PathoMap: Gene-Organ Relationships A Literature-based Investigation, Visualization & Analysis

by

Abhijit Raj

Submitted to the Department of Computational Biology
in partial fulfillment of the requirements for the degree of

Master of Technology in Computational Biology

at the

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY

July 2020

Author .....
Abhijit Raj
Department of Computational Biology
July, 2020

Certified by .....
Dr. Debarka Sengupta
Assistant Professor(C.B., CSE)
Thesis Supervisor

PathoMap: Gene-Organ Relationships

A Literature-based Investigation, Visualization & Analysis

by

Abhijit Raj

Submitted to the Department of Computational Biology
on July, 2020, in partial fulfillment of the
requirements for the degree of
Master of Technology in Computational Biology

Abstract

Genetics has brought huge breakthroughs in understanding human health & well-being, diseases and treatment through modern methods such as personalized medicine. This has been possible through substantial research in the area that reveals how genes are directly linked to our health. As genetic information is passed down in the family, genetic conditions are also hereditary. It is therefore very important to understand the pathogenic role of genes.

As new research progresses at a tremendous rate, a lot of insights can also be drawn from the volumes of literature already published by the scientists. Text mining and NLP have become indispensable tools to analyse large amount of textual data such as scientific literature and derive insights from it. Information preserving NLP methods distill from a vast corpus, the relevant pieces of information such as gene-organ relationships, pathogenic role of genes and more.

Ambitious efforts are being made to map the human body at the cellular level to understand variations in cells and how they lead to diseases. In this study, we aim to investigate gene-organ relationships through existing literature. We exploit visualization extensively as a tool to accelerate our understanding of this data. We introduce PathoMap - a novel tool to visualize any organ-related data on the human body. It is the first Python package to plot such organ-specific information. In the context of gene-organ relationships, we use PathoMap to draw conclusions in both healthy and pathological conditions. We hope that our visualization tool, PathoMap will be widely adopted and used in a range of studies to visualize organ-related data.

Thesis Supervisor: Dr. Debarka Sengupta
Title: Assistant Professor(C.B., CSE)

Acknowledgments

With deepest gratitude, I thank *Prof. Debarka Sengupta* for his constant support and brilliant ideas. He laid out the project structure in clear milestones and planned the thesis with goal-oriented steps that always kept the project moving forward. This project would not have been possible without his guidance which he provided any time I needed it. Every time I approached him for advice during the course of this project, he put forward beautiful insights on how to further refine the project and provided a clear direction to proceed.

I thank my parents who have taught me the importance of time and given me the knowledge of devoting it to the more important things in life. My mother has always been the guiding light in any adventure I have taken and her constant encouragement keeps the spirits high whenever I doubt my abilities. I thank Kajal who sparked in me the desire to further my education. She continuously strives to improve - a quality I wish for all of us.

My fellow students, Priyadarshini for her help in understanding the pathological context of the data, Neha, Shivani and Atishay for helping with the data collection, Vaibhav for testing the features of the online platform, Princey for her one page a day rule. Thank you all for your support. My friends for their support and encouragement, when the going was tough.

I also thank all the faculty members and staff of the Department of Computational Biology and IIIT Delhi as a whole for their cooperation and readiness to always help us when needed. They have taught me a lot in academics and in life during the time that I have spent here.

I am grateful to my grandfather, Late. Shri H.Shrikant, for showering me with his infinite wisdom that has molded me into what I am today.

To all the golf balls*[1] *in my jar...

[1] https://www.theweatherprediction.com/humor/life/

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human body has been the subject of research for centuries. As research progressed, we understood that this complex machinery is made up of different systems that interact with each other. These interacting systems are composed of organs which are made up of tissues. Further research revealed that tissues are composed of **cells**. A cell is the basic building block of life. A human body is made up of trillions of cells. They come in different types and perform different functions. WBCs(White blood cells), for example, form the body's defense system where as RBCs(Red blood cells) carry oxygen to the different parts of the body.

What are these cells made of? How do they know what their function is? More importantly, if all humans are made up of the same types of cells, why are we all so different from each other? The answer lies hidden inside the cell in the form of a slightly acidic substance known as the **DNA** (Deoxyribonucleic acid). DNA is made up of bases Adenine(A), Cytosine(C), Guanine(G) and Thymine(T). It has small sections - sequences of A, C, T and G, that perform a crucial function in the human body. These sections, known as **genes**, carry information to make molecules called **proteins**. Proteins are crucial for our survival as they are the functional units of a cell. Each protein performs a specific function. So, the body must understand how to synthesize protein molecules.

The synthesis of a protein happens due to the instructions contained in a gene through a process called gene expression. Francis Crick [4] proposed a fundamental

rule about this information transfer, known as the **Central Dogma of Molecular Biology**. This rule says that there is a unidirectional flow of information from a nucleic acid(DNA or RNA) to a protein.

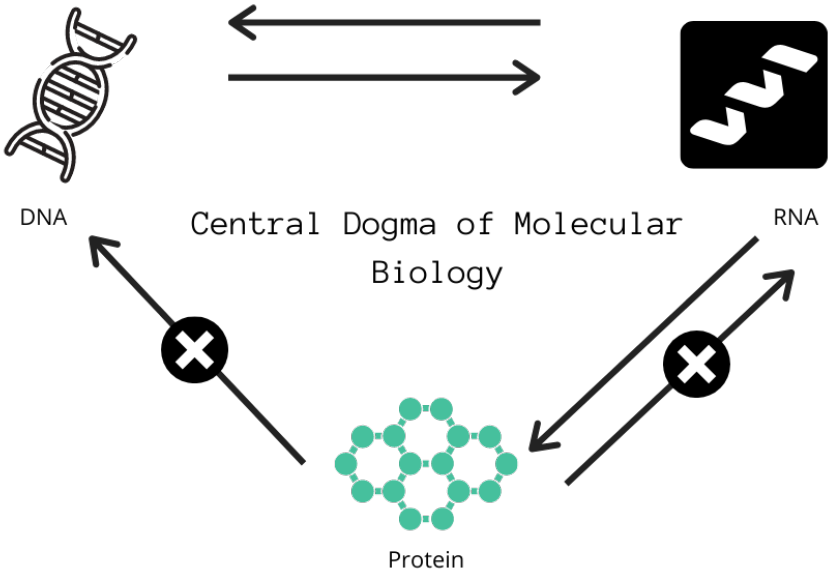

Figure 1-1: Central Dogma of Molecular Biology

As shown in the Figure 1-1, information can flow from DNA to RNA(Ribonucleic Acid) through a process called **transcription**. It can also flow from RNA to protein through a process called **translation**. What is not observed, however, is an information flow backwards from the protein. (It is worth noting here that some recent studies refute the Central Dogma of Molecular Biology [10])

Instructions for this information flow are stored inside genes. They also determine how we look like, or the color of our eyes and all the other features of our body. This links genes closely to human anatomy and physiology, as organs are made from these protein molecules. But where do these genes come from? The answer has led to huge breakthroughs in understanding the human body.

Genes are carriers of hereditary information. This is why we look like our parents.

We inherit genes from both our parents - the dominant copy determines our features. Thus, genes can tell us about our past. Studies to understand genes have revealed that genes have a direct linkage to human health/illness - they affect the parts of human body in which they are active. But why do genes affect human health?

1.1 Pathological Role of Genes

This pathological role of genes occurs through **mutations**. Sometimes, one or more bases in a gene's sequence get modified. This variation leads to different instructions for making the protein molecule. A protein molecule thus synthesized may vary in shape or function. Most of the time this mutation is harmless, but in rare conditions, it leads to some serious genetic problems. Sometimes, mutation in a single gene is enough to lead to a disease where as in other situations, environmental factors such as exercising habits, smoking etc also play a role.

