DEGREE PROJECT IN INFORMATION AND COMMUNICATION TECHNOLOGY,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, *SWEDEN 2019*

Domain Expertise–Agnostic Feature Selection for the Analysis of Breast Cancer Data

SUSANNA POZZOLI

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Domain Expertise–Agnostic Feature Selection for the Analysis of Breast Cancer Data

SUSANNA POZZOLI

Examiner: Sarunas Girdzijauskas
Supervisor at KTH: Leila Bahri
Supervisor at RISE SICS: Amira El Hosary

TRITA XXXX

Abstract

At present, high-dimensional data sets are becoming more and more frequent. The problem of feature selection has already become widespread, owing to the curse of dimensionality. Unfortunately, feature selection is largely based on ground truth and domain expertise. It is possible that ground truth and/or domain expertise will be unavailable, therefore there is a growing need for unsupervised feature selection in multiple fields, such as marketing and proteomics.

Now, unlike in past time, it is possible for biologists to measure the amount of protein in a cancer cell. No wonder the data is high-dimensional, the human body is composed of thousands and thousands of proteins. Intuitively, only a handful of proteins cause the onset of the disease. It might be desirable to cluster the cancer sufferers, but at the same time we want to find the proteins that produce good partitions.

We hereby propose a methodology designed to find the features able to maximize the clustering performance. After we divided the proteins into different groups, we clustered the patients. Next, we evaluated the clustering performance. We developed a couple of pipelines. Whilst the first focuses its attention on the data provided by the laboratory, the second takes advantage both of the external data on protein complexes and of the internal data. We set the threshold of clustering performance thanks to the biologists at Karolinska Institutet who contributed to the project.

In the thesis we show how to make a good selection of features without domain expertise in case of breast cancer data. This experiment illustrates how we can reach a clustering performance up to eight times better than the baseline with the aid of feature selection.

Keywords

breast cancer, clustering, clustering performance evaluation, feature selection, proteomics, unsupervised learning

Abstrakt

Högdimensionella dataseter blir allt vanligare. Problemet med funktionsval har redan blivit utbrett på grund av dimensionalitetens förbannelse. Dessvärre är funktionsvalet i stor utsträckning baserat på grundläggande sanning och domänkunskap. Det är möjligt att grundläggande sanning och/eller domänkunskap kommer att vara otillgänglig, därför finns det ett växande behov av icke-övervakat funktionsval i flera områden, såsom marknadsföring och proteomics.

I nuläge, till skillnad från tidigare, är det möjligt för biologer att mäta mängden protein i en cancercell. Inte undra på att data är högdimensionella, människokroppen består av tusentals och tusentals proteiner. Intuitivt orsakar bara en handfull proteiner sjukdomsuppkomsten. Det kan vara önskvärt att klustrera cancerlidarna, men samtidigt vill vi hitta proteiner som producerar goda partitioner.

Vi föreslår härmed en metod som är utformad för att hitta funktioner som kan maximera klustringsprestandan. Efter att vi delat proteinerna i olika grupper klustrade vi patienterna. Därefter utvärderade vi klustringsprestandan. Vi utvecklade ett par pipelines. Medan den första fokuserar på de data som laboratoriet tillhandahåller, utnyttjar den andra både extern data på proteinkomplex och intern data. Vi ställde gränsen för klusterprestationen tack vare biologerna vid Karolinska Institutet som bidragit till projektet.

I avhandlingen visar vi hur man gör ett bra utbud av funktioner utan domänkompetens vid bröstcancerdata. Detta experiment illustrerar hur vi kan nå en klusterprestation upp till åtta gånger bättre än baslinjen med hjälp av funktionsval.

Acknowledgements

I want to thank my examiner Sarunas Girdzijauskas and my supervisor Leila Bahri for guiding and helping me throughout the course of the degree project.

I would like to show my thankfulness to my industrial supervisor Amira El Hosary for all her work and all her precious feedback.

I wish to express my gratitude to Rui Mamede Branca for his contribution to the project.

Last but not least, I would like to thank my family, my friends, and Andrea for all their love and for all their support over the last months.

Contents

Chapter 1

Introduction

The use of high-dimensional data has become widespread; as a matter of fact, the mean number of features per data set has skyrocketed in recent years. For example, the field of proteomics analyses proteins in order to determine their connection with biological conditions such as diseases. Owing to the complexity of living things, the number of features is large in this case, i.e. thousands of proteins may be linked with a particular biological phenomena. However, it may be that the majority of proteins may not be relevant for a specific phenomena due to the fact that some are redundant, some are noisy, some are simply irrelevant. For example, it might be desirable to neglect the features strongly correlated to others as well as the features characterized by low variance. The remaining relevant features can be used to represent the high-dimensional data: this is advantageous in terms of computational resources and interpretability. At the same time, collecting samples in the field of proteomics is expensive both in terms of manpower and equipment, required to run laboratory tests. In summary, this type of data is characterized by a very low number of samples with a much larger number of features.

According to Cancer Today [Fer+18], in 2040 in Europe, there will be an estimated $567,564$ new breast cancer cases and $160,104$ breast cancer deaths. In this project, conducted in collaboration with Karolinska Institutet, the laboratory supplied us with data on breast cancer sufferers. There are $9,995$ gene products, or proteins, in the data set and only 45 patients. All these patients suffer from breast cancer, but different types of the disease. Earlier work by Johansonn et al. [Joh+19] on the same data set identified these types with success. However, it was necessary to hand-pick a subset of gene products known to contribute to the onset of the disease. Evidence supports their hypothesis that an abnormal behavior of some of the gene products is the cause of cancer. There is still a lot to be discovered when it comes to the inner workings of the human body and identifying which specific gene products contribute to a particular type of cancer. Hence, this work adopts a data-driven methodology in order to make it easier to identify candidate proteins which can be later investigated by the biologists.

1.1 Background

Perou et al. [Per+00] is one of the first to work on the classification of breast cancer cells. Unfortunately, their classification of breast cancer patients is based on the quantity of messenger RNA, whereas the usual cancer treatments regulate the amount of protein in a cancer cell, not the quantity of messenger RNA. In contrast, Johansonn et al. [Joh+19] grouped the breast cancer sufferers by protein content. The authors of the article were able to replicate the results of the original experiment by taking advantage of the domain expertise. In fact, 37 gene products out of $9,995$ were selected because it is a known fact that this group of gene products has a part in the onset of the disease. Basically, the authors of the article made a manual selection of the features. The authors were able to group the patients into different types of cancer, but the quality of such a grouping was not evaluated. In other words, it may be possible to find different groups of proteins that better separate the patients by cancer type in a data-driven way.

In order to find these groups, this work uses dimensionality reduction to move from the original high-dimensional space to a low-dimensional space. While feature extraction is the task of projecting the high-dimensional space onto the low-dimensional space, feature selection is the task of selecting from among the dimensions. There is a big difference between the former and the latter. Whilst feature extraction creates brand-new dimensions, feature selection chooses from among the existing dimensions without affecting the interpretation of the inference.

1.2 Problem

The majority of the feature selection processes require either the presence of ground truth or the availability of domain expertise, but in this particular case it is not possible to rely on the domain expertise because our understanding of the inner workings of the human body is limited. Indeed, there are still gene products almost unknown. Results are expected to point the experts to the gene products worth examining in deep.

The best option would be to test all the combinations of m gene products. It is just not feasible to use brute force. As a matter of fact, the number of combinations of m features is 2^m. For example, if the number of features is equal to $9,995$, then the number of combinations contains $3,009$ digits. It is like looking for a needle in a haystack.

Overall, the research question is as follows: “In absence of domain expertise, is it possible to make a selection of gene products which signal the onset of breast cancer as well as identify different types of breast cancer in an unsupervised fashion from the aforementioned data set?”.

1.3 Purpose

The purpose of the degree project is make a selection of features to try to find out more about the causes of the disease.

The whole idea of the project is to make use of the feature selection process because most likely just a handful of gene products will have a part in the onset of the disease. Also, it is a requirement that we will be able interpret the result of the experiments.

We want both to determine which gene products have a role in the onset of the disease and to characterize the breast cancer types in terms of gene products. Basically, we are interested in which gene products signal the presence of a given type of breast cancer, depending on patient. These groups of samples are expected to be uniform in terms of protein content. There are various options for the division of the features. Either two gene products belong to the same group because of a common behavior or the two features are put into the same group because these two gene product interact. In order to model the protein interaction, it is necessary to import external data, such as protein interaction networks, on the subject. For example, it is possible to extract a protein interaction network from the CORUM database [Giu+18], which connects proteins based on their cooperation in achieving a certain biological function.

1.4 Goals

In order to give an answer to the research question, it is necessary to go through a list of goals.

- To identify the feature selection process suitable for the analysis of breast cancer data.
- First, we want to extract some promising candidates from the input data. Secondly, we want to measure the clustering performance of the aforementioned candidates in order to determine which features are better at characterizing the samples.
- To identify the distance measures and the affinity measures suitable for the partition both of the gene products and of the patients.
- To join one external data set such as CORUM [Giu+18] or more, if possible.

