A Project Report

on

Computational Approach for Disease-Gene Associations

Submitted in partial fulfilment of requirements for award of degree

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

by

G. Satvik Kalyan (160116733106)
B. Sharath Chandra (160116733107)

Under the guidance of

Smt.K. Mary Sudha Rani
Assistant Professor
Department of CSE
CBIT(A), Hyderabad

Department of Computer Science and Engineering
Chaitanya Bharathi Institute of Technology (Autonomous)
(Affiliated to Osmania University, Hyderabad)
Hyderabad, TELANGANA (INDIA) – 500 075
May-2020

CHAITANYA BHARATHI
INSTITUTE OF TECHNOLOGY (A)
Kokapet(Village), Gandipet, Hyderabad, Telangana-500075. www.cbit.ac.in

Affiliated to | NBA | Re-accredited by NAAC | 100th Rank in nirf | ISO 9001:2015

COMMITTED TO RESEARCH, INNOVATION AND EDUCATION 41 years

CERTIFICATE

This is to certify that the project titled “**Computational Approach for Disease-Gene Associations**” is the bonafide work carried out by **G. Satvik Kalyan (160116733106) and B. Sharath Chandra (160116733107)**, students of B.E.(CSE) of Chaitanya Bharathi Institute of Technology, Hyderabad, affiliated to Osmania University, Hyderabad, Telangana(India) during the academic year 2019-20, submitted in partial fulfillment of the requirements for the award of the degree in **Bachelor of Engineering** (**Computer Science and Engineering**) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Supervisor	**Head**
Smt.K. Mary Sudha Rani	**CSE Dept**
Assistant Professor	**CBIT(A)**
CBIT(A), Hyderabad	**Hyderabad**

Place: Hyderabad

Date:

External Examiner

DECLARATION

We hereby declare that the research work entitled **Computational Approach for Disease-Gene Associations** is original and bonafide work carried out by us as a part of fulfilment for Bachelor of Engineering in Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Gandipet, Hyderabad, under the guidance of **Smt K. Mary Sudha Rani**, Assistant Professor Department of CSE, CBIT.

G. Satvik Kalyan (160116733106) B. Sharath Chandra (160116733107)

Place: Hyderabad

Date:

ACKNOWLEDGMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success. They have been a guiding light and source of inspiration towards the completion of the project.

We would like to express our sincere gratitude to our project guide, Mrs. K Mary Sudha Rani who has supported us throughout our project with patience and knowledge.

We are also thankful to Head of the department Dr. M Swamy Das for providing excellent infrastructure and a conducive atmosphere for completing this project successfully.

We are also extremely thankful to our Project Coordinators Dr. Y Rama Devi, Professor, Dept. of CSE and Smt. K Mary Sudha Rani, Assistant Professor, Dept. of CSE for their valuable suggestions and interest throughout the course of this project.

We convey thanks to the lab staff for allowing us to use the required equipment whenever needed.

Finally, we would like to take this opportunity to thank our families for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

G. Satvik Kalyan (160116733106)
B. Sharath Chandra (160116733107)

ABSTRACT

Understanding the link between genetic diseases and also the genes associated with them could be a crucial problem for human health. The vast amount of data created from a large number of high-throughput experiments performed within the last few years has resulted in a huge growth in computational methods to handle the disease gene association problem. Nowadays, it's clear that a lot of genetic diseases does not seem to be the consequence of defects present in a single gene. Proteins present together in a community are represented using a PPI network graph. These PPI networks indicate how proteins interact. In this work, a computational approach for the disease-gene association is devised using Genetic Algorithm and Protein-Protein Interaction Networks (PPI).

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1 Introduction

1.1 Problem Definition

Large amounts of biological data and causes for various diseases and information on genes involved in disease are generated nowadays, using high-throughput experimental techniques. The analysis of such biological data and in particular the identification of the genes and inter-molecular events leading to the formation of diseases remain essential towards the development of effective medical therapies. Since it is a very time consuming and complex process bioinformatics tries to tackle the disease-gene association problem via developing computational disease-gene prediction methods. These methods use the data obtained from the findings of research in related areas and extract useful information from the literature to associate genes with diseases.