Diseases that are caused by variations in genes are called genetic conditions. Since genes carry hereditary information, this variation can be passed down in the family. As a result, a person born with a certain variation in a gene may be more prone to a particular disease. For example, studies have shown that women with certain variants in the BRCA gene are at a high risk of developing breast cancer [29]. Another study attributes about 18% of disease risk in Crohn's disease (an inflammatory bowel condition) to genetic mutations in the NOD2 gene [9]. Other studies [25] associate Alzheimer's disease to genetics. Harlequin Ichtyosis, a severe skin condition that is often fatal, is caused by mutations in the ABCA12 gene which is responsible for making a protein essential for proper skin growth [12]. Many other examples of genetic disorders include Down's syndrome, Leukemia and Sickle Cell anemia.

This direct connection between genes and human health has sparked a lot of research in the area. From disease risk prediction to personalized medicine, several new discoveries are being made. Several studies and tools have tried to establish this gene-organ relationship using different methods - DAVID [11] uses biochemical pathways, where as Gene ORGANizer [6] follows a phenotype-based approach.

To assess the activity/expression level of gene in an organ, we introduce BodyMap - part of our PathoMap tool that can display gene expression data on human body. We follow an expression level based approach with the underlying data obtained from the GTex portal (https://www.gtexportal.org/home/). A gene that is highly expressed in an organ is expected to affect that organ when a mutation occurs.In our study, we will investigate such interactions and the pathogenic role of genes.

1.2 Human Anatomy

Human Anatomy deals with the study of the structure of human body. Our body is made of different types of organs - heart, lungs, stomach, brain etc. These organs interact together to form different functional systems that we call the organ systems. The digestive system, for example, is composed of the mouth, esophagus, stomach, pancreas, intestines, rectum and anus. The organ systems, though different, depend on each other to make the body a complete functional unit.

Human anatomical diagrams are an effective way of understanding how these organ systems are organized inside the human body. Figure 1-2 shows the typical human anatomical position with palms upright and spread out to the sides, feet slightly apart and facing the observer.

What makes the organs different? Why is the heart different from the brain and performs completely different functions compared to the other organs? An organ is made up of tissues which are created by cells. Different types of cells give rise to different types of issues. Cells of the nervous system, for example, are very different from the cells of the digestive system in function as well as structure. While skin cells are specific to the skin, bone cells help form the bone tissue and muscle cells form the muscles of the human body.

Single cell studies isolate individual cells through techniques such as FACS [5] in order to understand them further. Advances in single cell genomics show promise towards understanding the variations from cell to cell, and how despite those differences, the tissues and organs manage to work together. There are several ongoing

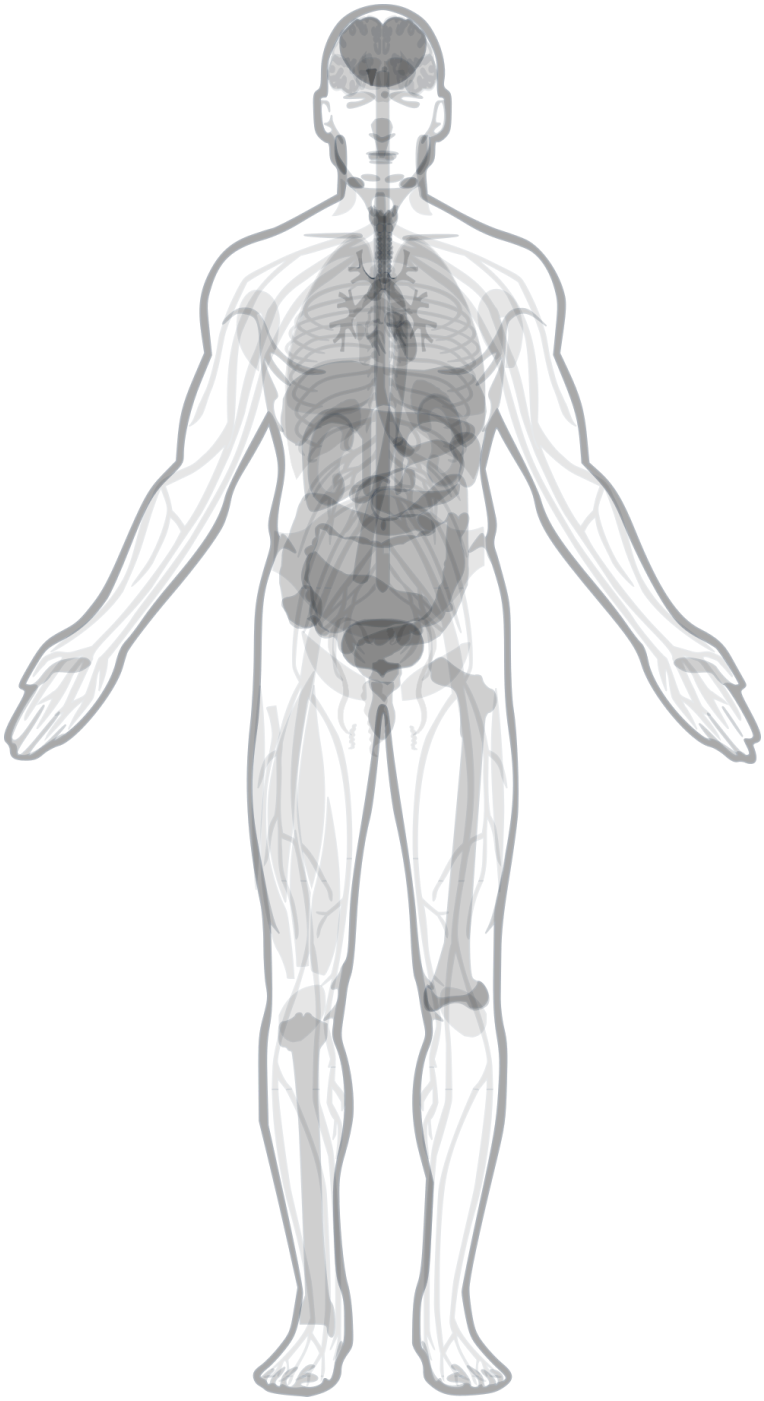

Figure 1-2: Human Anatomical Position: Drawn by our tool PathoMap using coordinates from *Expression Atlas*

efforts to create maps of single cell gene expression across the human body. Human Cell Atlas [24] is an ambitious project that aims to generate maps of different cell types and their location in the human body.

1.3 Importance of Visualizing Data through maps

Access to comprehensive reference maps of every cell in the human body can bring huge breakthroughs in health and disease treatment. It would enable us to understand how different cell types work together and how changes in their map can lead to diseases. It would help us reveal the identity of pathogenic genes and where in the human body are they active.

Biologists have too much raw data available to them. It is extremely helpful to have visualization tools that can help understand the data easily. A good visualization can significantly improve the understanding of available data. Moreover, when data is visualized against a familiar backdrop, it creates a mental model that enables the observer to make better connections. As an example, when gene expression data is visualized on human anatomical maps, it is easy to understand which genes are affecting which organs. Not only is it easy to understand, it also aides memory.

Prior to our work, no Python package was available that could plot/visualize any organ related data on the human body. Our tool PathoMap addresses that issue and we use it to analyse the gene-organ relationship in the pathological context.

1.4 Thesis Structure

We will begin chapter 2 by taking a look at the data mined through literature that forms the basis of this study. We will briefly explain the data acquisition techniques and preprocessing methods applied. Further we will deal with the annotation and classification of data. Then we will look at its transformation into pathovalues - values that indicate the relative significance of a gene under a disease in a particular organ. Finally, we will discuss the PathoMap database - an open repository derived from the gene-related information lying latent in previously published literature.

In chapter 3, we will introduce the PathoMap, a Python package for visualizing organ-related data. Prior to our study, no immediate package was available for plotting human organ related data anatomically in the Python programming language

- a language widely used by Computational Biologists today. We will look into the capabilities and features of PathoMap. Through case studies, we will demonstrate how this novel tool can be used to derive multiple interesting conclusions from the pathovalues discussed in chapter 2.

Chapter 4 presents the PathoMap web application - a tool designed to easily interact with the gene-organ data before exploring the complete features through the Python package. It comes with an easy to use interface, that displays gene-organ relationships in both healthy and pathological context. It also serves as a repository for all the available genes and their pathovalues with nice visualizations on the human body. One can search and find information for 15,000+ genes through this database. The web application is built with modern technologies that include ReactJS on the frontend, and Django on the backend.