1.4.1 Benefits, Ethics, and Sustainability

The most common cancer type is lung cancer, followed by breast cancer. The majority of cancer treatments such as chemotherapy regulate the amount of protein, whereas at the present time, we classify the patients by quantity of messenger RNA.

The project is expected to contribute to the development of new cancer treatments, which may bring new hope to cancer sufferers.

Apart from the study and treatment of cancer, the thesis contributes to Feature Selection for Unsupervised Learning. The methodology used in the research can be used for high-dimensional data if the domain expertise is not available.

The input data is sensitive. Thus, we handled it with care although it was anonymized data. However, given the type of cancer, it is possible for us to hypothesize that the patients are women.

In the long term, Feature Selection will bring many benefits. By extracting the relevant features from the irrelevant features, modeling the data will be an easy job. It may be that we will be able to extrapolate a trend from a huge number of features or to detect anomalies in the data.

1.5 Research Methodology

The idea of the thesis originated by Henrik J. Johansonn et al. [Joh+19]. First of all, we tried to replicate the results of the original experiment. The authors of the paper handpicked 37 gene products out of $9,995$ known to cause the onset of the disease. This group of gene products will be referred to as the baseline candidate as from now, given that it will be used for comparison with other candidates as detailed below.

We need to address the problem both of how to generate candidates and of performance evaluation of candidates.

1. *Candidate Generation.* To do so, we convert both the internal data on gene products and the external data on protein complexes into a graph. Next, we cluster the features by means of Spectral Clustering.

2. *Candidate Evaluation.* To do so, we convert the patients into a graph according to the candidate under evaluation. Next, we bisect the patient graph. In order to evaluate the clustering performance, we compare the candidate with the baseline.

We put two pipelines together. Whilst the first focuses its attention on the breast cancer data, the second takes advantage both of the external data and of the internal data,

This experiment demonstrates that the availability of domain expertise is not a requirement for making a selection of features.

Results show that it is feasible to find the gene products thanks to which it is possible to divide the collection of patients into homogeneous groups. The vast majority of the subsets of features are up to eight times better than the collection of gene products hand-picked by the experts.

1.6 Delimitations

We carried out an analysis of breast cancer data because all the input data is about breast cancer sufferers. Furthermore, the interpretation of the output data will probably be quite easy, since breast cancer is widely-known.

Unfortunately, the number of instances is small. It is likely that a larger number of instances will produce a better result.

We supposed that the intersection of the clusters of patients is empty, and as a result it is not possible that a point will be in two distinct clusters at the same time. Consequently, the thesis focused on what needed to be done to divide the samples into non-overlapping clusters.

Nevertheless, the methodology is expected to work in case of non-biological data, since we did not need to fine-tune the methods. In theory, it would work, but in practice we did not test it.

1.7 Outline

The following will explain the disposition of the thesis.

Chapter 2 presents the theoretic background and the related work. Chapter 3 presents the methods used in the degree project. Chapter 4 describes the results of the experiments. Chapter 5 presents the conclusion of the thesis and the future work.

Chapter 2

Theoretic Background

Reducing the dimensionality of a data set usually involves training a statistical learning model. We can think of a model as a black box that reads the input data and predicts the output data. There are two categories of model: supervised and unsupervised learning models. Classification is an example of a supervised learning model. The idea behind the classification of a collection of observations is to categorize them into classes identifiable from the corresponding labels. By knowing in which category each observation is in advance, we are able to train a model, but that is not the case when it comes to unsupervised learning. For example, we cannot count the number of wrong predictions in this case, because the true labels are unknown. For this reason, in the majority of cases, Supervised Learning is the preferred option for choosing features.

2.1 Dimensionality Reduction

Dimensionality Reduction is the task of moving from an m_1-dimensional space to an m_2-dimensional space, where $m_2 < m_1$, or even $m_2 \ll m_1$, if possible [Cun08]. Obviously, the low-dimensional data is fully expected to represent the high-dimensional data.

The following describes the two categories of methods in the field of Dimensionality Reduction.

2.1.1 Feature Extraction

Feature Extraction, also known as Feature Projection, is the task of projecting the m_1-dimensional space onto the m_2-dimensional space by creating brand-new dimensions via linear or non-linear combination.

For example, Principal Component Analysis (PCA) combines the dimensions in a linear fashion. As a matter of fact, this method of reducing the number of dimensions includes eigenvalues and eigenvectors. The idea behind the Principal Component Analysis is to explain as much variance in the data as possible by

means of orthogonal principal components. To do so, PCA finds the orthogonal directions characterized by the wider variance in the data [Cun08].

Eigendecomposition

The goal of the eigendecomposition of matrices is to perform matrix factorization by calculating eigenvalues and eigenvectors.

The decomposition of matrices is important to carry out multiple tasks, such as Dimensionality Reduction and Clustering; this is why we need to sketch an outline of the eigendecomposition of matrices.

Let $M \in R^{n \times n}$ be a square matrix.

$$M\mathbf{v} = \lambda\mathbf{v}, \tag{2.1}$$

where $\lambda \in \mathbb{R}$ is the n^{th} eigenvalue of M, and $\mathbf{v} \in \mathbb{R}^n$ is the n^{th} eigenvector of M.

As a rule of thumb, we see the n^{th} eigenvector as a dimension, and then we think of the corresponding eigenvalue as the variance in the data in the direction of the n^{th} eigenvector.

Usually the eigenvectors are scaled so that the magnitude is equal to 1.0.

2.1.2 Feature Selection

Feature Selection, also known as Variable Selection, is the task of selecting m_2 dimensions out of m_1.

The following describes the three categories of method in the field of Feature Selection.

Filter Methods

Filter Methods are being used to discard the bad features right away. To do so, firstly, the features are ordered according to numbers, such as the Information Gain and the Pearson's correlation coefficient, and secondly, the features with significance below a given threshold. After neglecting the insignificant feature, any statistical learning model would be trained only once using only the selected features.

There are not so many Filter Methods in the literature for the field of Unsupervised Learning. One of the few of them has been published by He, Cai, and Niyogi [HCN05]. To put it simply, the features are ordered according to their locality preserving power. Basically, the intuition is to attach a lot of importance to features able to agree with the structure of the input data in the shape of a nearest neighbor graph.

Wrapper Methods

Wrapper Methods are being used to limit the searching space. Heuristics such as Stepwise Selection are used for candidate generation. The whole idea of the algorithm is to find the most promising candidates. Next, to evaluate the performance

of the candidate under evaluation, firstly, the chosen model is trained according to the candidate, and secondly, the candidate is evaluated in a quantitative way based on the performance of the model.

It is worth remembering that we think of the statistical learning model as a black box.

For example, in case of Forward Stepwise Selection, we start from the empty set of feature. At every step, we add the feature making the greatest relative improvement to the statistical learning model [Jam+14]. Backward Stepwise Selection works the other way around, starting with all the features, all of which are removed step by step.

One disadvantage of this category of method is that the statistical learning model is trained n times, where n is the number of candidates. Thus, we are talking about high-cost algorithms in terms of resources. However, we get the best results with this group of methods [LY05].

Embedded Methods

Embedded Methods try to train the statistical learning model and to select the features simultaneously. For example, the idea of the Least Absolute Shrinkage and Selection Operator (LASSO) is to train a linear regression model and to select the features at the same time [Tib96]. To do so, by zeroing the coefficients of the features thanks to the ℓ_1 norm, the algorithm removes a number of features. The objective function is as follows:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{m}\beta_i x_{ij}\right)^2 + \lambda\sum_{j=1}^{m}\|\beta_j\|_1, \tag{2.2}$$

where x_i is the i^{th} observation vector in m dimensions and y_i is the corresponding response, β_j is the coefficient of x_j, and λ is the tuning parameter [Jam+14].

In terms of efficiency, this group of methods is between the Filter Methods and the Wrapper Methods [GE03].

2.2 Clustering

Clustering is the task of grouping n-dimensional points by similarity, so that the similarity between the points in the same cluster is high and the similarity between distinct clusters is low.

Clustering is a classic example of unsupervised learning.

The quality of predicted labels depends on the setting of the algorithm. If we alter the parameters, such as number of clusters and measure of affinity, we may get a better or worst result.

There is a variety of algorithms in the field of clustering. The following describes perhaps the best-know algorithms: k-means and hierarchical clustering. Additionally, Section 2.2.3 gives a description of Spectral Clustering.

2.2.1 *k*-Means

k-means is an easy way to divide n points into k clusters [Ber02].

The algorithm starts by assigning a label between 1 and k to each point. The steps are as follows:

- *Step 1.* Update the coordinates of the centroids. The centroid of the i^{th} cluster is the mean of the points in the i^{th} cluster.
- *Step 2.* Assign the label of the closest cluster to each point.

The algorithm repeats Step 1 and Step 2. When the labels stop changing, the convergence is reached and the algorithm stops.

One disadvantage of this algorithm is that it supposes that the clusters are globular – in other words, the distribution of points is normal and thus the clusters are linearly separable [LRU14].

2.2.2 Agglomerative Clustering

The most common type of Hierarchical Clustering is Agglomerative Clustering.

Let n be the number of points.

The algorithm starts from n clusters of size 1 each. Firstly, it calculates the pairwise distance between the clusters. Secondly, it finds the 2 clusters closer to each other, then it merges the two clusters together. These two steps are repeated again and again. When all the n points are in a single cluster, the algorithm stops.

