganize the information into a usable data format and save them.

There are multiple way in which all these tasks could be performed and multiple open source libraries provide user friendly interfaces for the creation of own web-scraper. The most common one (and also used in our applications) is the BeautifulSoup [8] Python pack-

[3] Famous English applications are SteadyMD, MDLIVE, Sherpaa, LiveHealth Online and so on. Each service provides slight different information and the choice of the best one vary according to the user needs.

[4] Very large if we consider a pure HTML web page.

age. This package provides a very powerful Python library designed to navigate and read website source codes. The integration of this library with other pre- and post-processing techniques allow the extraction of the desired information from websites and moreover their reorganization into a structured and usable data format.

1.3 SymptomsNet

Find relationships between symptoms and diseases and their reflections on system-wise perspectives such as genomics and metabolomics still remains a crucial issue for medical research but nonetheless an open one. The relation between symptoms and diseases can be used to see analogies and co-occurrences of different pathologies, including morbidity and co-morbidity. The construction of a unique and consistent database of these kind of data is an open problem for the research and a crucial task for many actual projects. The main problems arise from the complexity and heterogeneity of the available data and from the many nomenclatures used by different public databases. In fact, in many cases it is not so clear how to infer about the association between symptoms and diseases, and, in addition, different data sources provide different associations. These information are stored as sentences and periods, of variable length and we have to face on the problem of the different synonyms and periphrases used to describe the same concept.

In our work we used large-scale public on-line databases to construct a bipartite network of human symptoms-diseases. A bipartite network (or *bigraph*) is a graph whose nodes can be divided into two disjoint and independent sets: the underlying adjacent matrix is rectangular and it describes the connections between the N elements of the first set and the M elements of the second one. We can always lead back to a square matrix $(N \cdot M \times N \cdot M)$ using zero blocks for the intra-group connections. We used common tools of natural language processing to clean and standardize the data to maximize the overlap between the different sources. After its construction, this network is used to establish a score for the different words based on the centrality measure of the node. This complex map of association can further be used to connect other data sources and enrich the diseases description from other medical points-of-view.

Many on-line databases offer auto-diagnosis tools and search engine in which the user can insert a list of symptoms or diseases and obtain the corresponding “diagnoses”. While many international databases are quite consistent and supported by medical/biological research groups, the available data in Italian language are quite scarce.

Using the Italian version of the few public *on-line doctor* websites found we obtained the needed information. We applied a set of custom web scraping pipelines to different web pages to extract the medical information and we mainly focused on sites which highlight the relationships between symptoms and diseases. We would stress that the extremely variety of the websites requires an equally varied set of web-scraping algorithms. Thus for each web page taken into account a relative web-scraper was developed. As discussed above the Italian data sources are quite fewer than the English ones so only three web pages were involved into our analysis: My PersonalTrainer[5], SaniHelp[6] and Sapere.it[7]. All these three sites provide an organized series of tables which associate a disease to the corresponding symptoms and thus are easily to treat with web-scraping algorithms. These databases are not reliable from a scientific point-of-view and their vulnerability is shown also by a non-rigid labeling of the two classes: in multiple cases we can find a disease as symptom of a different one and in many cases there is not a perfect agreement between the three data sources. Possible issues related to an incorrect disease information could

[5] Arnoldo Mondadori Editore S.p.A.

[6] Terms and conditions available here.

[7] De Agostini Group.

not be attributed to our web-scraping pipeline, but they should be already present into the original data which, we want remark it, they are the only Italian datasets public available and found.

The data extracted from the three websites cover a wide range of possible diseases and from each of them we obtained a network with a size of few thousand of nodes, our SymptomsNet. The overlap of the single words contained in the "disease-sentences" is quite low so a robust pre-processing was needed. The nodes were processed by standard natural language processing techniques and from each name the word stem was extracted to maximize the merging between the sources. If two diseases show different symptoms we decided to concatenate the list of edges to not lose information.

The processed outputs generated a network with 2 285 nodes and more than 29k links (only the 1% of the total number of possible links). The final SymptomsNet obtained by the merging of the three database sources is reported in Fig.1.2 in which the node sizes are proportional to the number of their connections (Tab.1.1 for the top ranking links).

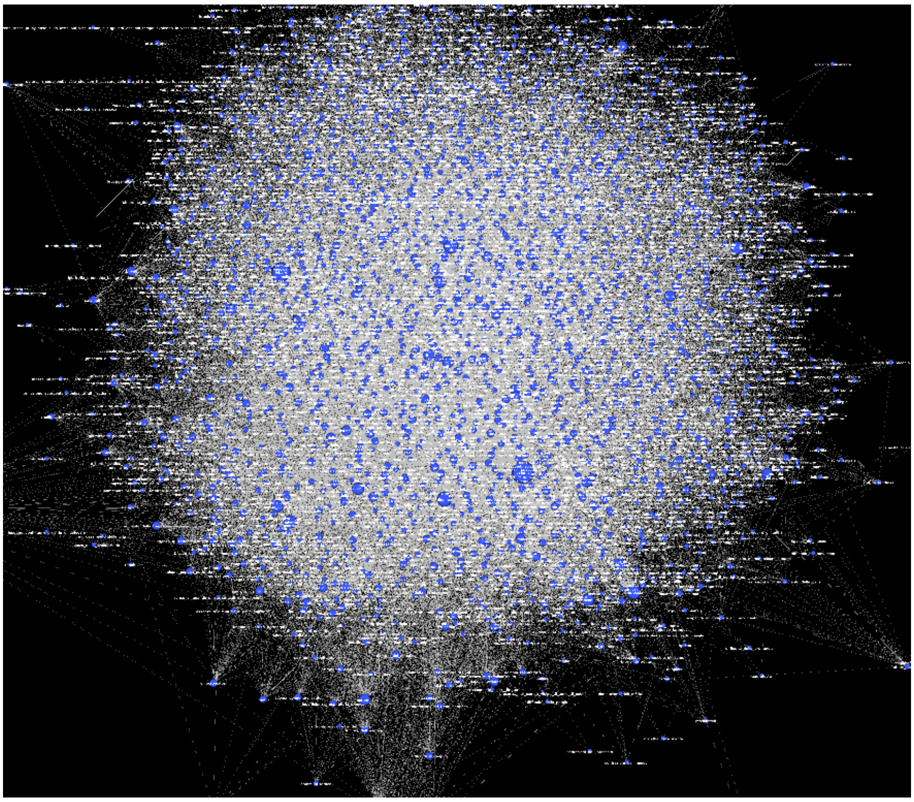


Figure 1.2: Symptoms-disease network generated by the merging of three public Italian web-pages of auto-diagnosis search engine. The network connects symptom and disease words according to validation agencies. The network comprise 2 285 nodes and more than 29k links. The node sizes are proportional to their centrality (number of connections or degree score). In this way the most common symptoms/diseases represent the biggest nodes.

In this simple example we can already notice as the most central (big) nodes are associated to the most common symptoms-diseases, as expected. This result can be already interpreted as a validation of the performed processing. We can notice from Tab.1.1 that also in the top ranking nodes we find some diseases and related synonyms: this could be an issue for the network structure since it means that the developed pipeline of processing is not able to merge together different words with equal meaning. However, the project purposes was to create a reasonably good diseases ontology and this issue could be turned

Disease/Symptom	degree
Astenia	384
Febbre	313
Dispnea	225
Nausea	222
Anoressia	201
Ematemesi	193
Vomito	182
Debolezza	176
Affaticamento	176
Esaurimento	172
Mancanza Forze	168
Edema	158

Table 1.1: Top ranking links in the SymptomsNet. We can notice "periphrases/synonyms" associated to the same symptom as *Debolezza* and *Mancanza Forze* which are left to increase the samples heterogeneity for the synthetic text generator.

to a strength of our applications since it proves that we have an agreement between the different databases (synonyms have comparable degree score and thus same importance) and it highlights the variety of mined terms (different names which identify the same disease). In fact, this kind of occurrences allow to consider a wide range of possible synonyms in the scorer attribution and so they can enforce the text analyses required by the FiloBlu project: the node degree can be used as weight (1/degree) for text words and thus we can obtain a simple score for the message given by the sum of the mapped keywords.