We apply the genetic algorithm on a set of a candidate gene, which tries to evolve a community consisting of genes which are highly associated and collaborative with the disease gene set and other genes.

1.2 Methodologies

In this, a computational disease-gene association method using a genetic algorithm is used. Using a genetic algorithm, our computational method tries to evolve a community containing the set of potential genes which are likely to be involved in a given genetic disease. Having a set of known disease genes already believed to be involved in a disease, we first obtain a protein-protein interaction network containing all the selected known disease genes. All the other genes inside the PPI network are then considered as candidate disease genes as they are present in the vicinity of the known disease genes in the human interactome.

Our method attempts to find a community of potential disease genes strongly working with one another and with the genes in the known disease genes set. In our

Genetic Algorithm based approach, each individual present in the population is a candidate community of genes that may be involved in a given disease. The initial population consists of individuals that are simply randomly-created communities. At each generation, the GA tries to evolve individuals (communities) that have a relatively high degree of collaboration between the known disease genes and the other genes within the individual, and that also that have a high degree of collaboration among all the genes inside the individual.

The more collaborative the genes are in the evolved individual with the known disease genes and with one another, the more likely it is that the individual contains genes potentially involved in the disease under study. Intuitively, in our method, the Genetic Algorithm tries to find subnetworks which is as modular and collaborative as possible, where the subnetworks also contain the known disease genes.

As the evolving populations contain a very large number of communities, over time different communities that work with the known disease genes have the opportunity to evolve and be in the same population; genes in all such communities will have high scores as they are frequently selected in a number of the communities over generations, which allows them to increase their scores. Also, potential disease genes that occur in more than one community working with disease genes are selected more frequently as they have the opportunity to be in many of the communities of the populations over time. The GA works to increase the degree of collaboration (i.e. the modularity) of the nodes (i.e. the genes) inside the evolving communities.

1.3 Outline of the results

The results obtained at the end of the algorithm consists of a community with all the candidate genes which are highly probable for causing breast cancer. Since the genetic algorithm is a heuristic-based optimization algorithm the results obtained at the end may vary for every different run of the algorithm. So, in order to draw out the possible candidate genes, the algorithm is performed 20 times and the genes which occur the greatest number of times in the best community are considered as the final genes.

1.4 Scope of the project

The normal determination of the genes that are involved in a disease experimentally is very expensive and also time-consuming. And other computational methods such as text mining doesn’t provide accurate results. But using the genetic algorithm we can mimic the evolution theory and converge into better results than other computational methods. It provides better results as well as saves an incredible amount of time spent in the experimental determination of Genes.

1.5 Organization of project report

This project report is organized as follows:

1. The first chapter mentioned in the table of contents deals with the Introduction of the project and gives an overview of the project.
2. The second chapter discusses the literature survey of this project which gives an insight into the core concepts of our project.
3. The third chapter deals with the methodology of the project i.e. design of our proposed system and theoretical foundation.
4. The fourth chapter displays our implementation of the proposed solution along with algorithm, dataset description and pseudo code along with the testing process.
5. The fifth chapter displays our results and discussions through a series of screenshots.
6. The sixth chapter deals with the conclusion and future scope of the project.

2 Literature Survey

2.1 Introduction to the Problem Domain and Terminology

2.1.1 Bioinformatics: Introduction, History and Goals

Bioinformatics has become an important segment of the many areas of biology. In experimental bioscience, bioinformatics techniques like image and signal processing allow the extraction of useful results from large amounts of source data. within the field of genetics, it aids in sequencing and annotating genomes and their observed mutations. It plays a task within the text mining of biological literature and therefore the development of biological and gene ontologies to arrange and query biological data. It also involves a task within the analysis of gene and protein expression and regulation. Bioinformatics tools aid in comparing, analyzing and interpreting genetic and genomic data and more generally within the understanding of evolutionary aspects of bioscience. At a more integrative level, it helps analyze and catalog the biological pathways and networks that are a prominent a part of systems biology. In structural biology, it aids within the simulation and modeling of DNA, RNA, proteins yet as biomolecular interactions.