Divisive Clustering works in the other way around.

Dendrograms are an easy way to visualize hierarchies of clusters. See Figure 2.1.

2.2.3 Spectral Clustering

Ng, Jordan, and Weiss [NJW01] proposed the state-of-the-art algorithm when it comes to Clustering.

Let k be the number of clusters. Let n be the number of points.

The input data is in the shape of a graph, so the algorithm starts from an adjacency matrix representing a graph. The number of nodes in the graph is equal to n. The weight of the edge between i and j is equal to the element of the adjacency matrix at the intersection of the i^{th} row and the j^{th} column.

The first step is to find the k largest eigenvectors of the normalized Laplacian matrix. The normalized Lapalacian matrix is as follows:

$$L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \tag{2.3}$$

where A is an adjacency matrix and D is a diagonal matrix, where $D_{ii} = \sum_{j=1}^{n} A_{ij}$. All these eigenvectors span a k-dimensional space.

The algorithm groups the eigenvectors via k-means, then it assigns the same labels as the eigenvectors to the points.

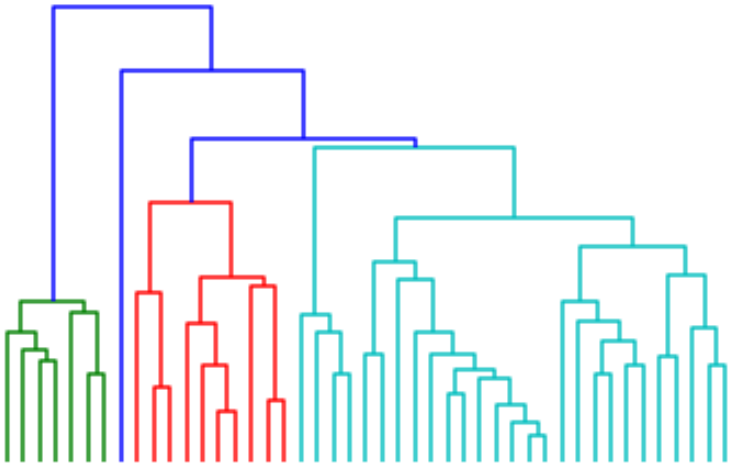


Figure 2.1: Example of a Dendrogram.

According to Ng, Jordan, and Weiss [NJW01], we clustered the k largest eigenvectors of the normalized Laplacian matrix, where k is the number of clusters, but in this particular case we minimize the conductance. Conductance is the number of edges departing from the community over the minimum of the number of edges in the community and the number of edges in the rest of the graph [For10].

So as to minimize the expansion, we should have clustered the k largest eigenvalues of the Laplacian matrix. Expansion is the number of edges per node leaving from the community [YL15].

In order to maximize the average degree, we should have clustered the k largest eigenvectors of the adjacency matrix. The average degree is the mean degree of the nodes in the community [YL15].

2.3 Clustering Performance Evaluation

When the true labels are unknown, it is not possible to calculate the error, such as the training error and the test error.

Fortunately, in case of unsupervised learning, there is a large variety of metrics to choose from.

2.3.1 Silhouette

In terms both of density and of separation, the average silhouette score is a measure of the quality of a model. Silhouettes are an easy way to make an evaluation of the

clustering performance.

The silhouette score of $i \in A$ is as follows:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{2.4}$$

where $a(i)$ is the average distance between i and all other points of A and $b(i)$ is the minimum average distance between i and all points of $C \neq A$ [Ros87].

The possible values range from -1.0 to 1.0. When groups are dense and well separated the silhouette score tends to 1.0.

2.3.2 Modularity

Yang and Leskovec [YL15] list a number of clustering performance metrics. Some are based on internal connectivity. Some are based on external connectivity. Some are based both on internal connectivity and on external connectivity. Metrics such as modularity are based on the structure of the graph. It is worth remembering that the problem of clustering is equivalent to the problem of community detection. As a result, it is possible to see the clusters as the communities. Thus, by studying the community structure of the graph, the clustering performance evaluation is made.

The difference between the number of edges on the inside of the communities and the expected number of edges in the random graph is the modularity [New06]. That is to say, the modularity is the density of the edges within the communities versus the density of the edges without the communities [Blo+08]. Basically, the modularity compares the community structure of the graph with the community structure of the random graph.

Modularity is as follows:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j), \tag{2.5}$$

where

- A_{ij} is the weight of the edge between $i \in c_i$ and $j \in c_j$,
- $k_i = \sum_j A_{ij}$,
- if c_i and c_j are both equal to each other, then $\delta(c_i, c_j)$ is equal to 0, else $d(c_i, c_j)$ is equal to 1, and
- $m = \frac{1}{2} \sum_i \sum_j A_{ij}$.

The possible values range from -1.0 to 1.0. The modularity of random graphs is equal to 0.0 If the partition of the nodes in the graph is of low quality, then the modularity is low. The modularity is high if the partition of the nodes in the graph is of high quality. That is to say, when the modularity is high, then the graph has a community structure by comparison with the random graph with the same characteristics.

2.4 Distance

The pairwise distance between n-dimensional points is required for clustering. Usually the pairwise distance between n-dimensional points is in the shape of a matrix.

According to Leskovec, Rajaraman, and Ullman [LRU14], the properties of distance are as follows:

- $d(x, y) \geq 0$,
- $d(x, x) = 0$,
- $d(x, y) = d(y, x)$, and
- *Triangular Inequality.* $d(x, z) \leq d(x, y) + d(y, z)$.

The following will explain the cosine distance, the euclidean distance, and the Pearson's distance to the reader.

2.4.1 Cosine Distance

The cosine distance between two points is equal to the cosine of the complementary angle between the points. The distance between u and v is as follows:

$$1 - \frac{u \cdot v}{\|u\|_2 \|v\|_2} = 1 - \frac{u \cdot v}{\sqrt{\sum_{i=1}^{n} u_i^2} \sqrt{\sum_{i=1}^{n} v_i^2}}. \tag{2.6}$$

The possible values range from 0.0 to 2.0.

The cosine similarity is not proportional to the magnitude of the vector representative of the n-dimensional point. For example, the use of cosine distances has already become widespread when it comes to calculating the similarity between two documents; indeed, we want the distance between the document to be independent of their magnitude, because it might be desirable to attach importance to short and to long documents in the same way.

2.4.2 Euclidean Distance

The euclidean distance between two points is equal to the length of the segment between the points. The distance between two points is as follows:

$$\|u - v\|_2 = \sqrt{\sum_{i=1}^{n} (u_i - v_i)^2}, \tag{2.7}$$

where u is a n-dimensional point and v is a n-dimensional point.

The possible values range from 0.0 to ∞.

Apart from the direction of the vector, the euclidean distance depends on the magnitude of the vector, unlike the cosine distance; this is why the upper bound is equal to ∞ instead of 2.0.

2.4.3 Pearson's Distance

The Pearson's correlation coefficient is a measure of the correlation between two variables, i.e., x and y. The Pearson's correlation coefficient is as follows:

$$\frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}}, \tag{2.8}$$

where μ_x is the mean of x and μ_y is the mean of y.

The possible values range from -1.0 to 1.0. When the correlation coefficient is equal to -1.0, the correlation between x and y is negative. That is to say, if x goes down, then y goes up, and the other way around. If the correlation coefficient is equal to 1.0, the the correlation between x and y is positive. As a result, both variables go down at the same time, and the other way around.

The Pearson's correlation coefficient is equal to the cosine similarity between u and v when u and v are standardized.

The Pearson's distance is as follows:

$$1 - \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}}. \tag{2.9}$$

The possible values range from 0.0 to 2.0.

2.5 Classification of Breast Cancer

Perou et al. [Per+00] were the first to classify the breast cancer sufferers by quantity of messenger RNA. There are five categories of breast cancer cell: *basal-like*, *HER2*, *luminal A*, *luminal B*, and *normal-like* breast cancer cells. The five groups of cancer sufferers are quite distinct from each other. Unfortunately, the usual cancer treatments regulate the amount of protein, not the quantity of messenger RNA.

Johansonn et al. [Joh+19] approached the problem in a different way. Rather than group their patients by quantity of messenger RNA, the authors of the article classified the cancer sufferers by protein content into five different groups. Firstly, $9,995$ proteins were selected for use in hierarchical clustering. The results of the experiment were not promising. Secondly, 37 proteins known to cause the onset of breast cancer were hand picked. Even though protein selection was based on domain expertise, the clustering was incomplete again. As a matter of fact, a cluster of type *basal-like* samples, a cluster of type *luminal A* samples, and a cluster of type *normal-like* samples distanced themselves from the large cluster of remaining samples. It is worth remembering that the number of samples was small. In addition, the authors of the article divided further the class of type *basal-like* samples and the class of type *luminal B* samples.

[Alb+07], [DHH09], and [Pen+03] employ Feature Selection to tackle the problem of the classification of cancer. Peng at al. [Pen+03] propose a classifier able to recognize cancer types. The main focus of Alba et al. [Alb+07] and Duval et al.