We conclude that from this very simple and preliminary work we are able to propose a novel symptoms-disease network based on Italian public databases and far as the author knows no other equivalent results are reported in literature. This work allowed also the realization of a novel database obtained by the union of public available data. The extracted centrality measures can be used as weights for the corresponding symptoms/diseases and a valid input to model the word frequency/importance in text analyses.

The developed network is based on a bipartite graph which associates disease nodes to symptom ones. These results highlight the potentiality of such structures and they brought us to further investigate about them and their creation. In particular, reiterating the same procedure we could be able to join together different bipartite graphs and obtain a network-of-networks structure which store multiple type of information. This is the main idea behind the CHIMeRA project. To this purpose we have to manage more reliable data sources and improve our natural language processing pipeline to increase the datasets overlap and minimize the amount of synonyms. All these tasks can be easier performed using English words and validated databases. In the next sections we are going to discuss about what natural language processing means in modern researches and we will describe the pipeline and databases used in the development of the CHIMeRA network.

1.4 Natural Language Processing

Natural Language Processing (NLP) is a quite novel research field driven by the increasing availability of textual data (ref. Fig. 1.3). As told in the previous sections the incoming of Internet world exponentially increase the amount of data shared by people

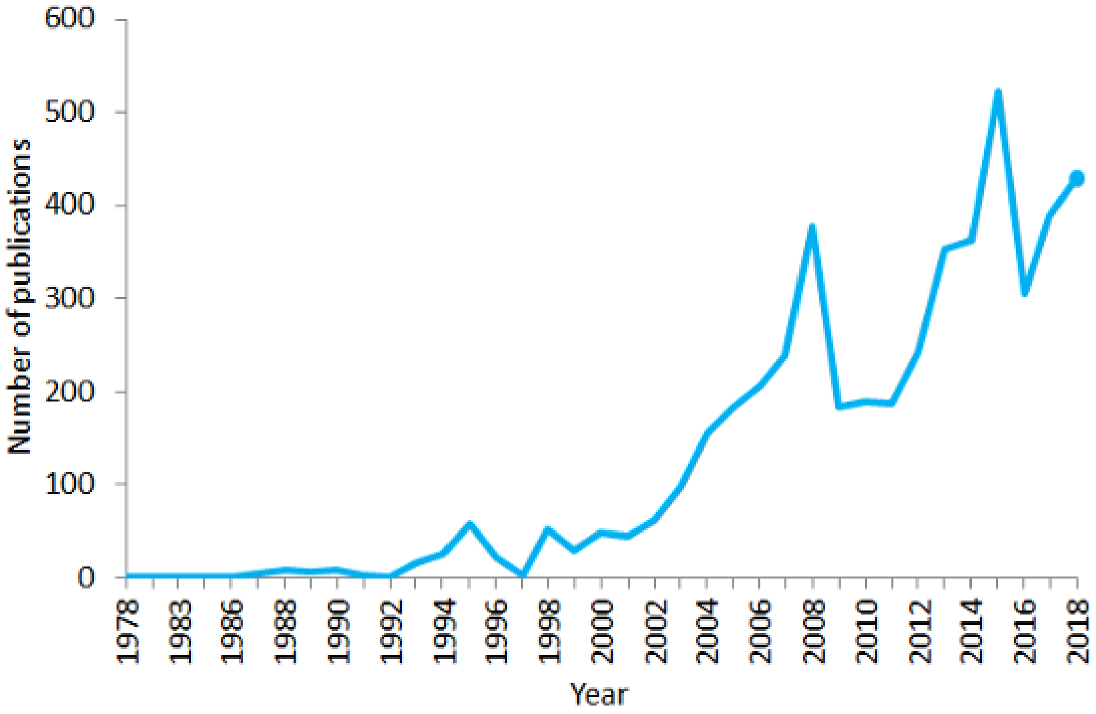


Figure 1.3: Number of publications containing the sentence "natural language processing" in PubMed in the period 1978–2018. As of 2018, PubMed comprised more than 29 million citations for biomedical literature.

and the major part of them are textual data, i.e data composed by words, phrases and more in general texts. The NLP joins together techniques coming from the linguistic, computer science, information theory and artificial intelligence researches and it concerns the interactions between human languages and computers, or in other words it studies how a computer can analyze a huge amount of natural language data and how it could extract numerical information from them. This is a very hard task to perform since it is not straightforward to teach to a machine how humans communicate between them so a key role is played by the artificial intelligence researches in the developing of new algorithmic techniques. The final purpose of the NLP is, in fact, to read, decipher, understand and make sense of the human languages extracting valuable and numerical results.

Most of the modern NLP techniques are based on a Machine Learning approach to the problem and thus we can find statistical methods against deep learning neural networks trained to face on these kind of problems. A first step has to be performed to convert the human speech into a machine readable input; then the audio signal is converted into a string text and only at the end the text can be analyzed from the machine. Applying this work-flow in forward and reverse mode we can perform a communication between a human and a machine and vice versa. In this section we will ignore how the conversion from human voice to numerical inputs could be performed and its related problems and solutions but we will focusing on the last part of this pipeline, i.e in the description of the most common techniques to process a string text into numerical values. This is also the case related to our CHIMeRA project, in which we have a huge amount of names and strings related to medical terms and we want to standardize and increase their overlap.

First of all we have to take care that each human language has its own characteristics and thus it is harder to create to a pipeline ables to process all the languages at the same time while it is easier to tune an algorithm on a particular language. In our work we were focused on the Italian language (SymptomsNet) and on the English language (CHIMeRA Network). Since the SymptomsNet project was developed as simple proof of concepts, the developed Italian pipeline was really naive and for sake of brevity we will focus only on the CHIMeRA pipeline, i.e the English one. We would stress that in our application we were not interested on the understanding of the words meaning but we want to minimize the word heterogeneity maximizing their overlap. In this way we can ignore the semantic meaning of the strings and we could focus only on their syntaxes.

The syntax is the set of rules, principles, and processes that govern the structure of sentences in a given languages. In this way we can create group of words applying grammatical rules: the grammatical rules have to be converted into algorithms which take in input a word and they give in output a processed version of the same word. In this case there is not a numerical output but just a reorganization of the string letters and words. The most common techniques involved in the syntactic analysis are:

- **Sentence breaking:** it divides a continuous text into sentences placing boundaries.
- **Word segmentation (tokenization):** it splits a large set of continuous text into units.
- **Parsing:** it provides the grammar analysis of the provided sentence.
- **Morphological segmentation:** it splits words into individual units called morphemes.
- **Part-of-speech tagging:** it identifies the grammatical part of speech for every word.
- **Lemmatization:** it reduces the inflectional forms of a word into a single form.
- **Stemming:** it cuts the inflected words to their root form.

All these algorithms are very closed each other so to better understand their functionality is useful an example. Lets start from a dummy text taken from the NLP Wikipedia web-page:

Listing 1.1: Original text

```
text = "Natural language processing (NLP) is a subfield of linguistics,
    computer science, information engineering, and artificial intelligence
    concerned with the interactions between computers and human (natural)
    languages, in particular how to program computers to process and
    analyze large amounts of natural language data. Challenges in natural
    language processing frequently involve speech recognition, natural
    language understanding, and natural language generation."
```

First of all we notice that the text is made by two sentences that can be broke using a *sentence breaking* algorithm. In this way we obtain a list of two strings given by

Listing 1.2: Sentence breaking

```
sentence_1 = "Natural language processing (NLP) is a subfield of
    linguistics, computer science, information engineering, and artificial
    intelligence concerned with the interactions between computers and
    human (natural) languages, in particular how to program computers to
    process and analyze large amounts of natural language data."

sentence_2 = "Challenges in natural language processing frequently involve
    speech recognition, natural language understanding, and natural
    language generation."
```

Now we can divide each sentence into its set of words using a word *tokenization*. Focusing only on the first sentence we obtain in output:

Listing 1.3: Tokenization

```
tokens = ['Natural', 'language', 'processing', '(', 'NLP', ')', 'is', 'a', 'subfield', 'of', 'linguistics', ',', 'computer', 'science', ',', 'information', 'engineering', ',', 'and', 'artificial', 'intelligence', 'concerned', 'with', 'the', 'interactions', 'between', 'computers', 'and', 'human', '(', 'natural', ')', 'languages', ',', 'in', 'particular', 'how', 'to', 'program', 'computers', 'to', 'process', 'and', 'analyze', 'large', 'amounts', 'of', 'natural', 'language', 'data', '.']
```

There are multiple useless tokens in the processed list and we can filter them using a type of *part-of-speech tagging* algorithm which removes the so-called *stop words* and punctuations. In our example our list of tokens become

Listing 1.4: Filtering stop-words and punctuations

```
tokens = ['Natural', 'language', 'processing', 'NLP', 'subfield', 'linguistics', 'computer', 'science', 'information', 'engineering', 'artificial', 'intelligence', 'concerned', 'interactions', 'computers','human', 'natural', 'languages', 'particular', 'program', 'computers', 'process', 'analyze', 'large', 'amounts', 'natural', 'language', 'data']
```

A final step could be given by the *stemming* algorithm which extract the root form of each word. Using the stemmer on the previous set of words we obtain

Listing 1.5: Stemming

```
tokens = ['natur', 'languag', 'process', 'nlp', 'subfield', 'linguist', 'comput', 'scienc', 'inform', 'engin', 'artifici', 'intellig', 'concern', 'interact', 'comput', 'human', 'natur', 'languag', 'particular', 'program', 'comput', 'process', 'analyz', 'larg', 'amount', 'natur', 'languag', 'data']
```

As can be seen by this example the stemming algorithm convert in lower case each letter of each words and remove the inflections from each of them. This is a very naive example but we can already notice as our processing allow to merge multiple words together. In the original sentence we have the word "*Natural*" (with capital letter) and two occurrences of "*natural*" (lower case). Moreover, we have three occurrences of the "*computer*" word but only two of them are in singular form. The tokenization + stemming processing allows to compare standardize the different word forms making them compatible.

Combinations of these algorithms can be found in everyday applications starting from email assistants or website chat box to the more advanced sentiment analyses and fake news identifiers [10, 1, 9, 13]. NLP pipelines are used also in biomedical applications and the modern multinational companies like Amazon, IBM or Google are financing different kinds of research on this topic. Amazon Comprehend Medical is a NLP service developed by Amazon to extract disease conditions, medications and treatment outcomes from patient notes, electronic health records and other clinical trial reports. At the same time also companies like Yahoo and Google based their filters and email classifiers on NLP algorithms to stop spam. Also the fake news hot topic of the these years is faced on by NLP pipeline and the NLP Group at MIT is developing new tools to determine if a source is accurate or politically biased based on analyses of texts.

In our applications we constructed a custom pipeline based on part of these algorithms. In the following sections we will describe in detail our pipeline which was tuned for our case study: we would stress that the efficiency of our pipeline could not be generalized to other datasets since our purpose was to obtain the best result for our applications. In other words, we can say that we had fine-tuned our pipeline based on the data used in this project. Moreover we have to clarify that our pipeline is not fully-automatic but it was made according to a semi-supervised approach: we customize the work-flow following the issues showed by our applications.

1.5 CHIMeRA datasets

We have seen how we can extract useful information also from unstructured databases using a web-scraping pipeline. The *on-line doctor* web pages could be very useful for a toy model application like the SymptomsNet one but if we want produce scientific relevant results we have to take care about the validity of data. Since the English datasets availability is easier than the Italian one we moved to more “robust” databases.

As told in the previous sections, there are a lot of studies performed on the disease associations to other biological compounds and in many cases the resulting datasets are public available on Internet. This is the case of the DisGeNET [2] or DrugBank [7] datasets which contain the relationship between a large number of diseases with genes/variants (SNPs) and drugs (and other information), respectively. The DisGenet web-page allows to download the datasets already stored into a well structured network format (sparse adjacency matrix, with 210 498 associations between 117 337 SNPs, 10 358 diseases and 17 549 genes) while the DrugBank poses more issues to the treatment of data: the DrugBank database was designed to provide a large set of information related to each drug using its own website and thus it needs a huge pre-processing of the JSON dataset structure to highlight all the possible network associations (14 812 drugs, 649 metabolite pathways, 3 256 gene targets, 40 SNP targets, and 532 food interactions). Using the DisGenet we can connect the diseases to their related genes and variants. From the reviewed format of the DrugBank, instead, we can link each disease to the associated drugs. Associated to each drug we have also a list of gene and SNP targets which can be merged to the information provided by DisGenet. Moreover, we can insert also food interactions, metabolite pathways and drug interactions (synergy or not) extracted from DrugBank. We would stress that, despite the trivial overlaps between the same data sources (genes, diseases and SNPs up to now), just using the rearrangement of these pair of datasets into a network structure, we can already provide a possible extrapolation of the underlying information using the paths between nodes. Starting from a disease inside the DisGenet, using a single-database approach we can study the “causality” relationships with the connected genes or SNPs. Using a multiple-databases (or a network-of-networks structure) approach we can map that disease to other kinds of information like drugs, foods and metabolite pathways. The aim of such a network-of-networks structure is to unveil relationships hidden by the underwhelming overlap between single-type information across different databases. The set of different information merged can thus be exploited for applications such as wide-scale drug effect evaluation and design, addressing general diagnostic questions for systems medicine and diseases etiology expansion. In other words, a network-of-networks structure allows the inference of missing connections using node contraction. A full list of the information collected by our web-scraping and rearrangement pipelines is shown in Tab. 1.2.

To enlarge our disease information we looked for other on-line data sources. A very interesting database is given by HMDB [11] (*Human Metabolite Data Bank*) which comprises a vast amount of metabolites and metabolite-pathways with the associated drugs and diseases (114 003 metabolite entries with chemical taxonomies and ~25 000 human metabolic and disease pathways[8]). The interconnections with the previous discussed datasets are straightforward but in this case the data are not public available and thus we had to apply

[8] The human metabolite-pathways can be divided into different types according to the informations stored in the HMDB dataset. The interactions between HMDB and DrugBank was already established through a vast series of hyper-links which connect them using metabolites and metabolite-pathways information. In this way we mapped also the information related to the metabolite-pathways types to the DrugBank dataset, obtaining a finer grain nomenclature and classification of these data. These informa-

	disease	drug	food	gene	metabolite	phenotype	SNP	metabolic pathway
disease	CTD RXList SNAP	RXList	x	DisGeNET	HMDB	CTD	DisGeNET	HMDB
drug	RXList	DrugBank	DrugBank	x	x	x	x	DrugBank
food	x	DrugBank	x	x	x	x	x	x
gene	DisGeNET	x	x	x	x	x	x	x
metabolite	HMDB	x	x	x	x	x	x	HMDB
phenotype	CTD	x	x	x	x	x	x	x
SNP	DisGeNET	x	x	x	x	x	x	x
metabolic pathway	HMDB	DrugBank	x	x	HMDB	x	x	x
# nodes	63974	35161	532	18799	114100	13214	117337	1329

Table 1.2: Description of the data mined by the CHIMeRA project before merging. The datasets were collected using custom web-scraping pipelines and by a rearrangement of the public data. For each pair of data types we report the list of datasets used to evaluate the interaction.

a web-scraping algorithm to get its information. An analogous procedure was applied to extract the data included into the RXList database. RXList is an on-line website very similar to the previous discussed auto-diagnosis tools in which we can find associations between diseases and drugs and other several pathogenic associations. In this case we have a further distinction between diseases: we have diseases related to drugs and diseases connected to other caused-diseases. We can take care of this kind association using directional links[9]. We remark that each web-scraping pipeline is customized according to a precise website, so for each analyzed case a different code was developed to address the data extraction.