In chapter 5, we will discuss the results of our study. We will look at pathogenic genes and how our visualization tools can help in understanding them. We will summarize the features of the PathoMap software. We will conclude with a discussion of future work and scope.

Chapter 2

The PathoMap Database - Background & Design

Genetic variations and their role in diseases have sparked a lot of interest in this area of research. The research community is producing papers at a staggering pace.

As a result, it is hard to keep up with the pace at which new work is getting published. Volumes of research material already published over the years also contain a lot of information on gene function and their pathogenic role. This already available information can be explored to generate multiple insights in the fields of disease risk, drug development & repurposing and more. To enable researchers to digest this information, several tools have been built using text mining techniques. Many published research papers have suggested that text mining and NLP(Natural Language Processing) methods can be utilized to extract useful information that is lying latent in the previously published literature. One study on disease-gene associations [18] focuses on using text mining to create DISEASES: a freely available database resource for disease-gene associations. Another study uses entity recognition for biomedical text mining [13].

The key idea that makes text mining useful in extracting relevant information from a text corpus is the fact that words with similar meaning usually appear in similar context. So, even without a rigorous training and background in the technical terms of a field, one can establish relationships based on the context in which the

words appear.

About 18 million abstracts form our text corpus. We selected abstracts as our data source because abstracts capture the key concepts and ideas of research papers. They are crisp, concise and avoid extra information unrelated to the main theme of the papers. We then quantify this data into numerical values through information preserving ML techniques described later in the chapter. This transformation allows for better processing and analysis on the data.

After preparing the Pathomap database, we use it to power Pathomap - an open-source package built as a part of this study to generate visualizations and develop mutliple insights which will be discussed in Chapter 3.

Below, we describe the various stages of PathoMap database development.

2.1 Data Acquisition

We acquired the literature data from NCBI servers through their text mining web services. We deployed an automated Python script to extract information in the *json* format. While the available data contains additional information such as the PubMedID, author details and publication details, our interest was in abstracts as they capture the objective and key findings of a research paper. The resulting corpus was composed of 18 million(approx.) abstracts. We subject this data to different preprocessing techniques as described below.

2.2 Data Preprocessing

Any algorithm follows the rule - Garbage In, Garbage out. To get better performance, a prerequisite is to remove the noise/clean the data. With the goal of refining the data for analysis by AI tools, we applied standard text mining techniques for preprocessing the corpus obtained in 2.1.

As a first step, we removed the metadata contained in the corpus. This included copyright information, reviews, acknowledgements etc. These do not contain the

information of interest - abstracts relevant to genes, gene functions, diseases etc.

Then, we cleaned the data by removing **HTML** tags and trimmed off the punctuation marks. We followed this with lower case conversion. Lower case conversion makes it easy to preprocess and parse the data. It also removes redundancy as words like 'Cancer', 'cancer', 'CANCER' etc. are no longer treated as different. We also converted numbers to their word forms to standardize the format even further.

Tokenization is a crucial step in NLP processing. It is a process to segregate the data into meaningful units of information. For example, a sentence can be broken down into words units or tokens, each of which carry a single bit of information. We used the **NLTK** [14] library which provides several open source tools for tokenizing the data and other standard tasks in NLP such as classification, tagging etc.

As a final step of our preprocessing on the text corpus, we apply **lemmatization**. Lemmatization is the process of identifying a representational word or lemma for a group of words that have similar meaning but are different due to grammatical uses. Lemmatization uses a language's vocabulary and removes inflectional endings and maps a word back to its dictionary form or *lemma*. As an example, lemmatization maps *read*, *reads*, and *reading* to *read*. It is useful as it helps in identifying the different forms of a word that have the same intended meaning.

2.3 Data Annotation & Classification

Not all abstracts obtained from NCBI, were relevant to our study of gene-organ relationships. We therefore, identified the subset of the corpus that relates to the field of molecular biology, genetics, gene functions, and human diseases. In order to effectively classify the dataset as positive(relevant to our study) and negative(not relevant), we used a binary classification method.

At first, we manually annotated a small subset of data. Approximately 2100 abstracts were manually classified as positive or negative. We trained a binary classifier on this data. A binary classifier is used whenever there is a yes/no question. For example: One can use a binary classifier to check if an email is spam or not. In

cancer therapy, whether a tumor is malignant or not is an important question that can be answered using a binary classifier.

We applied different binary classifiers including logistic regression, decision trees and SVM (Support Vector Machines).

A **Decision Tree** splits the dataset into smaller datasets such that we have a tree-like structure made up of - decision nodes and leaf nodes. The decision nodes implies more steps before classification, where as a leaf node means the classification result or label. The tree starts at the root node for an input and makes a decision based on some attribute. Then it moves on to one of the possible branches and recursively traces the path to a leaf node.

The **logistic regression** method uses a **logit** function to determine the probability that an entity belongs to a particular class. It can be used to perform binary classifications by calculating the probability of the entity belonging to one class or not. The logit function is the natural log of the odds of being in a class. Mathematically,

$$odds = \frac{p}{1-p}$$

where, p = probability of being in a certain class for a given condition

The natural log of the odds, can be used to classify an abstract as relevant or non-relevant.

$$ln(\frac{p}{1-p}) = x$$

SVM (Support Vector Machine) is a supervised learning algorithm that takes subsets of training data, called *support vectors* from each class to generate an optimal decision boundary (hyperplane). The idea of an optimal hyperplane is a decision surface that maximizes the margin around itself, i.e., the support vectors are the farthest from this hyperplane. We finally used the **SVM** classifier [26] as it performed better than the other techniques on our data. After classification, we obtained **7 million**(approx.) relevant abstracts.

2.4 Finding associations through Word Embeddings

In order to determine the relationship between words in our corpus, we first transformed them into their vector form or embedding. A word embedding is an n-dimensional vector that contains floating point values at each of n-dimensions. We used the **Word2Vec** [7] tool to generate these word embeddings.

Word2Vec converts the words into vectors which in turn envelops the entire text corpus into a few hundred dimensions. It works by assigning a unique vector to every word in the vocabulary. Thus, a word in the corpus has a corresponding vector representation in the vector space. By converting words into vector representations, the task of finding neighboring words becomes easier. And this process is also easy for any combination of words since now, the word combinations will yield just another vector in the vector space.

Each vector has a direction based on which its relationship with other vectors can be established. There are two popular methods used in Word2Vec - **CBOW**(Continous Bag of Words) and the **Skipgram** model. While CBOW uses surrounding words to predict the current word, the Skipgram model uses the current word to predict the words around it.

As an example, let us pick the word *gene* and input it to the Word2Vec skipgram. The skipgram will look at each word in the vocabulary and find the probability of it being a nearby word to *gene*. This is based on the idea that related words often appear togther in text. So, the skipgram will return higher probabilities for words like *mutation*, *expression*, *sequences* or *molecular* than for words like *fashion* and *Physics* which are unrelated to *gene*.

2.5 Pathovalues

In order to determine the nature of relationship between a gene and an organ in pathological condition, we needed a distance metric. One of the most common and widely used metrics with Word2Vec is cosine similarity [28].

Cosine similarity is a normalized dot-product of two vectors. Mathematically,

$$cosinesim(\vec{A}, \vec{B}) = cos(\theta) = \frac{\vec{A}.\vec{B}}{|\vec{A}||\vec{B}|}$$

where, $\vec{A}$ and $\vec{B}$ are two vectors. θ is the angle between them.

If cosine similarity = 1, vectors have the same orientation/meaning. If cosine similarity = 0, it indicates no relation and a cosine similarity of -1 indicates extremely dissimilar/opposite meaning words. Higher the cosine similarity is, the more similar the words are.

We call the values thus generated as **Pathovalues**. Each pathovalue is the cosine similarity score for a gene vector and an organ vector, based on the word embeddings obtained from the corpus of biomedical abstracts. The name pathovalue justifies itself in the sense that it is a measure of how related a gene is to an organ in pathological conditions.