[DHH09] is the detection of cancer sufferers among the samples, but in this particular case the authors experiment on data sets on several cancer types, such as breast cancer and lung cancer. Whilst Duval et al. [DHH09] propose an embedded method, Alba et al. [Alb+07] and Peng at al. [Pen+03] propose a wrapper method; they share the same classifier, i.e., Support Vector Machine, however. Furthermore, both Alba et al. [Alb+07] and Peng at al. [Pen+03] make use of evolutionary algorithms, such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO).

Chapter 3

Research Methodology

It is worth remembering that we need to address both the problem of Candidate Generation and the problem of Candidate Evaluation.

We developed a couple of pipelines. Whilst the first focuses its attention on the breast cancer data, the first takes advantage both of the external data on protein complexes and of the internal data on gene products.

The following will explain the pipelines.

3.1 Input Data

3.1.1 Internal Data

The input data is a collection of $n = 45$ observation vectors in $m = 9,995$ dimensions. Whilst a sample is a patient, a feature is a gene product.

The initial list of features is comprised of $12,890$ gene products; however, $2,895$ gene products out of $12,890$ were removed owing to large error.

The possible values range from 0.0 to ∞. The mean protein content is equal to 1.0 because they were so measured. As a result, we calculate the base-2 logarithm of the internal data. Basically, since the distribution of values is right skewed, we are interested in relative distances, not absolute distances. We want the distribution of values to be normal. See Figure 3.1.

3.1.2 External Data

Proteins are manufactured from amino acids. The messenger RNA gives the cell instructions on how to assemble them. For this reason, it might be desirable to classify the cancer sufferers by amount of protein instead of quantity of messenger RNA.

A protein complex is a complex system composed of proteins. All these proteins collaborate on a biological function. They play a leading role in the inner workings of the human body, seeing as how they are complicated.

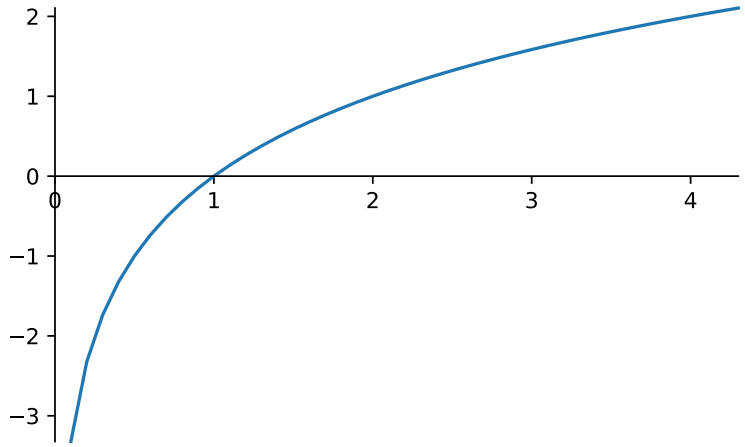


Figure 3.1: $y = log_2(x)$.

CORUM [Giu+18] is a database of protein complexes. The database contains $4,274$ records. All these records are of high quality, because all $4,274$ protein complexes are experimentally verified. $2,916$ records out of $4,274$ records are human protein complexes. The fields we are interested in are as follows:

- Complex Name,
- Organism, and
- Subunits.

To put it simply, the gene products listed in the last field are part of the protein complex identifiable from the first field.

3.2 Pipeline A

The first pipeline makes the most of the breast cancer data kindly offered by the laboratory.

3.2.1 Candidate Generation

The whole idea of the first step is to generate communities of gene products in a smart way. That is to say, we want to test the most promising candidates, not all the combinations.

Unfortunately, we do not have a clue about the categories of gene products. For this reason, we choose to cluster the features. Indeed, clusters of features are, by definition, subsets of features – in other words, candidates.

We select Spectral Clustering, because it is considered state-of-the-art. This is a general algorithm able to recognize clusters of arbitrary shape. As described in Section 2.2.3, the requirements are as follows:

- an adjacency matrix and
- a number of clusters.

An adjacency matrix represents a graph. The weight of the edge between two nodes in the graph, i.e., i and j, is equal to the element of the adjacency matrix at the intersection of the i^{th} row and the j^{th} column. The thing is, we need to convert the breast cancer data into a graph.

Section 4.1.1 describes how we select the number of clusters.

The whole idea behind the project is to share the list of gene products suspected to have a part in the onset of breast cancer with Karolinska Institutet. Since the experts have to investigate the most promising gene products, we want the candidates to be small enough. For this reason, we divide the large candidates in a recursive fashion with the aid of Spectral Clustering and Branch and Bound. Basically, in order to bind the searching space, we only partition the promising candidates if necessary. The remaining candidates are ignored, even though they are large.

Gene Product Graph

Let G be an empty graph. We add a node for all $9,995$ gene products. The weight of the edge between two gene products is equal to the absolute value of the cosine of the angle between the gene products. The affinity matrix is as follows:

$$\mathit{affinity\_matrix} = |1 - \mathit{dist\_matrix}|, \tag{3.1}$$

where *dist_matrix* is the collection of pairwise cosine distance between the $9,995$ gene products in the shape of a matrix.

The possible values range from 0.0 to 1.0. See Figure 3.3.

At the suggestion of the biologists, we select the absolute value of the cosine. The hypothesis is that the abnormal behavior of gene products is the cause of the onset of breast cancer. Basically, the unexpected presence as well as absence of gene products can affect the equilibrium between the various parts of the body. As a result, we are interested in anomaly detection. From a purely practical point of view, we want to find both the protein contents below the mean and the protein contents above the mean.

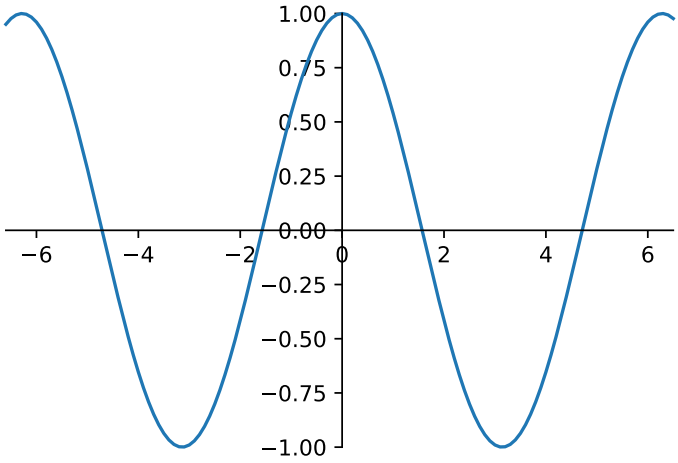


Figure 3.2: $y = cos(x)$

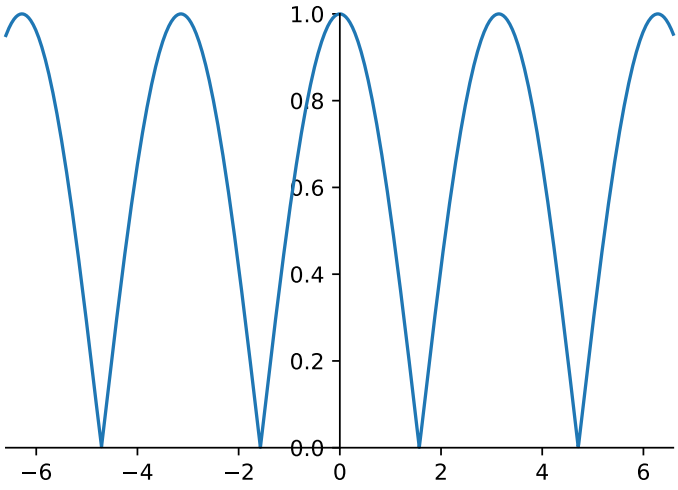


Figure 3.3: $y = |cos(x)|$

3.2.2 Candidate Evaluation

At this point, we need to address the problem of Candidate Evaluation. That is to say, we have to test the candidates.

This step will determine which candidates are good at characterizing the patients. That is to say, we quantify how good or bad the gene products in the n^{th} candidate are at characterizing the patients.

We are interested in generating communities of homogeneous patients. Whilst all these patients characterized by high protein content need to be on the one side, all these patients characterized by low protein content need to be on the other side.

Firstly, we create a patient graph according to the gene products in the n^{th} candidate, then we bisect the patient graph by means of Spectral Clustering once again. Basically, we read the rows of data connected to the gene products in the n^{th} candidate. All these values are used for creating the corresponding patient graph. Secondly, we calculate the modularity of the bisection of the patient graph, then we compare the modularity with the threshold. The number of clusters is unknown in this case too.

We choose to bisect the patient graph just to be on the safe side. We see no reason for us to select a number of clusters greater than 2 because we suppose that the gene products behave either well or badly. That is to say, we do not want to risk to divide a homogeneous cluster of patients. If necessary, we can always divide the clusters one more time. Furthermore, in the majority of cases, the largest eigengap is between the 1^{st} eigenvalue and the 2^{nd} eigenvalue, which means that the bisection may well be the best choice. It is worth remembering that according to Perou et al. [Per+00], there are five categories of breast cancer cells. However, we want the number of clusters to be equal to 2 instead of 5 because the input data is on the subject of gene products, not messenger RNA.