All these information can enrich our database and the description of a given disease but we have to face on the problem of data merging. As previously discussed we do not have a unique nomenclature for diseases and thus we can find analogous names (periphrases or synonyms) which identify the same concept (disease). A useful tool to overcome these issues could be given by a synonym dictionary: a powerful example is given by the CTD [3] (*Comparative Toxicogenomics Database*) (7 212 diseases with mapped synonyms and 4 340 diseases with related phenotypes). Using the CTD jointly with the SNAP [14] (*Stanford Large Network Dataset Collection*, 8 803 disease terms with related synonyms) database we could enlarge the number of synonyms associated to each disease name.

We remember that the crucial point of our merging procedure is given by the disease nodes since they are the node type shared along (almost) all the databases. The help given by synonym dictionaries certainly increases the overlap between the mined datasets but we chose to maximize it using a pre-processing NLP pipeline. So, we started our pipeline using a word *standardization*, i.e converting all the words into their lower case formats and replacing all the punctuation characters with a unique one[10]. Then, we noticed that a not

tions can be used to improve our disease information. In Tab.1.2 is shown only the aggregated data.

[9] For sake of clarity, we encountered the same issue also into the DrugBank dataset in which we had intra-drug connections.

[10] An unexpected issue arise in this step: different databases use different enumeration system. In some entries we found disease names associated to numbers which identify their multiple types. An example could be "Polyendocrine Autoimmune Syndrome type 1" but at the same time in a second database the same disease could be represented by "polyendocrine autoimmune TYPE I". Despite the global differences

negligible part of words involved into the disease names was useless for the description: words like "syndrome", "disease", "disorder", "deficiency", $\cdots$ are not descriptive and so we can filter them. Now, we could split the disease name into a series of token according to the list of words which compose it (*tokenization*) and sort them.

To further increase the overlap we cut the inflected words to their root form using a *stemming* algorithm: the stemmer strength has to be tuned according to the desired result. A first processing was performed using a Lancaster stemmer (more aggressive). If the resulting output was too short to be compared with other names the starting token was processed by a Porter Snowball stemmer (less aggressive). The stemmer algorithm choice is a very crucial task for NLP because using it we drastically loose information (it is not reversible). Other processing steps were performed for critical cases encountered during the analyses: these steps constrain our pipeline and they tuned it for the underlying application.

The work-flow outputs include multiple false-positive matches: the pipeline performs a brute force processing and some information lost along the steps could be significants. In these cases we have multiple processed names belonging to different kind of diseases: an example is shown in Fig. 1.4. Considering the original name and the processed one (pipeline output) we merged two names using a score match. This can be achieved introducing the standard word metrics: a common distance between words can be evaluated using the *Levenshtein Distance* which follows the equation

$$d_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} d_{a,b}(i-1,j) + 1 \\ d_{a,b}(i,j-1) + 1 \\ d_{a,b}(i-1,j-1) + 1_{(a \neq b)} \end{cases} & \text{otherwise} \end{cases}$$

where a and b are two strings of length $|a|$ and $|b|$ respectively. The *indicator function* $1_{(a \neq b)}$ is equal to 0 when $a_i = b_j$ and 1 otherwise. In this way the Levenshtein distance between a and b can be computed evaluating the distance between the first i characters of a and the first j characters of b. Despite the apparently complex mathematic formulation of this function, the *Levenshtein Distance* is a particular case of the more general *Edit Distance*, i.e a way to quantifying how dissimilar two strings are to one by counting the minimum number of operations required to transform one string into the other. Also in this case an example could be explainer: given the two string "*kitten*" and "*sitting*" their *Levenshtein distance* is equal to 3, in fact

1. **k**itten → **s**itten (substitute "s" for "k")
2. sitt**e**n → sitt**i**n (substitute "i" for "e")
3. sittin → sittin**g** (insert "g" at the end)

Using the Levenshtein equation we evaluated the distance between the two original names and we associate the disease to the higher scorer. A summary scheme of our pipeline is shown in Fig. 1.4.

The described NLP pipeline further increases the overlap between databases (e.g CTD-SNAP 24.17%; DisGenet-RXList 19.78%). We manually supervised and checked the merging procedure taking care to reduce the false positive percentage. In some cases the overlap percentage remained low also after the application of our pipeline (e.g. RXList-HMDB 8.03%; SNAP-HMDB 0.39%).

between the two names, given in this case by upper- and lower-cases of some letters and the deletion of some words, a very critical odds is the enumeration style. The performances of our pipeline dramatically increased using a roman_number_converter algorithm.

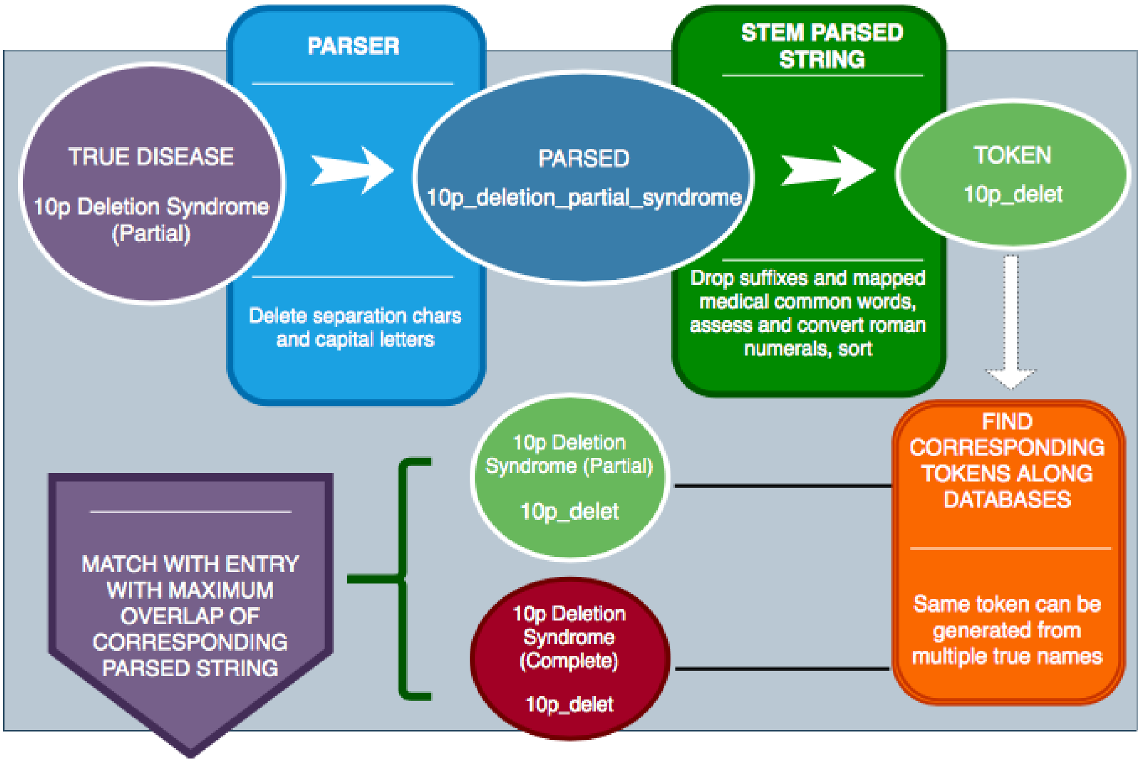

PARSER
Delete separation chars and capital letters
STEM PARSED STRING
Drop suffixes and mapped medical common words, assess and convert roman numerals, sort
TRUE DISEASE
10p Deletion Syndrome (Partial)
PARSED
10p_deletion_partial_syndrome
TOKEN
10p_delet
FIND CORRESPONDING TOKENS ALONG DATABASES
Same token can be generated from multiple true names
10p Deletion Syndrome (Partial)
10p_delet
10p Deletion Syndrome (Complete)
10p_delet
MATCH WITH ENTRY WITH MAXIMUM OVERLAP OF CORRESPONDING PARSED STRING

Figure 1.4: Scheme of the NLP pipeline developed in the CHIMeRA project. The disease words are processed in multiple step as showed in the example.

Different data sources could be focused on different types of information and it is therefore reasonable to assume that in some cases the overlap is low. We supervised these critical cases with a manual check and we demonstrated our hypothesis. This behavior supports our databases choice: they include complementary information which could improve the informative efficiency of our structure. At the same time this result also proved the efficiency of our pipeline and it confirmed that the union of multiple data sources could effectively enlarge our knowledge about biomedical compounds.