We also applied **Z-scoring** to observe how the scores vary compared to a normal distribution. A Z-score gives an idea of how far a data point is from the mean value. As an example, if a person is 220 cm tall, with a Z-score we can find out how this person's height compares to the average height of the population. Mathematically,

$$z = \frac{x-\mu}{\sigma}$$

where x is a data point, μ is the mean, σ is the standard deviation

Thus, we generated a score matrix of 15,000+ genes and 37 organs where each cell in the matrix is a cosine similarity value of a gene vector and an organ vector.

2.6 Gene Expression Data

Several tools and approaches have tried to establish links between genes and the organs of the body they affect. Some such as DisGeNet [17] and PhenGenl [21] use phenotype-based approach, where as others like OMIM [8] only provide gene-organ relationship for certain diseases and limited organ sets.

We introduce an expression level approach, where we acquired Gene Expression data available from GTex(https://www.gtexportal.org/home/). GTex provides

gene expression for non-diseased tissues. GTex or Genotype Tissue Expression project is an initiative to build a comprehensive collection of gene expression in different tissues of healthy human beings.

From the GTex data we generated a gene-organ matrix where cell values indicate the relative expression levels of genes in different organs.

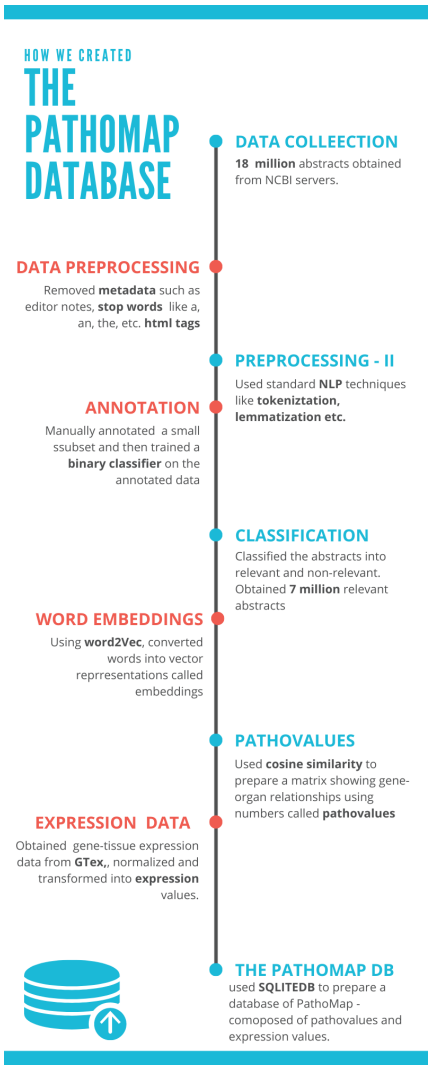

Figure 2-1: Pathomap DB design steps

2.7 The PathoMap DB

By collecting and processing the data in the manner discussed above, we obtained a distilled dataset of quantified values that retain the gene-organ relationship infor-

mation in both the healthy and pathological contexts in the form of expression and pathovalues. We converted this data derived from previously published literature into a database that could serve as a repository for different kinds of visualization tools and applications built on top of it.

We used SQLITEDB as our DBMS(Database Management System) which provides a fast, full-fledged database engine that is self-contained, cross-platform and requires minimum setup costs. It also provides the flexibility of moving to a more enterprise level database such as PostgreSQL when the need arises.

2.8 Conclusion

With a database derived from the information lying latent in the previously published literature at our disposal, we proceeded to create novel tools that would allow for comprehensive visualizations of the data on the human body. The database would serve gene-organ relationships in both pathological and healthy conditions and help us to identify pathogenic genes and hitherto unknown gene-organ relationships.

In the upcoming chapters we discuss the analysis tools we built as part of this study to visualize and draw insights from the PathoMap database.

Chapter 3

PathoMap - Python Package

While visualizing data on heatmaps and bar plots has its advantages, it is even more useful when the data is plotted on a related object e.g. the body of an organism related to the data. Jesper Maag in 2018 introduced gganatogram [15], an R package to plot data on anatomical maps of different organisms. But no such package was available for similar analysis in Python.

The Python PathoMap package solves that problem and adds more tools to the toolbox. While built for the analysis and visualization of the gene-organ relationship data that we have curated, it is extensible, can be used with any anatomical data in the correct format. In the following sections, we discuss the features and capabilities of the PathoMap package with examples.

We have implemented the PathoMap package to generate visualizations of data on human body with a single command. The information displayed is not limited to our pathovalues and the user can display their own data by conforming to the expected input of the command. Currently, data can be plotted and viewed on both human male and female body maps.

The package is in active development and we hope that the community finds it useful and contributes to its growth.

3.1 A Python Package

Python is a wonderful and exciting programming language that is suited to multiple tasks such as Data Science, Machine Learning, Web Development, Games and more. Many times when working in any of these areas, we need to solve a problem that has been previously solved by someone else. These good people make their solutions available freely online for others to use in their own projects.

A Python package is a collection of programs that collectively target a common task. For example, the Pandas package makes it easy to do data analysis and manipulation. Each program inside a package is a **module**, that does a specific task. In other words, a Python package is a collection of modules.

PyPI (https://pypi.org/) is an online repository of Python packages that hosts over 250,000 Python packages and more are added daily. These packages can be downloaded and installed through a one-line command. After installation they can be used in any project as an already available resource.

3.2 Software Resources

We have released the Pathomap package through the standard way of releasing Python software packages from https://pypi.org/. One can install and use Pathomap using Python pip as shown in the command below.

```
pip install pathomap
```

- **Source code available at Github**: https://github.com/Br34th7aking/Pathomap
- **Link to the development version**: https://test.pypi.org/project/pathomap/

3.3 Implementation Details

A Python package is surprisingly easy to create. You write your modules, pack them into a directory and add a special **__init__.py** file. This file tells Python to treat

this directory as a package. Then, you build the software into a distribution for your target OS(Operating System) and upload it on PyPI for others to download and install.

In PathoMap v1.0.0 available at the links in Section 3.2 we have implemented the following modules:

- **Bodymap**: The BodyMap module is responsible for plotting the human anatomical maps according to the organ-data supplied through a function call. Currently, it provides two anatomical maps - human male and human female. The maps drawn are SVG images. We obtained the co-ordinates for these SVG images from the Expression Atlas (https://www.ebi.ac.uk/gxa/home) website. The Bodymap module also allows you to draw organ level plots of different organs. You can adjust the size of these plots as well as pick a color of your liking.

- **The Gene module**: The Gene module provides functions for generating the expression values and the pathoscores for any gene of your choice. It accesses the PathoMap database, to return the result for your query. You can query these values for a single or a list of organs and sort them in ascending/descending order.

- **The Organ Module**: We created the organ module that provides functionality to explore the data through organs. You can query for an organ and get the top N genes for that organ.

These modules were then packed into a package, and built for cross-platform(OS independent) use and then uploaded to PyPI using **twine** - a standard Python tool.

3.4 Datasets

The main functionality of the Pathomap package is based on the data we have collected, annotated and processed. The resulting dataset used by Pathomap is:

- **Gene Organ Data Healthy**: A collection of 15,000+ genes with their computed scores for 37 different human organs for a healthy human.

- **Gene Organ Data Pathological**: A collection of 13,500+ genes with their computed scores for 37 different human organs for a diseased human.

3.5 Dependencies

PathoMap package is written in Python and requires Python >= 3.6 to operate successfully. Internal dependencies include Pandas library which gets installed if it is not already available on the system.

Most of the PathoMap package features work with the command line, but to view the body maps, it requires to be used with graphical tools such as the Jupyter notebook.

3.6 Example Usage

Pathomap comes packed with a range of features some of which are discussed below through examples.