Patient Graph

Let G be a weighted graph. There are 45 nodes in this graph. Basically, a node is a patient. The weight of the edge between two patients is equal to the cosine of the angle between the two observation vectors representing the patients. We have increased the value of affinity by 1, because we want the minimum weight to be greater than 0.0. The affinity matrix is as follows:

$$affinity\_matrix = 2 - dist\_matrix, \tag{3.2}$$

where $dist\_matrix$ is the collection of pairwise cosine distance between the 45 patients in the shape of a matrix.

The possible values range from 0.0 to 2.0. See Figure 3.4. Intuitively, if the affinity between two patients tends to 2.0, then in terms of protein content, these two patients will be similar to the other one, and the other way around.

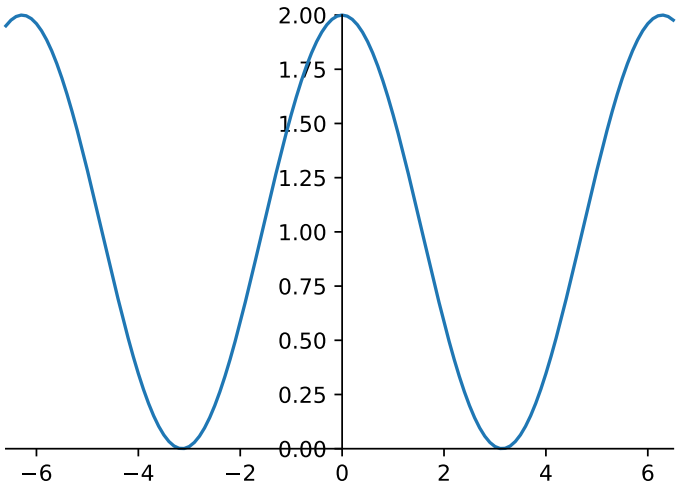


Figure 3.4: $y = 1 + cos(x)$

Modularity

We opted in favor of modularity in order that we may measure the quality of the bisection of the patient graph.

Modularity is one of the most popular measures of clustering performance evaluation. Modularity has three strengths according to Fortunato [For10].

- Modularity includes a definition of "community".
- Modularity makes a comparison between the graph and the random graph.
- Modularity calculates the quality of the community.

Intuitively, a community is a subgraph such that the density of the edges within is greater than the density of the edges without. It is worth remembering that the modularity of the random graph is equal to 0.0.

As a rule of thumb, the modularity of the graph is the sum of the values of modularity of the clusters of samples. However, we are interested in the best group of patients in this case, hence the choice to return the maximum value of modularity of the clusters of samples.

Modularity has been adopted as the measure of clustering performance because it meets our requirements. Unfortunately, it is not always possible to assert that the one community is better than the other community in terms of modularity.

Let k be the number of clusters. The modularity matrix is as follows:

$$\begin{matrix} q_{11} & q_{12} & \cdots & q_{1k} \\ q_{21} & q_{22} & \cdots & q_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ q_{k1} & q_{k2} & \cdots & q_{kk} \end{matrix},$$

where q_{11} is the value of modularity of the first cluster, q_{22} is the value of modularity of the second cluster, and so on.

In case of bisection, the modularity matrix is as follows:

$$\begin{matrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{matrix},$$

where q_{11} is the value of modularity of the first cluster and q_{22} is the value of modularity of the second cluster. The sum both of all rows and of all columns of the modularity matrix is, by definition, equal to 0.0.

$$\begin{cases} q_{11} + q_{12} = 0.0 \\ q_{21} + q_{22} = 0.0 \end{cases} \tag{3.3}$$

$$\begin{cases} q_{11} + q_{12} = 0.0 \\ q_{21} + q_{22} = 0.0 \end{cases} \tag{3.4}$$

If Equation 3.3 and Equation 3.4 are true, then q_{11} is equal to q_{22}. That is to say, in terms of modularity, the two clusters are identical. So, there need to be at least two distinct metrics for us to determine which cluster is better. For example, one could opt for metrics based on internal and/or external connectivity, such as average degree and conductance, plus modularity.

Threshold

The baseline is a hand-picked set of 37 gene products know to cause the onset of breast cancer. At the present time, all these gene products are used for classifying the breast cancer cells. As a matter of fact, it is possible to recognize three categories of breast cancer cells: *basal-like*, *luminal A*, and *normal-like* breast cancer cells. See Figure 3.5. The threshold is equal to the modularity of the bisection of the patient graph created according to the baseline.

3.3 Pipeline B

The second pipeline takes advantage both of the internal data on gene products and of the external data on protein complexes.

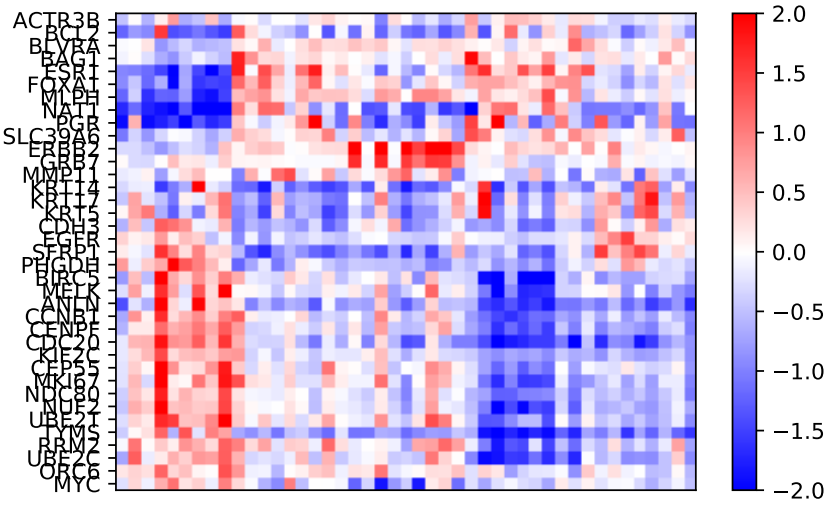


Figure 3.5: Heatmap of the Protein Content. Whilst the horizontal axis shows the patients, the vertical axis shows the gene products of the baseline.

3.3.1 Candidate Generation

The first step of the pipeline is different in this case.

Rather than create the gene product graph, we create an empty bipartite graph, i.e., B. There are two sets of nodes in this bipartite graph. The node set of protein complexes is on the one side. The node set of gene products is on the other side. The only edges in the bipartite graph are between gene products and protein complexes.

There are $9,995$ gene products in this graph and only $2,916$ protein complexes. If a protein is part of a protein complex, then we add an edge between the nodes.

To create a protein interaction network, we return the projection of B onto the gene products. On the one hand we want the weight of the edge between two gene products to be proportional to how many common neighbors they share, but on the other hand we want to penalize high-degree protein complexes because a high-degree protein complex does not add as much information as a low-degree protein complex. To do so, we return the projection of B, as described in Newman [New01]. Let $i \neq j$ be two nodes in the projection of B. Let k be a protein complex. The weight on the edge between i and j is as follows:

$$w_{ij} = \sum_k \frac{\delta_i^k \delta_j^k}{n_k - 1}, \tag{3.5}$$

where

- if the edge between i and k is in the bipartite graph, then δ_i^k is equal to 1, else δ_i^k is equal to 0,
- if the edge between j and k is in the bipartite graph, then δ_j^k is equal to 1, else δ_j^k is equal to 0, and
- n_k is the degree of k.

If n_k is equal to 1.0, then we ignore it. For example, there are two gene products, i.e., i and j, in this graph and only one protein complex, i.e., k. If n_k is equal to 1.0, then it is not possible that our gene products share a common neighbor.

Once we have created the protein interaction network, we will generate connected components. The connected components of the graph include nodes such that the nodes are reachable from each other. Since a connected component is already a subset of features, we add the connected components to the list of candidates as well.

We can expect to see a few connected components to be large. As a consequence, we recursively split the large connected components in the same way as we did before.

3.3.2 Candidate Evaluation

Section 3.2.2 describes how to assert our candidate is either good or bad.

Firstly, we calculate the modularity of the bisection of the patient graph created according to the gene products in the candidate under evaluation. Secondly, we benchmark the candidate against the baseline in terms of modularity. If the candidate scores worst than the baseline, then we will discard it, else we will keep it.

Chapter 4

Results

4.1 Pipeline A

4.1.1 Candidate Generation

As described in Section 3.1.1, firstly, we need to address the problem of Candidate Generation.

The input data is in the shape of an adjacency matrix.

Obviously, the number of candidates, or clusters, is unknown. It may be that the presence of a large gap between two consecutive eigenvalues will suggest the true number of clusters. So, we plot the eigenvalues of the Laplacian matrix, starting with the adjacency matrix of the gene product graph. The Laplacian matrix is as follows:

$$L = D - A, \tag{4.1}$$

where A is an adjacency matrix and D is a diagonal matrix, where $D_{ii} = \sum_{j=1}^{n} A_{ij}$. Unfortunately, there is not a large eigengap. See Figure 4.1. As a result, we choose a variable number of clusters. The possible values range from 3 to 15. On the bright side, overlapping clusters of gene products are made possible by a variable number of clusters.

The majority of candidates are large, hence the additional splits we have done. We have repeated the step again and again. In the end, the size of candidates is smaller than 62.