The output of our merging procedure allowed the realization of the CHIMeRA network, i.e a network with more than 3.6×10^5 nodes and more than 3.8×10^7 links (ref. Fig.1.5). In our resulting structure we have 7 node types: disease (63 974), drug (35 161), gene (18 799), SNP (117 337), metabolite (114 100), phenotype (13 214), metabolite-pathway (1 329) and food (532). The full network adjacency matrix is still a block matrix, i.e we have not all the combinations of information in our databases. An emblematic case is given by the food nodes: we have food information only into the DrugBank dataset and thus they would be connected only with drug types. On the other hand our network architecture could be easily improved adding new data sources: the same food nodes are pendant nodes that could be easily connected to other kind of data introducing novel node types or just filling the available blocks. CHIMeRA is still a work in progress project so we are still looking for improvements and new databases to add.

1.6 CHIMeRA analyses

The large amount of information provided by the CHIMeRA network has to be analyzed to prove its efficiency. A preliminary analysis was performed evaluating the node degree centrality (the number of links associated to each node). The degree centrality is the simpler measure to quantify the importance of a node inside a network and since our

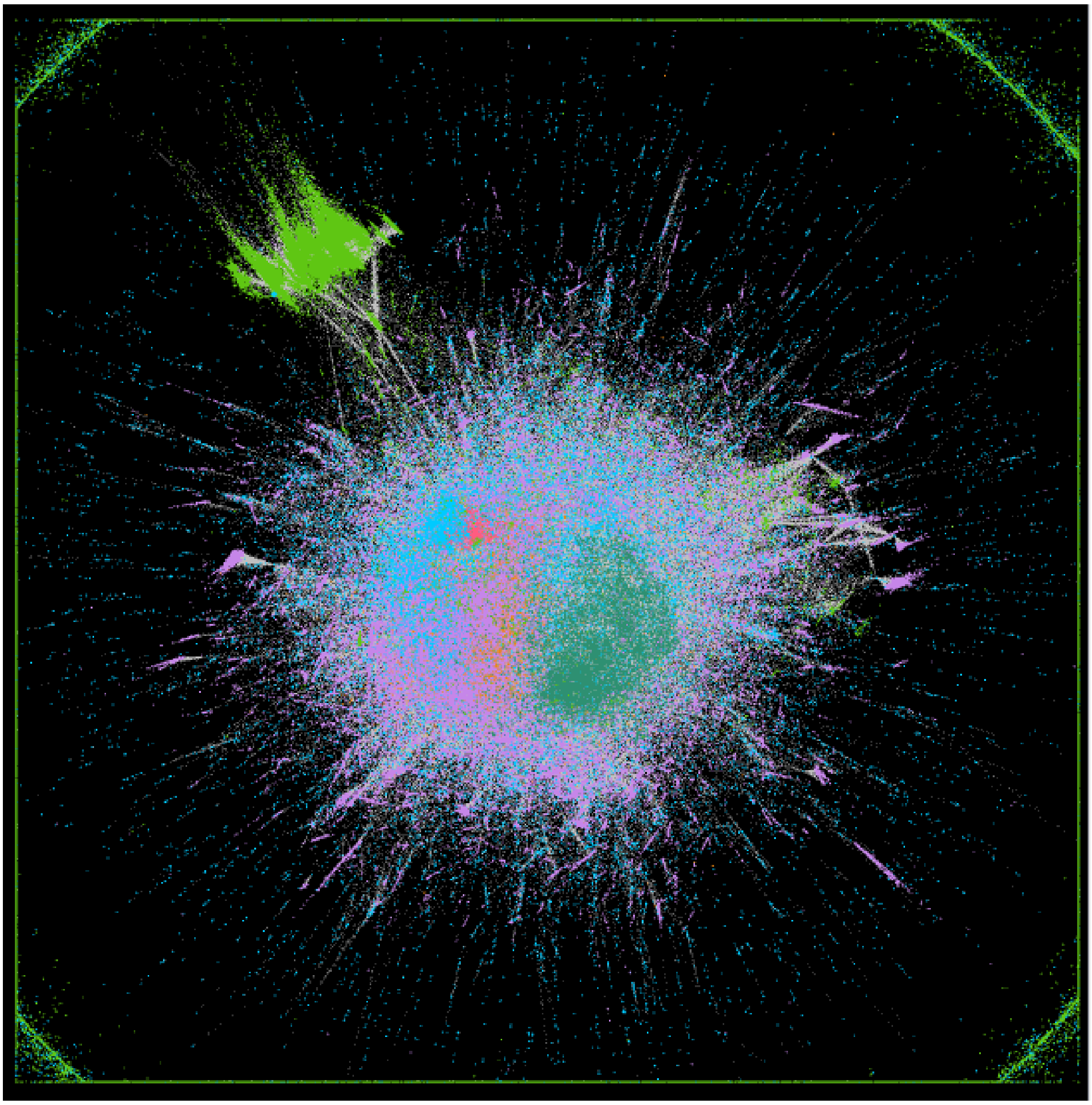


Figure 1.5: Graphical rendering of the first version of the textsfCHIMeRA network. The visualization was performed before the inclusion of the DrugBank dataset. For computational issues we have not performed newer image of the global structure. In the image we represented disease nodes (azure), gene nodes (orange), SNP nodes (purple), metabolite nodes (light green), drug nodes (pink) and phenotype nodes (dark green). The visualization was obtained by the *Atlas layout* provided by Gephi.

	mean	std	min	25%	50%	75%	max
global degree	121.44	784.86	1	1	2	7	108147
disease	18.54	109.25	0	0	1	3	22911
drug	93.59	737.60	0	0	0	0	17750
food	0.03	0.32	0	0	0	0	12
gene	1.96	38.81	0	0	0	0	5605
metabolite	0.37	154.73	0	0	0	0	85236
phenotype	5.50	100.09	0	0	0	0	9732
SNP	0.65	21.96	0	0	0	0	4866
metabolic pathway	0.09	2.81	0	0	0	0	594
disease pathway	0.05	1.27	0	0	0	0	283
drug-action pathway	0.14	6.79	0	0	0	0	1006
drug-metabolism pathway	6×10^{-3}	0.28	0	0	0	0	60
signaling pathway	4×10^{-3}	0.29	0	0	0	0	49
physiological pathway	1×10^{-3}	0.07	0	0	0	0	13
macro pathway	0.48	106.06	0	0	0	1	59993

Table 1.3: Degree connection statistics related to each node type stored in the CHIMeRA database. For each node type is reported the average, standard deviation, minimum value, maximum value and percentiles of the degree distribution. The first row shows the aggregated value of degree scores.

network-of-networks structure includes multiple node types we monitored it for each of them[11]. First of all we evaluated the number of connections of each node unpacking the value according to the possible node types. In this way we can perform a preliminary overview of the full set of information in the network. The summary results obtained by the degree centrality are shown in Tab. 1.3.

For each node type we computed the average number of connections and the main important parameters of each distribution (minimum, maximum, standard deviations and percentiles). This preliminary analysis confirms what we have already discussed during the creation of the CHIMeRA structure and thus that the each data type has at least one connection with a disease node (ref. min row in Tab 1.3): disease nodes are the core of our network-of-networks model while the other node types could have just single connections to the others. Using the finer grain distinction between the metabolite pathways (obtained by the information available in HMDB) we can also noticed that the major part of their connections concern the *macro pathway* category as expected: *macro pathway*s identify the more general category in our nomenclature and they include biological processes like apoptosis, dna replication fork and phosphatidylethanolamine biosynthesis. A better visualization of the network structure could be done counting the average number of connections between each node group, i.e the block matrix visualization of the underlying bipartite-graphs. In Fig. 1.6 is shown the block matrix representation.

In Fig. 1.6 we can better appreciate the connections between the available information and, moreover, we can visualize and quantify them. As expected the only node type which is connected to all the others is the *disease* one: the only exception is given by *food* nodes which are not directly connected with diseases since their information are available only in the DrugBank database and thus their connections are related only to *drug* nodes.