The term bioinformatics didn't mean the identical as today. Ben Hesper and Paulien Hogeweg devised within the year 1970 to sit down with the study of data processes in biotic systems. This definition placed bioinformatics as a field parallel to biochemistry (the study of chemical processes in biological systems).[6]

The goal of the Bioinformatics field is to know how simple cellular activities are altered in numerous disease states, the biological data must be combined to come up with a comprehensive picture of these activities. Therefore, the sector of bioinformatics has evolved in an exceedingly way that the foremost important task now involves the analysis and interpretation of various variants of information. This includes nucleotide and aminoalkanoic acid sequences, protein domains, and protein structures. the particular process of analyzing and interpreting data is brought up as computational biology.

Important sub-disciplines within bioinformatics and computational biology include:

1. Development and implementation of computer programs that enable efficient access to, management and use of, various types of information.
2. Development of recent algorithms (mathematical formulas) and statistical measures that assess relationships among members of huge data sets. For instance, there are methods to locate a gene within a sequence, to predict protein structure and/or function, and to cluster protein sequences into families of related sequences.

The fundamental goal of bioinformatics is to improve the understanding of biological processes. What makes it unique from other approaches, however, is its focus on devising and applying computationally intensive techniques to attain this goal. Examples include pattern recognition, data processing, data mining, machine learning algorithms, and visualization. Major research efforts in the field include sequence alignment, gene finding, genome assembly, drug design, drug discovery, protein structure alignment, protein structure prediction, prediction of gene structure, expression, and protein-protein interactions, genome-wide association studies, the modeling of evolution and cell division/mitosis.

Bioinformatics now encompasses the creation and advancement of databases, algorithms, computational and statistical methods, and theory to resolve formal and practical problems that are being generated from the management and analysis of biological data.

Over the past few decades, rapid developments in genomic and other molecular research technologies and developments in information technologies have merged to create immense amounts of knowledge or data related to biological science. Bioinformatics is the name given to these mathematical and computing methods used to obtain an understanding of biological processes. Trivial tasks in bioinformatics include analyzing DNA, mapping and protein sequences, aligning DNA and protein sequences to contrast them and generating and viewing 3-D models of protein structures.

2.1.2 Basic Biological Concepts and Terminology

Gene

Every cell in the human body contains the same DNA sequence, which is composed of a long string of chemicals called nucleotides. There are four types of nucleotides: guanine (G), adenine (A), thymine (T) and cytosine (C). In fact, all the genetic information of the living organisms is encoded in DNA. In humans, DNA contains two long strands each composed of almost 3 billion nucleotides.

Not all parts of DNA are the same. Some parts of the DNA handle more functionalities than the others. In other words, some specific parts of the DNA have been less subjected to mutation and more conserved in the course of evolution. These segments of the DNA which carry inheritable information about one or more functionalities are called genes. A gene is usually defined as the molecular unit of heredity of a living organism. Every cell of every individual contains two complete sets of genes, with each set lying on one of the two strands of the DNA. If one of the sets does not function correctly, the other one may take over the functionalities. The below Figure 2.1 shows both DNA and RNA.