To use the PathoMap package, it needs to be imported in the standard Python way.

```
from pathomap import Gene as g # accesssing the Gene Module
```

With a one-line command, PathoMap provides access to the pathoscores for any gene in our database.

```
g.pathoscore('MT–TL1')
```

When run, it returns a data frame containing the pathoscores for that gene across all 37 organs in the dataset. By providing additional parameters, it is possible to normalize the data and view the results in descending order. It is also possible to limit the search to a particular set of organs.

```
g.pathoscore('MT-TL1',
             organ_list['liver', 'lung'],
             normalize=True, desc=True)
```

Executing the above command, displays the data frame containing the Z-score normalized patho values for lung and liver in decreasing order for the gene MT-TL1.

It is also possible to explore the data organ-wise using the *Organ* module of the Pathomap package.

```
from pathomap import Organ as organ
organ.gene_values('stomach', condition='pathological')
```

The above line of code will return the genes with patho values for 'stomach'. The default limit is set at 10. It is possible to change parameter values to get genes in descending order by patho score, as well as Z-score normalised data for this organ.

```
o.gene_values('stomach', condition='healthy',
             normalize=True, limit=15)
```

When the limit is set to -1, Pathomap returns the data for all the genes.

3.6.1 Usecase: Healthy vs Pathological

With just the features mentioned above, powerful analyses can be done on the pathoscore dataset.

```
## compare values of same genes for a given organ
## under healthy vs pathological conditions
## get all the gene values using limit=-1
df1 = o.gene_values('stomach', condition='pathological',
                    normalize=True, limit=-1)
df2 = o.gene_values('stomach', condition='healthy',
                    normalize=True, limit=-1)


gene_list = ['MT-TL1', 'FGR', 'MT-RNR1']
```

```
## select the subset of the dfs
df1 = df1.loc[df1['gene_name'].isin(gene_list)]
df2 = df2.loc[df2['gene_name'].isin(gene_list)]
```

The block of code above, gives the patho scores for 3 genes MT-TL1, FGR, and MT-RNR1 for 'stomach' under both pathological and healthy conditions. This comparative analysis can be done with different organs and the same set of genes and much more.

As mentioned before, Pathomap is completely compatible with libraries such as *matplotlib*, so it is possible to use the plotting features available in combination with Pathomap.

Let us now explore some gene-organ relationships in pathological context through the PathoMap visualization tool.

3.7 BDNF Gene - Brain Relationship

PathoMap package has the built-in capability to plot any organ-related data on human anatomical diagrams - both for male and female. As an example, Figure 3-1 shows the PathoMap for the *BDNF* gene. The regions highlighted in purple indicate that this gene is pathogenically related to the nervous system, specifically the brain. (The color intensity is representative of the magnitude of the pathovalues). Our PathoMap database provides the cosine similarity with brain related areas as shown in Table 3.1. It is clear that *BDNF* has high cosine similarity with tissues in the brain.

A quick search on the NCBI gene database reveals that the BDNF gene is indeed related to brain diseases such as Alzheimer's, Parkinson's and Huntington's diseases, and may affect stress response and mood disorders.

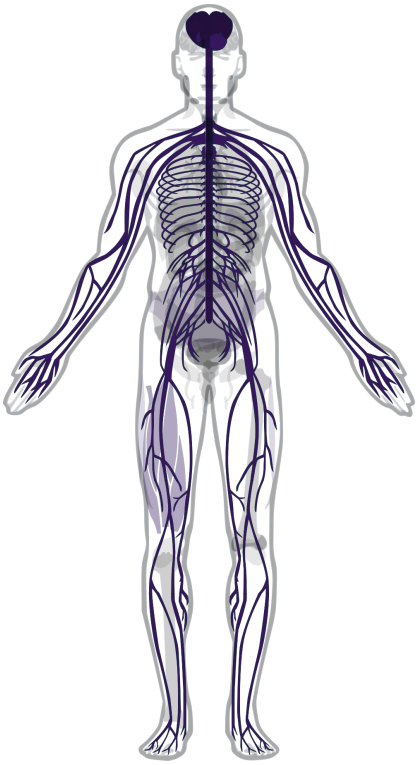

Figure 3-1: PathoMap of BDNF gene

Table 3.1: Cosine Similarity of BDNF gene with regions of the brain

Organ	Cosine Similarity
Hippocampus	0.618
Frontal Cortex	0.515
Amygdala	0.514
Hypothalamus	0.452
Cerebellum	0.4

3.8 EGFR Gene - Lung Relationship

The *EGFR* gene contains instructions for making the EGFR protein which is a cell surface protein and is responsible for epithelial growth. It forms the inner linings of organs of the human body, hence it is expected to be expressed in many organs. Figure 3-2 shows the bodymap generated from our PathoMap tool that shows the gene expression of EGFR gene under normal conditions. As expected, it is expressed in the entire body.

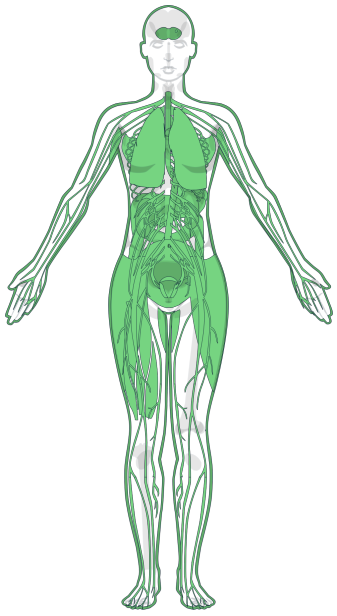

Figure 3-2: BodyMap of *EGFR* gene showing gene expression in normal conditions

The PathoMap plot however tells another story. PathoMap shows EGFR to be related to lungs, colon, and prostate under pathological conditions.

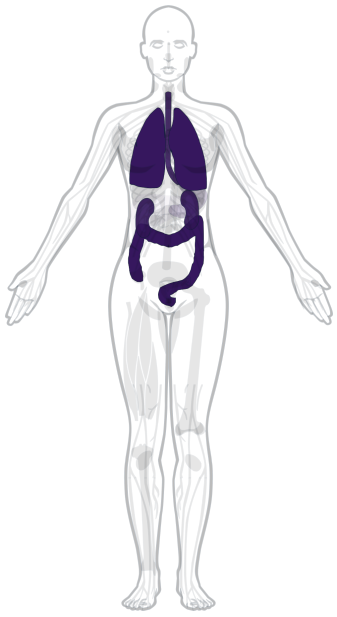

Figure 3-3: PathoMap of *EGFR* gene

Multiple mutations in *EGFR* gene have been linked to lung cancer [16]. Such mutations are only present in the cancer cells and cause lungs to develop a tumor. Interestingly, such mutations have a high chance of occurring in people who have

never smoked. This hints at environmental factors playing a role in genetic disorders.

3.9 AKT1 Gene - Various organs

The AKT1 gene contains information for synthesis of AKT1 Kinase which regulates growth, metabolism and cell survival. This gene is highly expressed in many different organs of the human body. Figure 3-4 illustrates the bodymap of AKT1 gene.

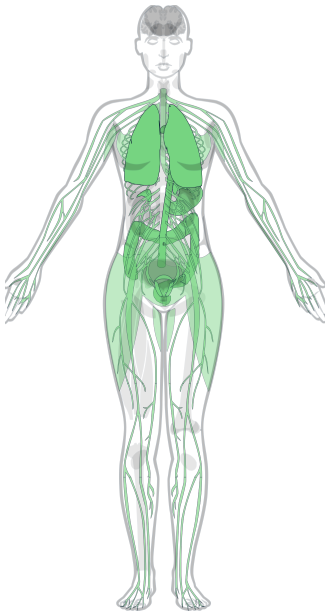

Figure 3-4: BodyMap of *AKT1* gene

Mutations in the *AKT1* gene can lead to several disorders including breast, colorectal and ovarian cancer, Proteus syndrome (overgrowth of tissues) and Schizophrenia. PathoMap highlights the related organs in Figure 3-5. Parts of the brain, breasts, colon, uterus etc. are highlighted in the figure.