[11] We chose the degree centrality rather than other standard measures due to its numerical-simplicity/informative ratio. The CHIMeRA network includes a large amount of nodes so the algorithm complexity drastically affect the time performances. The degree centrality is given by the simple sum of the in- out-connections and thus it is faster also with large matrix as in our case.

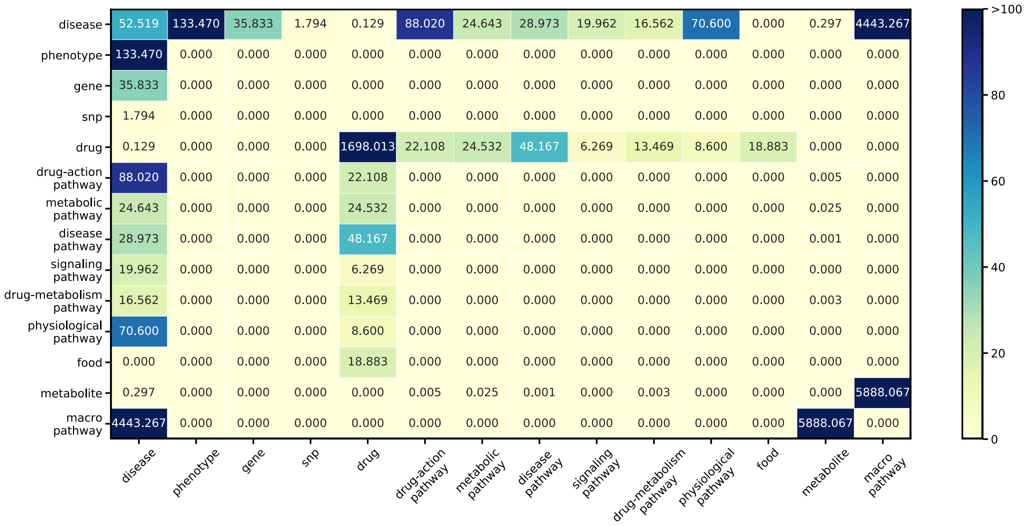


Figure 1.6: Block matrix representation of the CHIMeRA network. We computed the average number of connections between each node group and the bipartite-graphs structure is highlighted.

Our network-of-networks structure is very sparse and we have not direct information of the major part of combinations (null blocks). The only two node types which show a reasonably good interaction with the other blocks are the *disease* and *drug* ones[12].

The sparsity of CHIMeRA network-of-networks highlights all the pros and cons of our work. More a matrix is sparse and more its management could be efficient from a numerical point-of-view: we are able to use a wide range of algorithms developed and tuned according to the sparse algebra; also the memory occupation could be optimized and it is a crucial task when we work with such a big quantity of data. At the same time, it also highlights the potentiality of our work: each block connection derives from a single database evaluation (in first approximation) and we can reasonably assume that each block represents a possible output of query performed on that database. In our global database we have the join merging of all these information and using at least the 2nd neighbors of a node (the nearest neighbors of each nearest neighbor of a node) we could obtain a mapping of all the available information about it. Moving along the matrix blocks, in fact, we can start from a gene an we can only see the associated disease information which is comparable to a single-database query; since each disease is connected to all the other node types, its 2nd neighbors give us a panoramic view of all the biological compounds associated to that gene. This process is equivalent to an inference procedure of the missing blocks: if a gene is connected only to disease types, since all the other blocks are null, we could infer the missing blocks using the links provided by the disease nodes. This is the real power of such a network-of-networks structure. We would stress that the inference procedure could bring to incorrect biological associations since it represents only an hypothesis unbacked up by data. However, using more database sources we can easily integrate the missing information using the developed processing pipeline and, thus, increase the reliability of

[12] For sake of clarity we have to highlight also the two diagonal blocks in our matrix, given by these two node types. They arise from the synonyms and related causes in the first case (information provided by the synonym dictionaries and the RXList database), while in the second case they highlight the synergies or not (information given by the DrugBank database). A network adjacency matrix tends to nullify the diagonal information and node self-loops to prevent numerical issues and moreover to increase the amount of mathematical theorems for its analysis. We would stress that the showed matrix is not the adjacency matrix of the CHIMeRA network but it is an aggregate representation of it. Thus, in our network each node has connections only with other nodes and no self-loops are present.

our hypothesis.

To further investigate the information provided by our network using the computed degree scores, we evaluated the degree distributions of each node type. In Fig. 1.7 we show the degree distributions obtained considering the different node type individually.

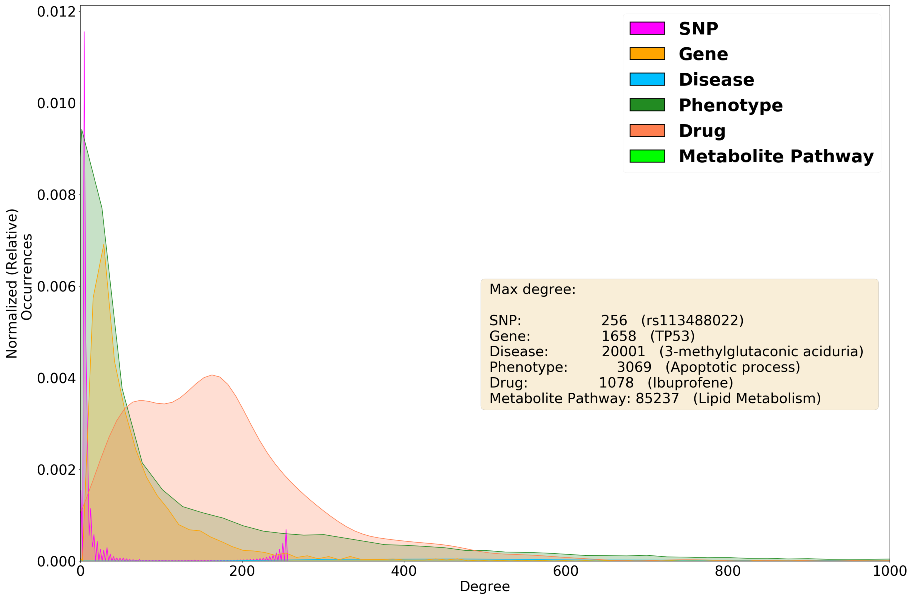


Figure 1.7: CHIMeRA degree distributions for different node types. The plot is cut for visualization purposes. The maximum degree node information are highlight in the box.

All the distributions showed in Fig. 1.7 have long tails but we cut them for visualization purposes. As expected and already highlighted by the previous analyses, the major part of node types have a not negligible amount of nodes with very low connectivity. Many node types have also isolated elements, i.e nodes with 0 degree. This behavior could be due to two possible causes: 1) our pipeline tends to remove some information and, thus, some true positive associations; 2) there are some missing information in the original databases that could not be overcome by our merging. Since the most part of isolated nodes are given by *metabolites* (ref. Fig. 1.5 in which the green dots around the plot are isolated metabolite nodes) we checked into the HMDB database the origin of this issues. In a not negligible number of cases the HMDB does not provide a disease association to a given metabolite and it proves our second hypothesis, preserving the efficiency of our pipeline.

A more interesting result is obtained considering the most central nodes, i.e the node related to the maximum degree score. This information could be used also as validity check of the structured. As in the SymptomsNet case (ref. 1.3), we expect a reasonable interpretation of the most central nodes. These results are shown in the yellow box in Fig. 1.7.

The most central node for the SNP node type is the rs113488022, well known gene-mutation validated by 72 public research. This SNP is related to a wide range of cancer diseases and its clinical significance has been proved in different studies. It is always hard to discuss about the more or less importance of a SNP mutation compared to the others, but its relation to many cancer types confirms its centrality score in our network structure. A more easy interpretable results is given by the most central gene, the TP53. The TP53 is a crucial gene in many tumor diseases and its importance is well mirrored in our network structure. The major part of diseases inserted into public databases are related to tumor researches and thus it was quite obvious to obtain a most centrality of this gene in our network. For the same reasons also the most central node for the *phenotype* type should be a tumor related characteristics: as expected the most central node in this case is given by the apoptotic process, i.e the process which regulates the programmed cell death, which is largely involved in tumor diseases.