It is interesting to note well that 12 gene products of the baseline out of 37 are in the final candidates.

4.1.2 Candidate Evaluation

Section 3.2.2 describes how we discarded some bad candidates.

Fist of all, we have to set the threshold. To do so, we calculate the modularity of the bisection of the patient graph when the candidate is the baseline. The threshold is equal to 0.03. It is worth remembering that the modularity of the random graph is

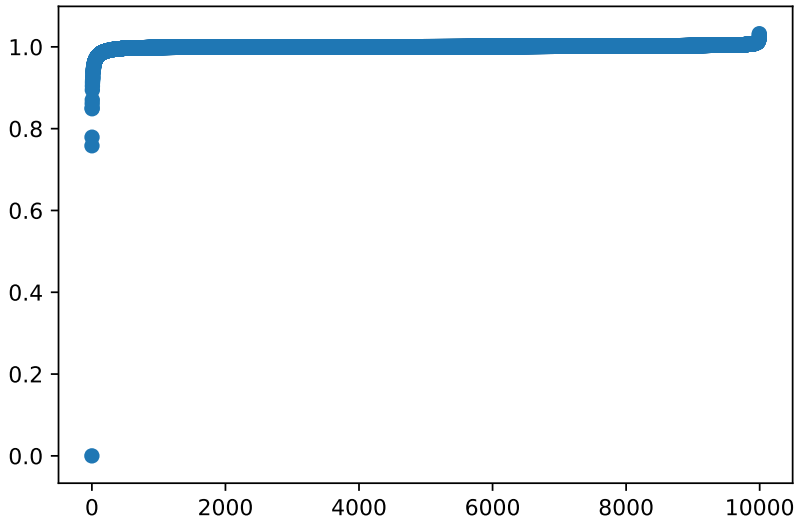


Figure 4.1: Eigenvalues of L.

equal to 0.0. Also, thanks to the threshold, we mark the boundaries of the searching space by means of branch and bound.

The candidates number over a hundred after the first iteration. The next iteration produces $11,734$ candidates. The final number of candidates is equal to $707,098$.

The vast majority is above the threshold. See Figure 4.2.

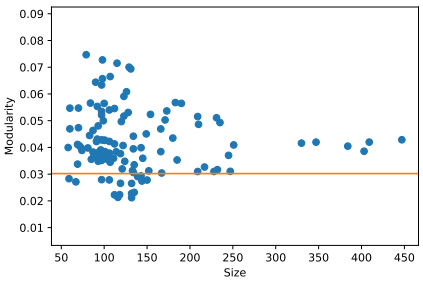
4.2 Pipeline B

4.2.1 Candidate Generation

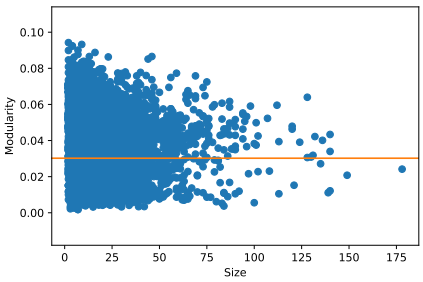
As described in Section 3.1.2, we created an unweighted bipartite graph, i.e., B. There are $9,995$ gene products in this graph and only $2,916$ protein complexes. See Table 4.1.

The average degree of the nodes in the graph is equal to 1.6. Since the minimum degree is equal to 0.0, there need to be at least one node isolated. That is to say, an isolate is a node without neighbors. If we remove all isolates in the graph, then the number of nodes is equal to $5,676$, not $12,911$. It is possible that a gene product will be part of one protein complex or more. As a matter of fact, the mean degree of the gene products in the graph is equal to 6.5.

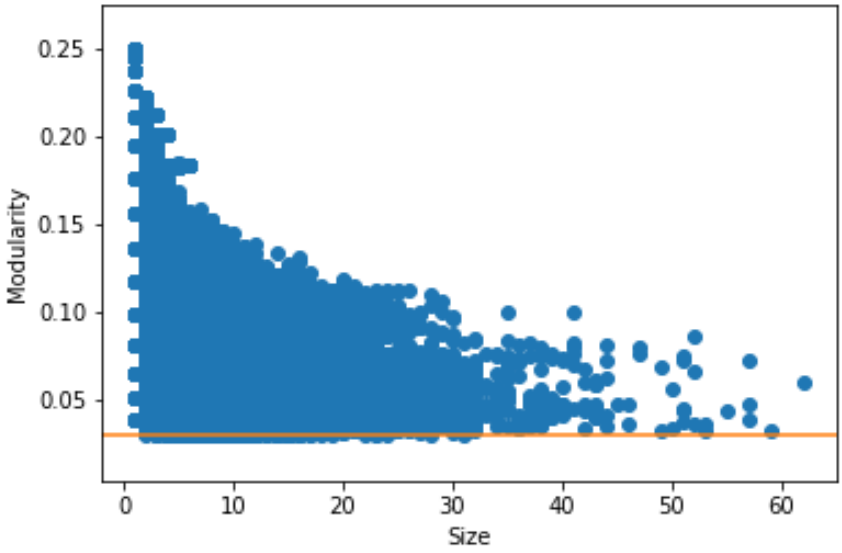
As described in Section 3.1.2, we created a weighted graph, i.e., G. The number of nodes in this graph is equal to $9,995$; as a matter of fact, the number of gene products is equal to $9,995$. See Table 4.1. However, $7,170$ nodes out of $9,995$ are



(a) 1st Iteration



(b) 2nd Iteration



(c) 3rd Iteration

Figure 4.2: Modularity is on the vertical axis, and the size of the candidate is shown on the horizontal axis.

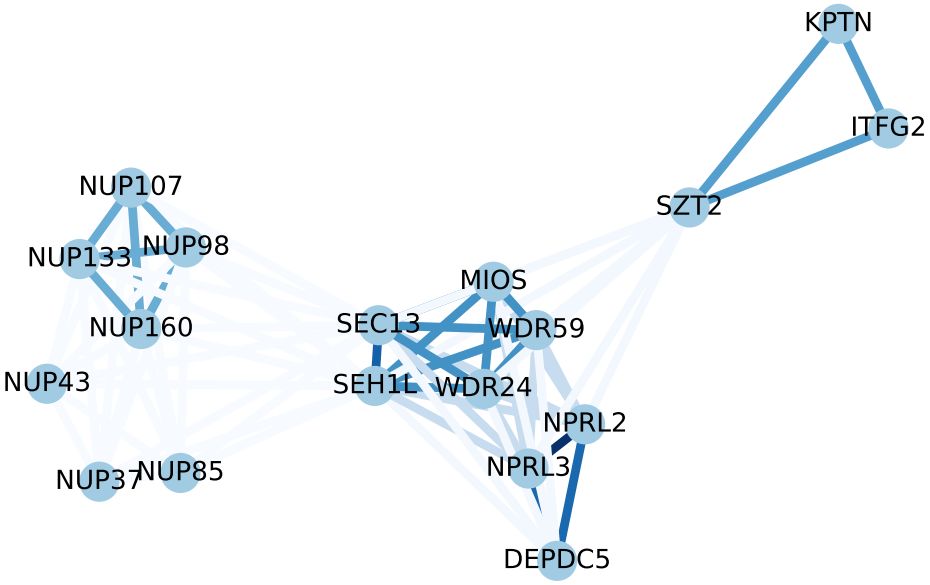


Figure 4.3: Third-Largest Connected Component of G.

isolates.

The mean degree is equal to 6.5. Whilst the minimum weight tends to 0.0, the maximum weight is equal to 15.1.

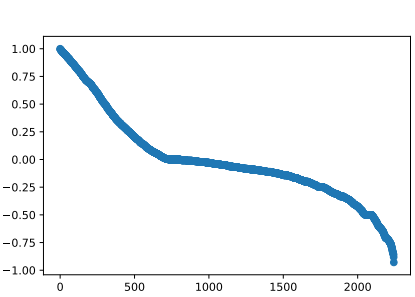
We generated connected components. There are $7,326$ connected components in this graph. The size of the largest connected component and the size of the second-largest connected component are equal to $2,244$ and 74, respectively. See Table 4.1 and Table 4.2.

All 37 gene products of the baseline are in the graph. The largest connected component contains 14 gene products out of 37, whereas 19 gene products out of 37 are isolates.

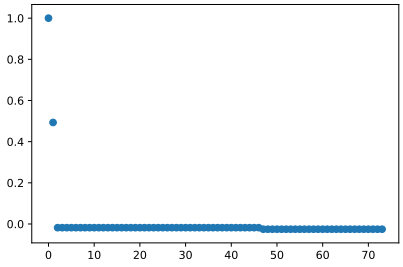
Both the largest component and the second-largest connected component are too large for the experts to investigate. We plot the eigenvalues of the normalized Laplacian matrix because the number of clusters is unknown once again. The normalized Laplacian matrix is as follows:

$$L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \tag{4.2}$$

where A is an adjacency matrix and D is a diagonal matrix, where $D_{ii} = \sum_{j=1}^{n} A_{ij}$. As far as the largest connected component is concern, there is not a large eigengap. See Figure 4.4a. Thus, we choose a variable number of clusters. The possible values range from 3 to 15. If it comes to the second-largest connected component, we calculate the eigenvalues of the normalized Laplacian matrix again, but in this particular case there is a large eigengap between the 1^{st} eigenvalue and the 2^{nd}



(a) Largest Connected Component of G



(b) Second-Largest Connected Component of G

Figure 4.4: Eigenvalues of L.