3.10 CFTR Gene - Lungs & The Digestive System

Expression of CFTR gene directs the synthesis of the cystic fibrosis transmembrane conductance regulator protein. This protein acts as a channel via which water and ion are secreted & absorbed. It is responsible for the transportation of mucus, saliva,

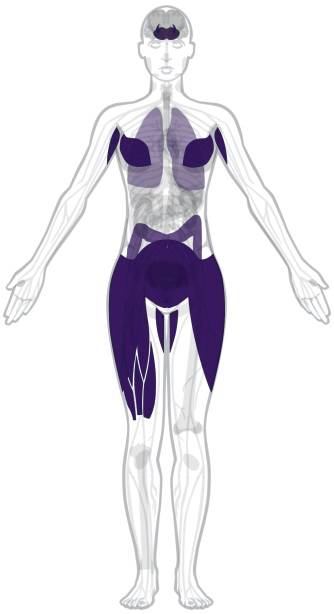

Figure 3-5: PathoMap of *AKT1* gene

tears etc. CFTR is expressed in different parts of the body including lungs, liver, gall bladder, colon, pancreas and intestines. The bodymap of CFTR is shown in Figure 3-6.

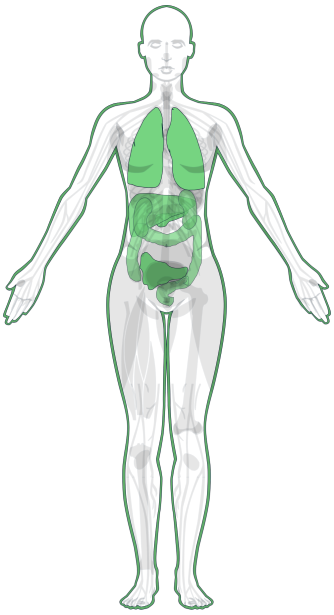

Figure 3-6: BodyMap of *CFTR* gene

Mutations in the CFTR gene lead to Cystic Fibrosis - a lethal disorder that causes thickening of fluids like mucus, sweat and saliva. Cystic fibrosis damages the lungs and

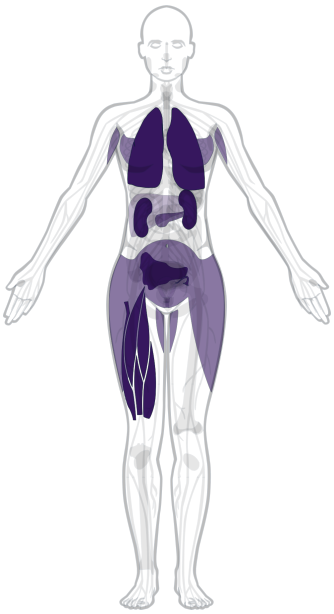

Figure 3-7: PathoMap of *CFTR* gene

the organs of the digestive system. Mutations in CFTR can also lead to respiratory defects and pancreatitis. The PathoMap shown in Figure 3-7 closely captures these associations with the related organs in the body.

3.11 Summary

We have developed and released a Python package that allows comprehensive and interactive visualizations of gene-organ relationships on human body. The package draws inspiration from gganatogram(an R-package for plotting anatomical diagrams) and uses the tissue coordinates available from Expression Atlas.

We have ensured compatibility with mainstream tools commonly used by biologists, thus, the package has potential to be a promising Python tool to plot organ-related data on the human body.

Chapter 4

PathoMap - Web Application

Scientists and researchers from all fields (Biology included) have embraced computing as a vital tool in their research work. However, there is still a need for an easy to use interface, for those not yet versed in the art of using software tools. The PathoMap web application provides a familiar environment for users to explore, and understand the capabilities of PathoMap before they can use it in their own projects.

We discuss the application's architecture, features, and capabilities in this chapter. There is also a how to use section that provides all the information to get started with PathoMap.

4.1 Application Architecture

PathoMap web-application has been developed using the API(Application Programming Interface)-first approach. This helps to keep the frontend and the backend of the application separate from each other. This separation allows the backend to be connected to several different frontend UIs on different platforms such as the web or the mobile phone without much rework. Figure 4-1 provides a rough picture of the web application's architecture.

The PathoMap web application uses Django REST Framework(a Django-based framework) for the backend [4.2.2] API development and React JS for the frontend [4.2.3]. Both Django and React are popular frameworks for creating highly scalable,

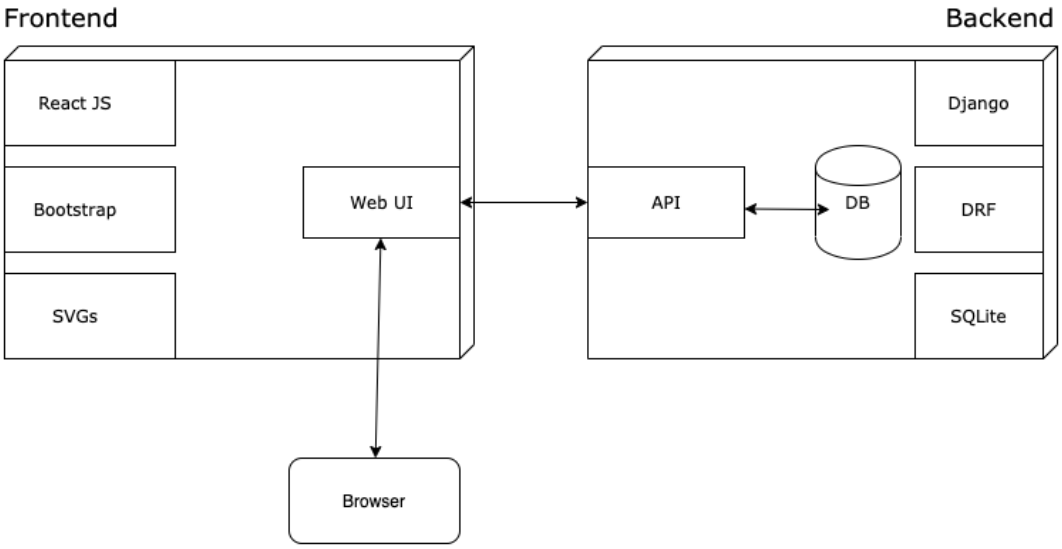

Figure 4-1: PathoMap - Web Application Architecture

performance-optimised web applications. The separation of the backend and frontend components makes the platform easily adaptable to any new technology on either end. These frameworks were chosen because they are actively developed and maintained and have risen as the standard way to build maintainable and robust applications.

4.2 System Design

PathoMap web application is a data-powered web software. Following standard practices, we have built the separated the application into three components - database, backend, and frontend. Below, we discuss each one of them.

4.2.1 Database

As discussed in Chapter 2, the underlying database of PathoMap web application is composed of pathovalues. We have followed the relational database design and used SQLite as our DBMS(Database Management System).

4.2.2 Backend Design

PathoMap's backend is built using Django and the Django REST framework for the APIs. Any data-powered application needs a database for CRUD operations. Django generates the database for PathoMap through its ORM(Object Relational Mapper) eliminating the need to write the SQL queries for every table in the DB. The application is currently using a SQLite DB which is quite capable for projects like PathoMap. For handling the HTTP requests, PathoMap uses Django REST framework, for creating the API endpoints which serve the data to the UI. REST(Representational State Transfer) APIs provide a range of benefits to a web application such as PathoMap including the following:

- Support all standard HTTP methods
- Light, fast and scalable
- Easy to understand and implement
- Highly performant
- Support multiple formats of data transfer

4.2.3 Frontend Design

PathoMap's frontend is built using React JS - a library designed by Facebook, for developing web applications. React JS follows a component-based architecture where the different parts of a webpage are designed using components. These components can be used anywhere in the application, much like how Python's import feature works. For adding responsiveness to the application, Bootstrap(specifically, the React Bootstrap library) is used. To draw the human body maps, I have utilized the SVG path features with tissue coordinates taken from Expression Atlas: https://www.ebi.ac.uk/gxa/home

4.3 Deployment

PathoMap version 1.0 is currently deployed as a free service on Heroku - a platform that allows developers to build and run web applications on the cloud. Just like you have containers in Docker, Heroku provides dynos, which are virtual Linux containers that run your code. In other words, a dyno is a web-server instance. You can find a demo application here: http://bodymap-demo.herokuapp.com/

4.3.1 Deployment Prerequisites

Every React + Django app needs to be built according to the server before it can be deployed there. For Heroku, below are the required steps:

1. Generating a folder structure according to Heroku requirements
2. Creation of a Procfile (this file contains commands to be executed by your Heroku dyno)
3. Pushing the app's source code to Heroku git
4. Heroku takes care of the rest and builds your code on the server

Figure 4-2 is a screenshot from the current deployment of the PathoMap web application.