As previously discussed about the *disease* and *drug* types, we have to consider a large set of available connections and, thus, we could expect that a central node in these cases would be given by a quite generic entry. For the *disease* type, in fact, we found as most central node the 3-methylglutaconic aciduria, a congenital metabolism anomaly related to leukemia. Despite this disease could be considered quite rare its importance in our network structure is certainly given by the large amount of metabolite connections (we remember that the HMDB database provide a very large amount of *metabolite* nodes that overcome the number of all the other node types, except by the SNPs). The large quantity of metabolites in our structure certainly weights on the number of disease connections and it proves the more centrality of a metabolic disease despite a genetic one. Moreover, we have also to take into account that this disease is also related to the leukemia disease, so we would have also a large set of genes and SNPs associated to it.

Considering the *drug* type we obtained a very common drug as expected, given by the *Ibuprofene*. *Ibuprofene* is a very common anti-inflammatory drug that is used for treating pain, fever and inflammation which are all very general disease symptoms associated to wide range of possible diseases. Thus, it is reasonable to assume that its number of connections is greater than other (more selective) drugs.

A key note has to be spent for the *metabolite pathway* type in which there are not apparently reasonable explanations which prefer the found lipid metabolism to other macro-metabolite pathways. To explain its centrality we, once more time, come back to the original databases and to the HMDB in this case. By a deep inspection of HMDB we noticed that the major part of them are studied using NMR chemical shift procedure. The NMR chemical shift is a very common spectroscopy procedure to analyze biological compounds but its signal is hardly related to particular nucleus types (e.g ^{1}H, ^{13}C, ^{15}N, $\cdots$). We could broadly describe this technique saying that as much hydrogen-like or carbonic-like structures are found into the biological sample and much the signal should be easy to analyze. The metabolite relation to the lipid metabolism is thus easy to study than other metabolism kinds due to the large quantity of resonant nuclei involved[13]. This proved the most centrality of the lipid metabolism regard other metabolite pathways.

These results are only preliminary analyses of the CHIMeRA network-of-networks structure, but they are already able to clarify some potentiality usage of our work. The only thing that remain to discuss is about the usability and release of this new database to the research community.

1.7 CHIMeRA as Service

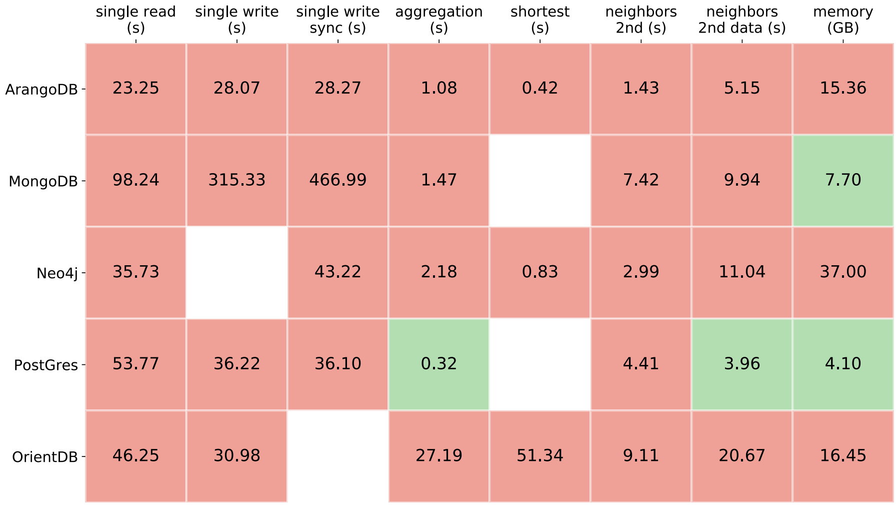
We have discussed about the information stored into the CHIMeRA database, but we have ignored how we could manage these data. More than the realization of a useful database, we have to provide an easy-to-use interface to encourage the research community to manage our processed information. We have already discussed about how the modern databases are shared along the Internet and how these large quantities of data could be managed using database languages (ref. 1.1). Now we have to find the best solution for our application.

We developed a first database version of CHIMeRA using SQL (*Structured Query Language*) language[14] and in particular the SQLite one. SQLite is probably the easier solution

[13] Fatty acids in lipid mixtures are widely studied using NMR chemical shift since their molecular structure involves multiple resonant nuclei suck ^{13}C, ^{31}P and ^{1}H.

[14] SQL is a domain-specific language designed for managing data held in a relational database management system (RDBMS) and it is particularly efficient in handling structured data.

for database management and the creation of efficient queries is straightforward. It is a well performing solution for standard relational databases but it does not provide any facility for network structure. Moreover, SQLite database is not directly comparable to client/server SQL database engines such as MySQL, Oracle, PostgreSQL or SQL Server since it is designed only for local data storage and individual applications. It is extremely efficient and simple in its applications but it does not cover the requirements pose by our CHIMeRA structure and our needs about sharing information.



	single read (s)	single write (s)	single write sync (s)	aggregation (s)	shortest (s)	neighbors 2nd (s)	neighbors 2nd data (s)	memory (GB)
ArangoDB	23.25	28.07	28.27	1.08	0.42	1.43	5.15	15.36
MongoDB	98.24	315.33	466.99	1.47		7.42	9.94	7.70
Neo4j	35.73		43.22	2.18	0.83	2.99	11.04	37.00
PostGres	53.77	36.22	36.10	0.32		4.41	3.96	4.10
OrientDB	46.25	30.98		27.19	51.34	9.11	20.67	16.45

Figure 1.8: NoSQL Performance Benchmark 2018 (source here). Absolute & normalize results for ArangoDBD, MongoDB, Neo4j and OrientDB. Comparison of time-performances using different (common) NoSQL queries.

A more efficient solution is provided by the modern graph databases (GDB). GDB are databases which use graph structures to represent and store information: there are two needed information for the database given by nodes and edges. The key concept behind this kind of storage is the relationship between the entries. They go under the NoSQL (*not SQL*, or better "*Not only SQL*") database category which storage data according to more sophisticated models than simple tabular relations (typical model of SQL databases). GDBs allow simple and efficient retrieval of complex hierarchical structures by definition and thus they represent the most efficient solution for our CHIMeRA database which is born as a network-of-networks architecture. Multiple different solutions have been proposed to address graph storages and there are a wide range of possible GDB languages public available on-line (e.g Neo4j, OrientDB, Sparksee, AllegroGraph, $\cdots$). Based on our experience about these topics and driven by the available documentation, we have chosen to use ArangoDB in our application. ArangoDB is an open-source and free software released on Github for multi-model database management with a unified query language AQL (*ArangoDB Query Language*). ArangoDB database system is NoSQL but its queries are very closed to SQL ones and thus are easier to write also by no-expert users. The core is written in C++ and thus extremely efficient from a numerical point-of-view (ref. Fig.1.8). Moreover, it provides also a user-friendly web interface for network visualization and query development. The possibility to have a web interface allows an easy way to share our database on Internet as service increasing the usability of our tool. Moreover, the query outputs can be also downloaded and used by external tools. Thus, using ArangoDB as service management we can provide a *Software as a Service* (SaaS) interface of our CHIMeRA database. This project is still in work in progress and this SaaS is not yet public available[15].

[15] As soon as possible we intend to create it jointly to an adequate computational environment ables to

We re-formatted the CHIMeRA network following the ArangoDB requirements and we created the graph database structure of our data. Using this database we were able to perform the first queries and discuss about the results. The University of Bologna is currently involved into the HARMONY European project for the analysis of hematological data provided by multiple pharmaceutical companies. The HARMONY project aims to describe, analyze and model multiple the data collected by the various partners producing a personalized medicine framework for the study of hematological diseases. This project is based on the harmonization of different databases in the same way as our CHIMeRA project aims to merge multiple public data sources. The main focus of the HARMONY project is about the diseases related to the different kinds of *leukemia*. *Leukemia* is the most common type of cancer in children and it causes hundred of thousands of death every year. It is a hematological disease and the exact causes of it is still unknown. The developed CHIMeRA project could be used to contribute to this kind of researches giving a wider biological panoramic overview about these diseases. Thus, we decided to formulate our first query about the *leukemia* disease.