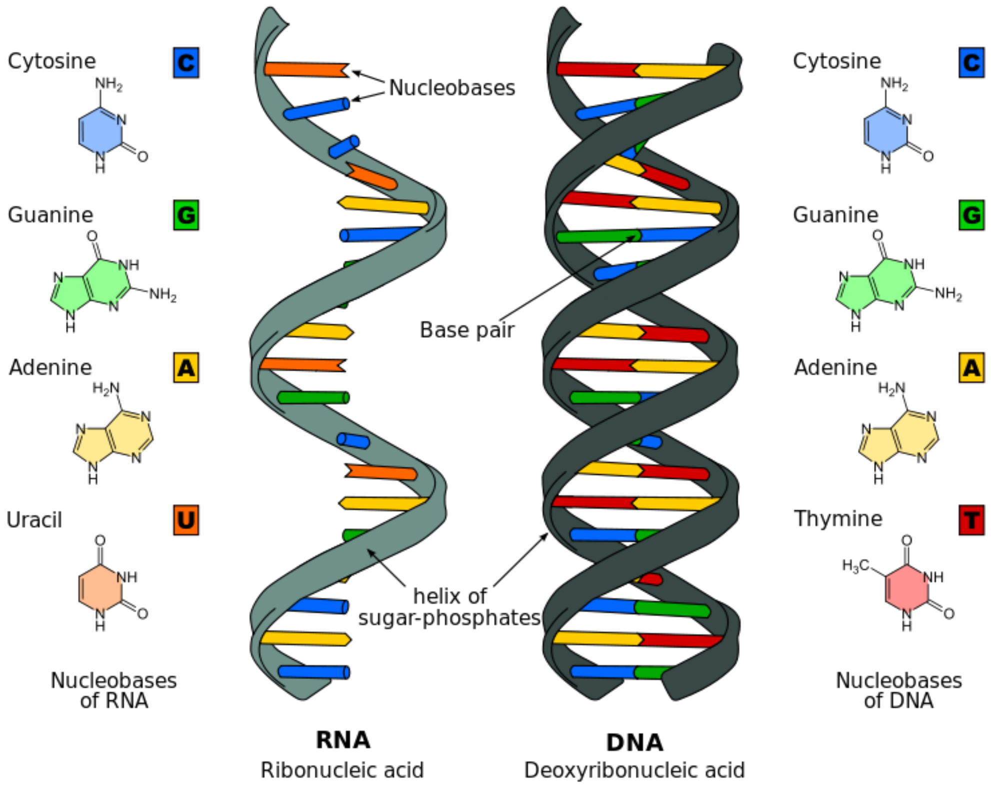


Figure 2.1 RNA and DNA

Proteins

The encoded genetic instructions in the genes serve as blueprints to produce proteins, the large biological molecules performing different functionalities within living organisms. In fact, genes can be interpreted as algorithms which develop proteins, the foremost important functional units of each living organism. The procedure of genes applied to develop proteins is called gene expression. When the instructions in a particular given gene are applied to create proteins, it's said that the gene is expressed. Even though all cells carry all of the genome (i.e. all the genes, or the entire DNA), only a tiny fraction of the genome is expressed in every cell with regard to the specific tasks and functionalities each cell performs (nerve cell, muscle cells, bone cell, etc). As a matter of fact, In each cell type, a specific set of genes is activated, and the proteins they encode give the cell the characteristics of different types of cells such as bone cell,muscle cell, nerve cell and so on. Furthermore, depending on the environment cells alter their patterns of gene expression in order to respond to nutrients, temperature, hormones, infectious agents, and so on. The below Figure 2.2 shows the process of the formation of protein.

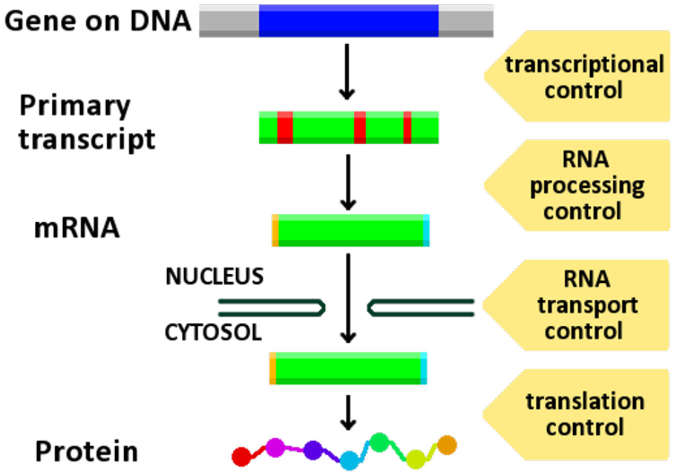


Figure 2.2 Protein formation

Protein-protein interaction networks (PPI)

Protein-protein interaction networks (PPI) represent physical interactions between proteins. This is the most commonly used and powerful type of evidence for disease gene prediction. With respect to the principle of guilt by association, proteins that physically interact are more likely to be involved in the same functionality or phenotype. Therefore, the harmful alteration of any of the proteins interacting in the same functional module can adversely change the functionality or phenotype and consequently lead to the same disorder. The below Figure 2.3 is the graphical visualization of the PPI network.