Table 4.1: Number of Nodes, Number of Edges, and Number of Connected Components of B and G.

Graph	Number of Nodes	Number of Edges	Number of Connected Components
G	$12,911$	$10,506$	$7,476$
B	$9,995$	$32,491$	$7,326$

eigenvalue. See Figure 4.4b. As a result, we bisect the second-largest connected component in this case.

4.2.2 Candidate Evaluation

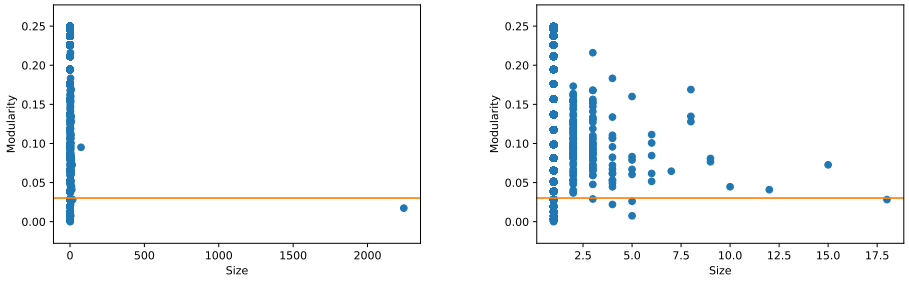
In terms of modularity, the majority of connected components rank above the baseline.

The size seems to have a weak influence over the modularity. Apparently, the only thing that matters is that the relevant gene products are actually selected. However, the high-performing candidates are small connected components. Surprisingly, some isolates are eight times better than the baseline.

Let `{GINS2, GINS1, GINS4, GINS3}` be a candidate. We plot the corresponding four rows of data in order that we may double-check results. See Figure 4.6. The one cluster of patients is on the left. The other cluster of patients is on the right. As expected, there is a split between high values on the left and low values on the right. As a matter of fact, the value of modularity is equal to 0.18 in this case.

Figure 4.7 shows that there is not a clear distinction between the cluster of patients on the left and the cluster of patients on the right when the value of modularity is less than 0.03.

Furthermore, the results of the recursive division both of the largest connected component and of the second-largest connected component are good. See Figure 4.8 and Figure 4.9. For example, the largest connected component is below the baseline;



(a) (b)

Figure 4.5: Connected Components of G.

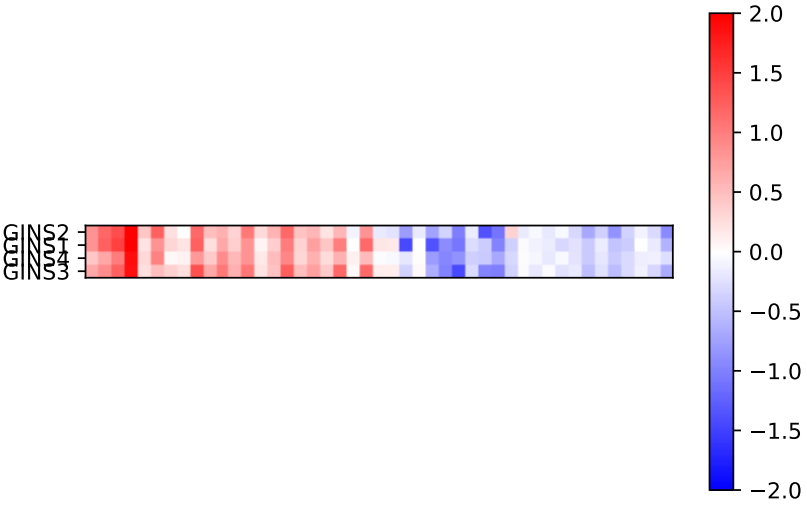


Figure 4.6: `{GINS2, GINS1, GINS4, GINS3}`

Table 4.2: Size of the Connected Components of G.

Number of Nodes	Number of Connected Components
2,244	1
74	1
18	1
15	1
12	1
10	1
9	2
8	3
7	1
6	5
5	7
4	16
3	42
2	74
1	7,170

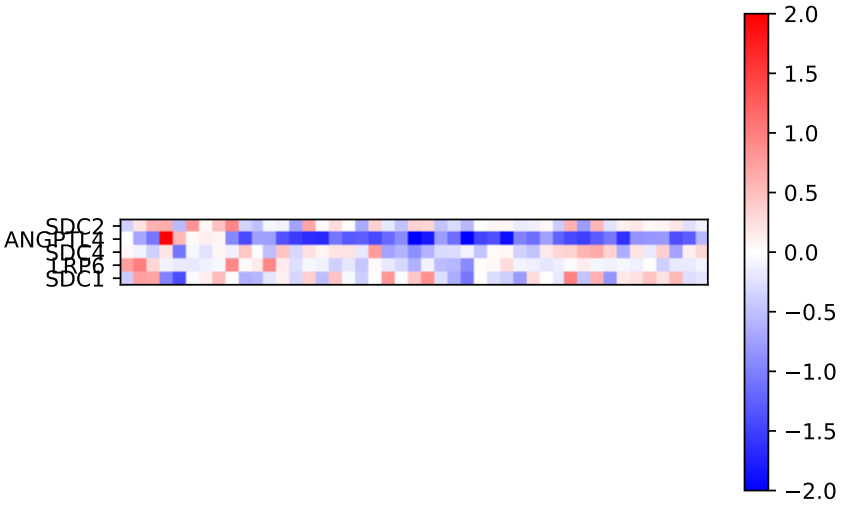
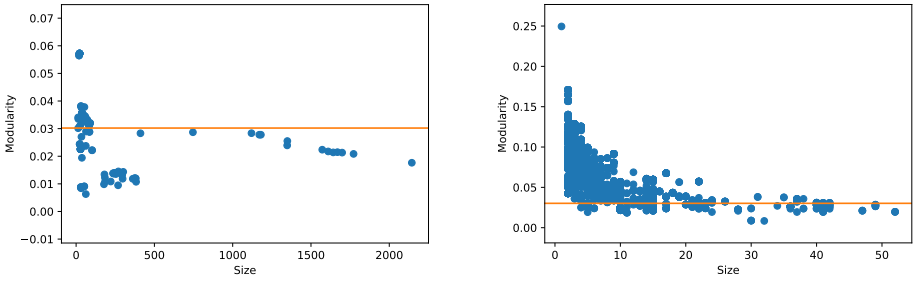


Figure 4.7: `{SDC2, ANGPTL4, SDC4, LRP6, SDC1}`



(a) 1st Iteration (b) 2nd Iteration

Figure 4.8: Largest Connected Component of G.

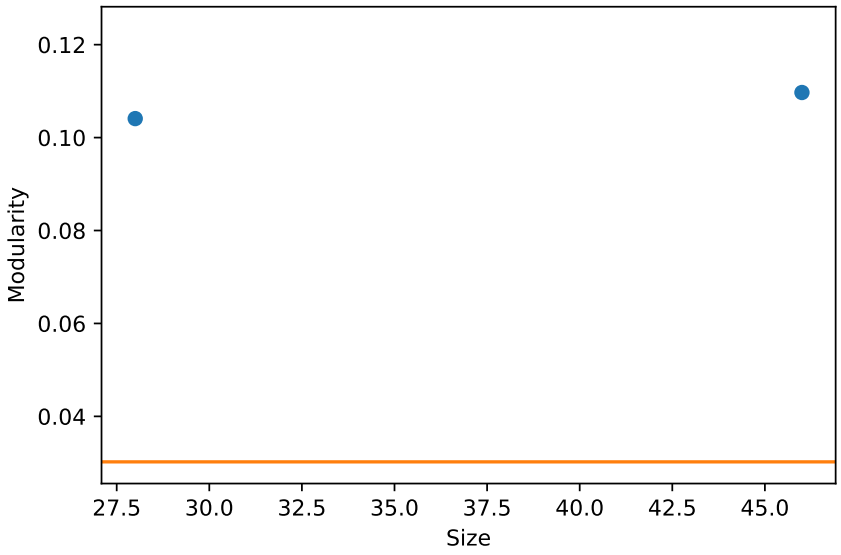


Figure 4.9: Second-Largest Connected Component of G.

however, several clusters of gene products in the largest connected component are above the baseline, regardless.

Chapter 5

Conclusion

Karolinska Institutet posed the question of whether it is possible to find gene products guilty of causing the onset of breast cancer so as to group the cancer sufferers by protein content, and thus recognize the presence of many different breast cancer types. Our understanding of the inner workings of the human body is limited. We do not have a clue what a number of gene products do. So, we resolved the issue in a domain expertise–agnostic fashion.

We developed a new method in the category of wrapper methods. Firstly, by partitioning the protein interaction networks available via Spectral Clustering, we tackled the problem of how to generate candidates. Secondly, we addressed the problem of performance evaluation by comparing the gene products to those known to be the cause of the disease in terms of the value of modularity of the bisection of the collection of patients.