4.4 How to use PathoMap

The current version of the PathoMap web application allows you to visualize the gene-organ relationship from data accumulated through extensive literature exploration. Currently, the data consists of 13,500+ genes with their impact on over 37 organs in the human body (both male and female). The PathoMap UI is extremely easy to use for anyone with some experience in using web apps. Below is a step-wise demonstration of how to interact with the platform.

Figure 4-2: PathoMap - Homepage

1. Enter the PathoMap URL in the browser: https://bodymap-demo.herokuapp.com/. You will be presented with the home screen with a search page.

2. Search for a gene of your interest. The search bar will provide recommendations based on your input.

3. You can choose to enable the Z-scoring method. PathoMap currently allows you to either use cosine similarity or z-score as a method to find the Gene-tissue similarity scores.

4. Click on the 'Search' button, and PathoMap returns the data related to that Gene visualized on the human body.

4.5 PathoMap Web Application Features

At its core, the PathoMap web application is an online repository of how genes impact human organs under normal and diseased conditions. It provides a searchable

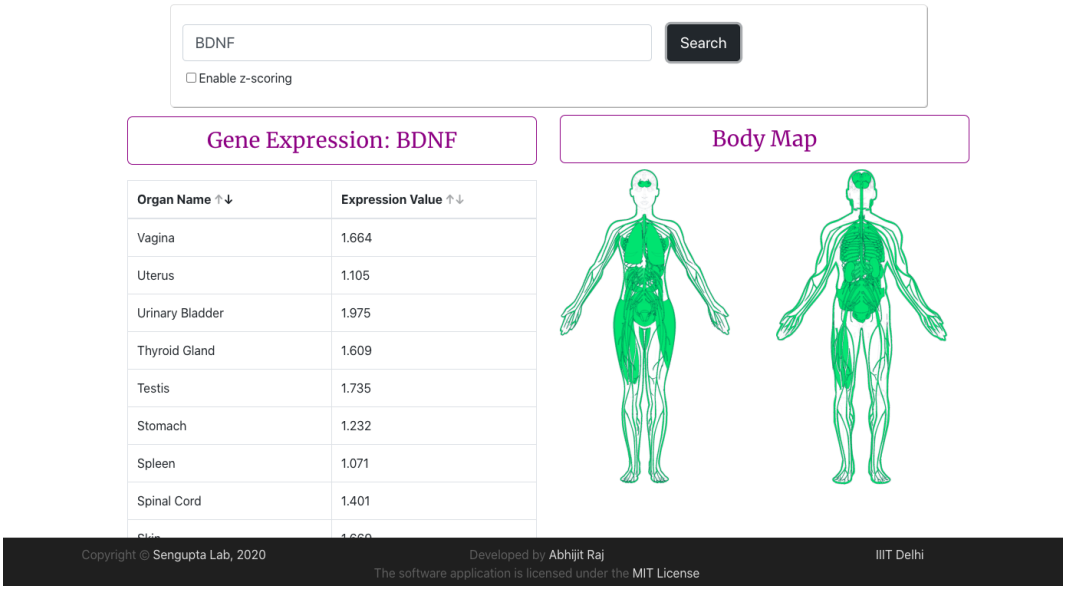

Figure 4-3: Gene Expression in Organs: BDNF

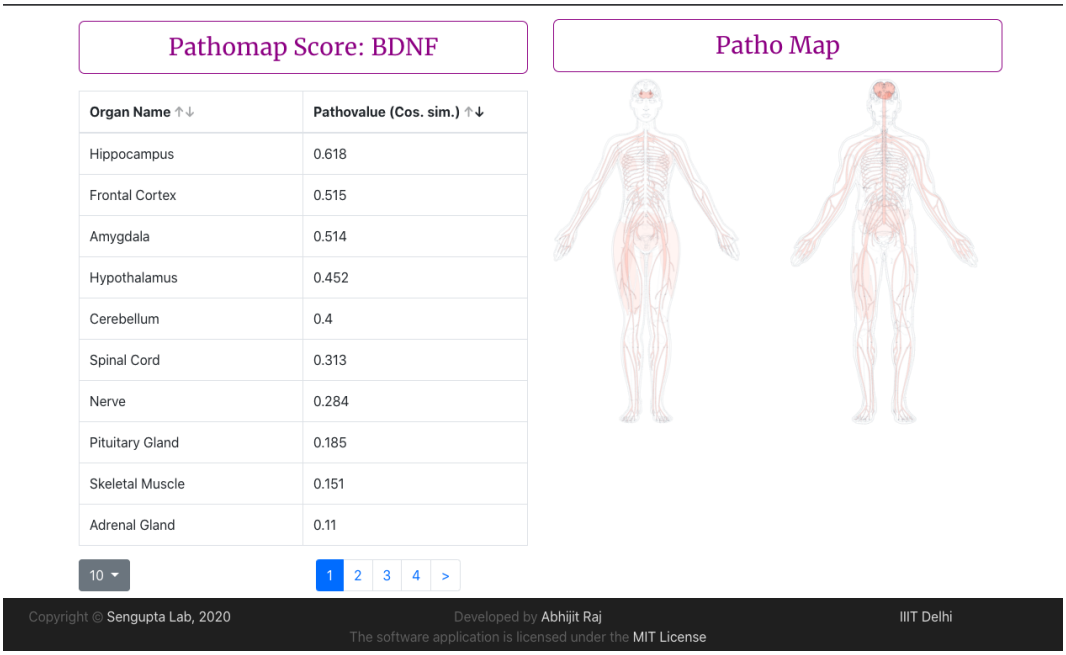

Figure 4-4: PathoMap: BDNF

database of thousands of genes, curated from published literature. Below are some of the main features of the web server.

4.5.1 Search Across a Vast collection of Genes

There are currently 15,000+ genes in our database. It is highly likely that you will find the information related to your gene of interest.

4.5.2 Sort and View Paginated Results

Sometimes, one is interested in questions like:

- What are the top 10 organs for XYZ gene?
- What is the organ most impacted by a gene?
- Which organ is the least affected?

You can get the answers to these questions simply by sorting the data provided in a neat tabular format. With a simple click, you can rearrange the view, and get the answer to questions like the above. For ease of digestion, the data is presented in a paginated format, that is easy to browse through.

4.5.3 Anatomical Maps of Human Male & Female

Utilizing the tissue coordinates from Expression Atlas, the PathoMap server, provides you a visual representation [4-3, 4-4] of the data on the human body(shown separately for male and female). These plots are SVG paths and provide a near accurate representation of the organs of the human body. You can hover over any organ, and get the score for that organ from the figure itself. The opacity of the organ colors in the body maps is proportional to the score for that gene-organ pair.

4.5.4 Different Scoring Methods

Currently, PathoMap offers two different ways to investigate the data.

1. Raw Pathovalues: These are obtained from during the data curation process as described in Chapter 2.

2. Z-Score: If you want to study how this data compares to the normal distribution, PathoMap provides you a Z-score feature which you can enable from the UI. Once enabled, you can see how far the data points are from the mean value.

The online version of PathoMap (https://bodymap-demo.herokuapp.com/) is intended to help biologists and other researchers who don't have sufficient technical skills in a language like Python to explore the data and use the PathoMap software.

Chapter 5

Conclusion & Future Work

5.1 Conclusion

This thesis achieves two goals - first, to distill the latent information about gene-organ relationships present in previously published research work through use of NLP techniques. We extracted this information and made it available in the form of an open-source repository that can be used and accessed by everyone. Secondly, to develop tools & methods that allow a comprehensive visualization of a large amount of organ-related data on the human body. We developed PathoMap - a Python package that can be used in a variety of studies related to organ data.