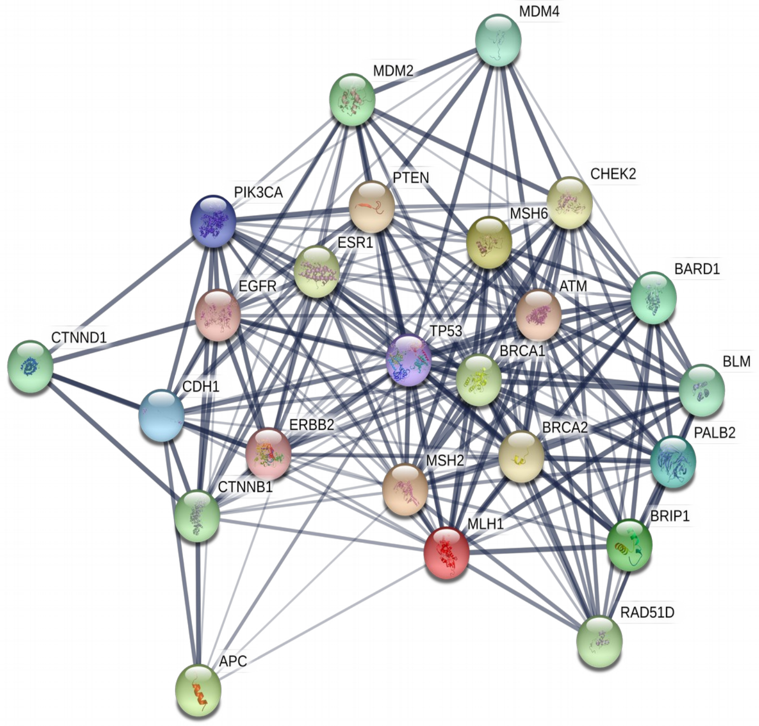


Figure 2.3 PPI network

Chromosomes

In the nucleus of each cell, the DNA molecule is packaged into thread-like structures called chromosomes. Each chromosome is made from DNA tightly coiled multiple times around proteins called histones that support its structure.

Chromosomes aren't visible within the cell’s nucleus not even under a microscope when the cell isn't dividing. However, the DNA that forms up chromosomes becomes more tightly packed during the process (i.e cell division) and is then visible under a microscope.

Each chromosome has a constriction point called the centromere, which divides the chromosome into two sections, or “arms.” The short arm of the chromosome is labeled the “p arm.” The long arm of the chromosome is labeled the “q arm.” The location of the centromere on each chromosome gives the chromosome its characteristic shape, and can be used to help describe the location of specific genes. The below Figure 2.4 shows the Chromosomes.

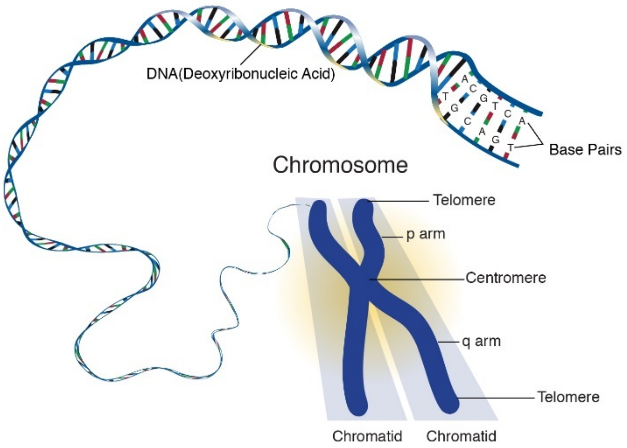


Figure 2.4 Chromosomes

Cross-over

Homologous recombination is the process by which two chromosomes, paired up during prophase 1 of meiosis, exchange some distal portion of their DNA. Crossover occurs when two chromosomes, normally two homologous instances of the same chromosome, break and then reconnect but to the different end piece. If they break at the same place or locus in the sequence of base pairs, the result is an exchange of genes, called genetic recombination. This outcome is the normal way for crossover to occur.

If they break at slightly different loci, the result can be a duplication of genes on one chromosome and a deletion of these on the other. This is known as an unequal crossover. If chromosomes break on both sides of the same centromere and rejoin to exclude the centromere, the result can be one chromosome being lost during cell division.

The below Figure 2.5 shows the Crossover between a Paternal Chromosome and Maternal Chromosome and both Meiosis 1 and Meiosis 2.