We customized our query to extract only the 2nd neighbors related to this node. The pseudo-code used for our queries is shown in 1.6.

Listing 1.6: CHIMeRA 2nd neighbors query

```
FOR x IN node_type_vertex
  FILTER x.name LIKE "looking_for_entry"
    FOR v, e, p IN 1..3 ANY x GRAPH "CHIMeRA"
      RETURN p
```

The query takes the node-collection (ArangoDB nomenclature) related to the searched node type (node_type_vertex in the code) and filter all the names which satisfy the LIKE condition. Starting from the found nodes it returns the output graph preview made by the 1st and 2nd neighbors (range of values 1..3 in the code).

We applied this kind of query for the *leukemia* node and we processed the results using Gephi as network viewer. The obtained network is shown in Fig.1.9: the network involves 9 460 nodes and 26 646 links. As can be seen by the plot, just considering the 2nd neighbors the obtained subnetwork is quite large and it highlights the biological complexity of this disease.

Using the “generic” name of *leukemia* we found 291 different types of leukemia diseases into the CHIMeRA network which denote the different facets of this disease. Despite these multiplicities of results, we noticed that they clustered in only 82 connected components highlighting multiple similitudes between them. In particular we found a giant components of 9 108 nodes and only other 6 components with more than 10 nodes. The giant component includes 165 different facets of *leukemia* disease while the other connected components describe the remaining ones. The powerful of CHIMeRA network born exactly from these cases, in which we can infer missing information starting from the knowledge about analogous researches given by the full set of information related to the giant component found. In the giant component we can appreciate a description of the *leukemia* disease given by all the other node types: we have, in fact, 587 diseases related to them, 4 drugs, 2409 genes, 40 metabolites, 154 metabolite pathways, 5195 possible phenotypes related to them a 719 SNPs. The diseases associated to *leukemia* can help to highlight possible analogies between this “difficult” disease and “easier” ones (cause and related disease connections) or simply provide a bridge to other node types (e.g drugs or genes) which are not directly related to the *leukemia* using the databases individually. We would stress that, despite the *phenotype* node-type which includes the more general biological information, all the other amount of node-types represent only a small percentage of the available information (disease 0.9%,

support multiple external queries.

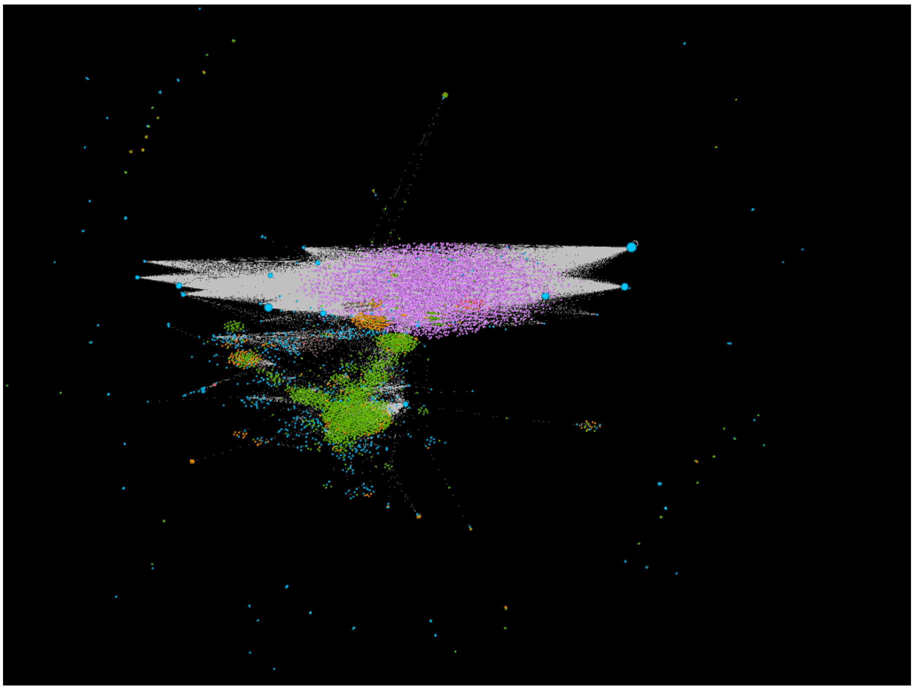


Figure 1.9: Output of *leukemia* query obtained by CHIMeRA graph database using 1.6. The subnetwork is made by the 2nd neighbors starting from all the nodes which include "leukemia" in their names. The subnetwork includes 291 different types of leukemias clustered into 82 connected components. The giant components is made by 9 108 nodes. CHIMeRA query is able to give a panoramic biomedical overview of the *leukemia* diseases mapping 838 diseases, 2 463 genes, 5195 SNPs, 154 metabolite pathways, 40 metabolites and 5 drugs associated to them.

drug 0.01%, gene 12.8%, metabolite 0.03%, pathway 11.5%, SNP 0.6%, phenotype 39.3%). It is important to monitor also this kind of percentage because it could bring to possible biases in our description. A such biomedical panoramic overview could not be found using a single-database approach and, to the best of the author's knowledge, only the CHIMeRA database is capable to map them.

The subnetwork extracted has more than half nodes as pendants (5 270/9 108 or 57%), i.e with degree score equal to 1. We have already discussed about this feature of our CHIMeRA network and in also in this case we can use this behavior to connect other (possible) kind of information to improve our disease description. We are still working on the analyses of the extracted information and, especially, about their biomedical interpretation. Moreover, we have to see how we can combine our data to the HARMONY project samples. Thus, we conclude this chapter remarking the potential applications of such network-of-networks structure and its capability of give us a more global overview of biomedical compounds in scientific researches.

Bibliography

[1] M. Desai and M. Mehta. Techniques for sentiment analysis of twitter data: A comprehensive survey. pages 149–154, 04 2016.

[2] G.-S. et al. DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Research*, 45(D1):D833–D839, 10 2016.

[3] C. J. e. a. Grondin. The Comparative Toxicogenomics Database: update 2019. *Nucleic Acids Research*, 47(D1):D948–D954, 09 2018.

[4] C. A. e. a. Hidalgo. A dynamic network approach for the study of human phenotypes. *PLOS Computational Biology*, 5(4):1–11, 04 2009.

[5] M. Lee, K. Lee, N. Yu, I. Jang, I. Choi, P. Kim, Y. E. Jang, B. Kim, S. Kim, B. Lee, J. Kang, and S. Lee. Chimerdb 3.0: an enhanced database for fusion genes from cancer transcriptome and literature data mining. *Nucleic Acids Research*, 45, Jan 2017.

[6] J. Loscalzo, I. Kohane, and A.-L. Barabasi. Human disease classification in the postgenomic era: a complex systems approach to human pathobiology. *Molecular systems biology*, 3, Jul 2007.

[7] A. e. a. Maciejewski. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Research*, 46(D1):D1074–D1082, 11 2017.

[8] L. Richardson. Beautiful soup documentation. *April*, 2007.

[9] K. Sharma, F. Qian, H. Jiang, N. Ruchansky, M. Zhang, and Y. Liu. Combating fake news: A survey on identification and mitigation techniques, 2019.

[10] A. Sood, M. Hooda, S. Dhir, and M. Bhatia. An initiative to identify depression using sentiment analysis: A machine learning approach. *Indian Journal of Science and Technology*, 11(4), 2018.

[11] D. S. e. a. Wishart. HMDB 4.0: the human metabolome database for 2018. *Nucleic Acids Research*, 46(D1):D608–D617, 11 2017.

[12] X. Zhou and et al. Human symptoms–disease network. *Nature Communications*, 5.

[13] X. Zhou and R. Zafarani. Fake news: A survey of research, detection methods, and opportunities, 2018.

[14] M. Zitnik, R. Sosič, S. Maheshwari, and L. Jure. BioSNAP Datasets: Stanford biomedical network dataset collection. http://snap.stanford.edu/biodata, Aug 2018.