To illustrate the usability of the PathoMap package, and to validate our Pathomap database, we visualized the pathovalues on our software package and found that the pathovalues closely capture the gene-organ relationships. For example, the BDNF gene was found to be pathogenically related to brain diseases. Similarly, we validated that the EGFR gene has pathogenic relationship with lungs (cancer). Our visualizations also suggested an interesting property for genes such as the MLN gene which no protein expression in organs like Spleen but under pathological conditions, they appear to be related. This behavior can be further investigated in future work.

PathoMap provides a novel way to interact with organ-related data and we hope that it will be widely adopted in different studies related to human body parts. We have built the Pathomap Python package to be compatible with several mainstream

tools e.g. matplotlib, pandas. The data can also be exposed to existing analysis methods through the package. To make the data available through a more easily accessible interface, we developed a web application that is powered by the PathoMap database. It is built using the API-first approach, so the same data can be used by different web-based or even mobile applications. Biologists can use this application to explore the information from a rich gene-related literature within minutes and develop their own ideas.

The results we have obtained confirm that the pathogenic role of genes can be analyzed using scientific literature. There is also scope of discovering new relationships through the use of this data.

5.2 Future Work

There are myriad ways in which the gene-organ relationship data can be studied. One can choose to explore the ectopic behavior of genes. We can also investigate the data on an organ-system level such as genes and the nervous system.

For the tools we developed, the web application can be further enhanced by adding ways to visualize custom data on the body maps. We can also add the ability to do plot comparisons on the application.

The human bodymap coordinates can be further refined to provide better visualizations. We currently have tissue coordinates for 37 different organs of the human body. This can be extended to encompass all the organs in the human body.

Bibliography

[1] Stefanie Brüninghaus and Kevin D Ashley. The role of information extraction for textual cbr. In *International Conference on Case-Based Reasoning*, pages 74–89. Springer, 2001.

[2] Qingyu Chen, Yifan Peng, and Zhiyong Lu. Biosentvec: creating sentence embeddings for biomedical texts. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–5. IEEE, 2019.

[3] Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. How to train good word embeddings for biomedical nlp. In *Proceedings of the 15th workshop on biomedical natural language processing*, pages 166–174, 2016.

[4] Francis Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970.

[5] Anne Y Fu, Charles Spence, Axel Scherer, Frances H Arnold, and Stephen R Quake. A microfabricated fluorescence-activated cell sorter. *Nature biotechnology*, 17(11):1109–1111, 1999.

[6] David Gokhman, Guy Kelman, Adir Amartely, Guy Gershon, Shira Tsur, and Liran Carmel. Gene organizer: linking genes to the organs they affect. *Nucleic acids research*, 45(W1):W138–W145, 2017.

[7] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[8] Ada Hamosh, Alan F Scott, Joanna S Amberger, Carol A Bocchini, and Victor A McKusick. Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic acids research*, 33(suppl_1):D514–D517, 2005.

[9] Jochen Hampe, Andrew Cuthbert, Peter JP Croucher, Muddassar M Mirza, Silvia Mascheretti, Sheila Fisher, Henning Frenzel, Kathy King, Anja Hasselmeyer, Andrew JS MacPherson, et al. Association between insertion mutation in nod2 gene and crohn's disease in german and british populations. *The Lancet*, 357(9272):1925–1928, 2001.

[10] PETER HEUSSER. The central dogma according to watson and crick and its refutation by modern genetics. *When Healing Becomes Educating*, page 72, 1988.

[11] Lempicki RA. Huang da W, Sherman BT. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists., 2009.

[12] P David Kelsell, E Elizabeth Norgett, Harriet Unsworth, Muy-Teck Teh, Thomas Cullup, Charles A Mein, J Patricia Dopping-Hepenstal, A Beverly Dale, Gianluca Tadini, Philip Fleckman, et al. Mutations in abca12 underlie the severe congenital skin disease harlequin ichthyosis. *The American Journal of Human Genetics*, 76(5):794–803, 2005.

[13] Donghyeon Kim, Jinhyuk Lee, Chan Ho So, Hwisang Jeon, Minbyul Jeong, Yonghwa Choi, Wonjin Yoon, Mujeen Sung, and Jaewoo Kang. A neural named entity recognition and multi-type normalization tool for biomedical text mining. *IEEE Access*, 7:73729–73740, 2019.

[14] Edward Loper and Steven Bird. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[15] Jesper L V Maag. gganatogram: An r package for modular visualisation of anatograms and tissues based on ggplot2. 2018.

[16] Tetsuya Mitsudomi and Yasushi Yatabe. Epidermal growth factor receptor in relation to tumor development: Egfr gene and cancer. *The FEBS journal*, 277(2):301–308, 2010.

[17] Janet Piñero, Núria Queralt-Rosinach, Alex Bravo, Jordi Deu-Pons, Anna Bauer-Mehren, Martin Baron, Ferran Sanz, and Laura I Furlong. Disgenet: a discovery platform for the dynamical exploration of human diseases and their genes. *Database*, 2015, 2015.

[18] Sune Pletscher-Frankild, Albert Pallejà, Kalliopi Tsafou, Janos X Binder, and Lars Juhl Jensen. Diseases: Text mining and data integration of disease–gene associations. *Methods*, 74:83–89, 2015.

[19] Abhijit Raj. Pathomap python package: https://pypi.org/project/pathomap/. 2020.

[20] Satish K Rajagopal, Qing Ma, Dita Obler, Jie Shen, Ani Manichaikul, Aoy Tomita-Mitchell, Kari Boardman, Christine Briggs, Vidu Garg, Deepak Srivastava, et al. Spectrum of heart disease associated with murine and human gata4 mutation. *Journal of molecular and cellular cardiology*, 43(6):677–685, 2007.

[21] Erin M Ramos, Douglas Hoffman, Heather A Junkins, Donna Maglott, Lon Phan, Stephen T Sherry, Mike Feolo, and Lucia A Hindorff. Phenotype–genotype integrator (phegeni): synthesizing genome-wide association study (gwas) data with existing genomic resources. *European Journal of Human Genetics*, 22(1):144–147, 2014.

[22] Allan L Reiss, Michael T Abrams, Ronald Greenlaw, Lisa Freund, and Martha B Denckla. Neurodevelopmental effects of the fmr-1 full mutation in humans. *Nature medicine*, 1(2):159–167, 1995.

[23] Xin Rong. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.

[24] Orit Rozenblatt-Rosen, Michael JT Stubbington, Aviv Regev, and Sarah A Teichmann. The human cell atlas: from vision to reality. *Nature News*, 550(7677):451, 2017.

[25] Rudolph E Tanzi. The genetics of alzheimer disease. *Cold Spring Harbor perspectives in medicine*, 2(10):a006296, 2012.

[26] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[27] Vahe Tshitoyan, John Dagdelen, Leigh Weston, Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A Persson, Gerbrand Ceder, and Anubhav Jain. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763):95–98, 2019.

[28] Wikipedia. Cosine similarity: https://en.wikipedia.org/wiki/cosine_similarity.

[29] Jin Xu, Baosheng Wang, Yanjun Zhang, Ruihui Li, Yuehua Wang, and Shaokun Zhang. Clinical implications for brca gene mutation in breast cancer. *Molecular biology reports*, 39(3):3097–3102, 2012.

[30] Fan Zhang and Jake Y Chen. Homer: a human organ-specific molecular electronic repository. 12(S10):S4, 2011.

[31] Magdalena Zoledziewska, Gianna Costa, Maristella Pitzalis, Eleonora Cocco, C Melis, Loredana Moi, Patrizia Zavattari, Raffaele Murru, Rosanna Lampis, L Morelli, et al. Variation within the clec16a gene shows consistent disease association with both multiple sclerosis and type 1 diabetes in sardinia. *Genes & Immunity*, 10(1):15–17, 2009.