This experiment demonstrates that there is a number of gene products up to eight times better at representing the groups of breast cancer sufferers than the gene products hand-picked by the experts. Results show that the absence of domain expertise is not an obstacle to the feature selection process.

Overall, it is possible to give a positive answer to the research question.

5.1 Discussion

As described in Chapter 3, we developed a pair of pipelines. There is a difference in number of candidates between the first and the latter. In terms of modularity, these two pipelines are similar to the other one. The possible values range from approximately 0.00 to roughly 0.25. Note that the size of the high-performing candidates is surprisingly small and furthermore the majority does not overlap the set of gene products hand-picked by the experts.

The candidates here number over a million. Next, we will list the top gene products in order that biologists may investigate whether these gene products are the cause of the onset of the disease or not.

5.1.1 Domain Expertise–Agnosticism

It is worth remembering that one of the goals of this project is to require no domain expertise at all.

On the one hand, the first step of the first pipeline is the beneficiary of the affinity matrix at the suggestion of the biologists at Karolinska Institutet. On the other hand, the idea of the first step of the second pipeline is to derive the protein interaction network from the external data, but in this particular case the weight of the edges in the graph has been assigned by the projection of the bipartite graph. See Equation 3.3.

However, there are not so many ways to calculate the affinity matrix and furthermore the external data is considered a source of domain expertise in case of proteomics, not oncology.

5.2 Future Work

5.2.1 Internal Data

First of all, we plan to carry out an analysis of lung cancer data. It is likely that it will be harder to choose the threshold, because lung cancer is one of the lesser-known cancer types. However, it is likely that we will benefit from the larger number of patients available.

Furthermore, we are already planning how to test the wrapper method to see if it works in case of non-biological data. It is always interesting to include both categorical features and numerical features, if possible.

5.2.2 External Data

To put it simply, the whole idea of the degree project is to extract some candidates from the protein interaction networks available. For this reason, it might be desirable to add new external data sets.

We attempted to make use of the Gene Ontology [Ash+00] [Con19] in the first few weeks. Unfortunately, we failed in our attempt to use the ontology as a protein interaction network. Nevertheless, we are still interested in extracting a protein interaction network from the Gene Ontology. The Gene Ontology is one of the largest sources of biological data; as a matter of fact, the number of citations of [Ash+00] is huge.

5.2.3 Clustering

As described in Section 1.6, the project is limited by the boundaries of the non-overlapping clusters of patients. For this reason, we plan to test the algorithms able to generate overlapping communities of samples.

As described in Section 2.2.3, we clustered the k largest eigenvectors of the normalized Laplacian matrix, where k is the number of clusters. Ideally, each gene

product will be as similar as possible to the other one in the same group when it comes to partitioning the gene product graph, and it is the same with the patients in our minds. For this reason, we believe that we will see even better results by finding the k largest eigenvectors of the adjacency matrix instead of the normalized Laplacian matrix.

5.2.4 Forward Stepwise Selection

The majority of candidates are small and furthermore the best-performing candidates are extra small. It is always interesting to merge two small candidates into one in order to see if the value of modularity increases. It might be desirable to generate candidates via Forward Stepwise Selection.

Bibliography

[Fer+18] Jacques Ferlay et al. *Global Cancer Observatory: Cancer Today.* [Online; accessed Jun, 12, 2019]. 2018. URL: https://gco.iarc.fr/today.

[Joh+19] Henrik J. Johansonn et al. "Breast cancer quantitative proteome and proteogenomic landscape". In: *Nature* 10 (2019). DOI: 10.1038/s41467-019-09018-y.

[Per+00] Charles M. Perou et al. "Molecular portraits of human breast tumors". In: *Nature* 406 (2000), pp. 747–752. DOI: 10.1038/35021093.

[Giu+18] Madalina Giurgiu et al. "CORUM: the comprehensive resource of mammalian protein complexes 2019". In: *Nucleic Acids Research* (2018). DOI: 10.1093/nar/gky973.

[Cun08] Pádraig Cunningham. "Dimension Reduction". In: *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval.* 2008, pp. 91–112. DOI: 10.1007/978-3-540-75171-7_4.

[HCN05] Xiaofei He, Deng Cai, and Partha Niyogi. "Laplacian Score for Feature Selection". In: *Proceedings of the 18th International Conference on Neural Information Processing Systems.* 2005, pp. 507–514.

[Jam+14] Gareth James et al. *An Introduction to Statistical Learning. with Applications in R.* 2014.

[LY05] Huan Liu and Lei Yu. "Toward integrating feature selection algorithms for classification and clustering". In: *IEEE Transactions on Knowledge and Data Engineering* 17.4 (2005), pp. 491–502. DOI: 10.1109/TKDE.2005.66.

[Tib96] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288.

[GE03] Isabelle Guyon and André Elisseeff. "An Introduction to Variable and Feature Selection". In: *Journal of Machine Learning Research* 3 (2003), pp. 1157–1182.

[Ber02] Pavel Berkhin. *A Survey of Clustering Data Mining Techniques.* Tech. rep. 2002.

[LRU14] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Datasets*. 2nd. 2014.

[NJW01] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. "On Spectral Clustering: Analysis and an Algorithm". In: *Proceeding of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. 2001, pp. 849–856.

[For10] Santo Fortunato. "Community detection in graphs". In: *Physics Reports* 486.3 (2010), pp. 75–174. DOI: 10.1016/j.physrep.2009.11.002.

[YL15] Jaewon Yang and Jure Leskovec. "Defining and Evaluating Network Communities Based on Ground-truth". In: (2015).

[Ros87] Peter J. Rosseeuw. "Silhouette: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7.

[New06] Mark E. J. Newman. "Modularity and community structure in networks". In: *Proceeding of the National Academy of Sciences* (2006).

[Blo+08] Vincent D. Blondel et al. "Fast unfolding of communities in large networks". In: *Journal of Statistical Mechanics: Theory and Experiment* (2008), P10008. DOI: 10.1088/1742-5468/2008/10/P10008.

[Alb+07] Enrique Alba et al. "Gene Selection in Cancer Classification using PSO/SVM and GA/SVM Hybrid Algorithms". In: *2007 IEEE Congress on Evolutionary Computation*. 2007, pp. 284–290. DOI: 10.1109/CEC.2007.4424483.

[DHH09] Béatrice Duval, Jin-Kao Hao, and Jose Crispin Hernandez Hernandez. "A Memetic Algorithm for Gene Selection and Molecular Classification of Cancer". In: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. 2009, pp. 201–208. DOI: 10.1145/1569901.1569930.

[Pen+03] Sihua Peng et al. "Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines". In: *FEBS Letters* 555.2 (2003), pp. 358–362. DOI: 10.1016/S0014-5793(03)01275-4.

[New01] Mark E. J. Newman. "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality". In: *Physical Review E* 64 1 Pt 2 (2001), p. 016132.

[Ash+00] Michael Ashburner et al. "Gene Ontology: tool for the unification of biology". In: *Nature Genetics* 25 (1 2000), pp. 25–29. DOI: 10.1038/75556.

[Con19] The Gene Ontology Consortium. "The Gene Ontology Resource: 20 years and still GOing strong". In: *Nucleic Acids Research* 47.D1 (2019), pp. D330–D338. DOI: 10.1093/nat/gky1055.

[Oli07] Travis E. Oliphant. "Python for Scientific Computing". In: *Computing in Science & Engineering* 9.3 (2007), pp. 10–20. DOI: 10.1109/MCSE.2007.58.

[MA11] K. Jarrod Millman and Michael Aivazis. "Python for Scientists and Engineers". In: *Computing in Science & Engineering* 13.2 (2011), pp. 9–12. DOI: 10.1109/MCSE.2011.26.

[J+01] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. [Online; accessed Jun, 12, 2019]. 2001. URL: http://www.scipy.org/.

[Oli06] Travis E. Oliphant. *A guide to NumPy*. USA: Trelgol Publishing, 2006.

[WCV11] Stéfan van der Walt, S. Chris Colbert, and Gaël Varoquaux. "The NumPy Array: A Structure for Efficient Numerical Computation". In: *Computing in Science & Engineering* 13 (2011), pp. 22–30. DOI: 10.1109/MCSE.2011.37.

[Hun07] John D. Hunter. "Matplotlib: A 2D Graphics Environment". In: *Computing in Science & Engineering* 9 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[McK10] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. 2010, pp. 51–56.

[Ped+11] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[HSS08] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring Network Structure, Dynamics, and Function using NetworkX". In: *Proceeding of the 7th Python in Science Conference*. 2008, pp. 11–15.

Appendix A

Packages

Python 3 was used for developing.

The following will explain the imported packages to the reader.

SciPy [Oli07] [MA11] is a collection of open-source packages. The SciPy ecosystem is intended for scientific calculations. The following have helped us with this thesis:

- SciPy [J+01],
- NumPy [Oli06] [WCV11],
- Matplotlib [Hun07],
- pandas [McK10], and
- scikit-learn [Ped+11].

Apart from SciPy, we also imported NetworkX [HSS08]. This package is designed for analysis of networks. The program comes packaged with some data structures and algorithms.

TRITA-EECS-EX-2019:305