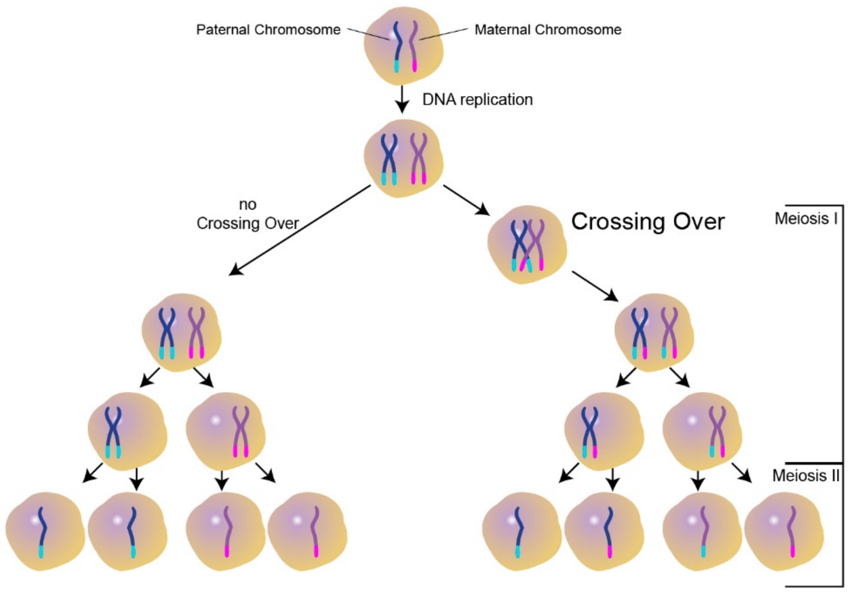


Figure 2.5 Crossover

Mutation

Mutation is a permanent alteration in the DNA sequence that creates a gene, such that the sequence differs from what is found in most peopleis called genetic mutation. Mutations may vary in size; they can affect anywhere from a single DNA building block (base pair) to a large segment of a chromosome that includes multiple genes.

Over a lifetime DNA can undergo changes or mutations in the sequence of bases A, C, G and T. This results in changes in the proteins that are made. This can be both bad or a good thing. Mutations can occur during DNA replication if errors are made and not corrected in time. before it becomes a fixed mutation these cells can recognize any potentially mutation-causing damage and repair it. Mutations contribute to genetic variation within species. particularly if these mutations have a positive effect they also be inherited.

For example, the disorder sickle cell anemia is caused by a mutation in the gene that instructs the building of a protein called hemoglobin. This causes the red blood cells to become an abnormal, rigid, sickle shape. However, in African populations, having this mutation also protects against malaria. However, a mutation can also disrupt normal gene activity and cause diseases, like cancer. Cancer is the most seen human genetic disease; it is caused by mutations occurring in a number of growth-controlling genes. Sometimes faulty, cancer-causing genes can exist from birth, increasing a person's chance of getting cancer. The below Figure 2.6 shows the mutation in which base T is replaced with base C.

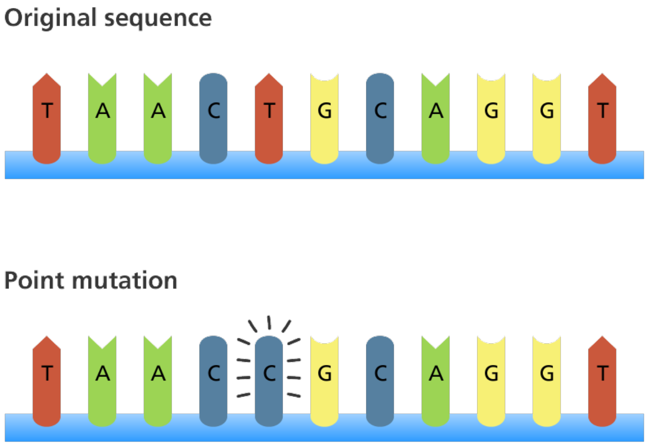


Figure 2.6 Mutation in which base T is replaced with base C

Genetic disease as a type of phenotype

As a conclusion of what has been discussed in previous sections, it is fairly true to suggest that all the functions of living organisms like humans depend on the proteins.

Hundreds of thousands of proteins in our body carry out thousands of chemical reactions required for life.

Humans are a huge set of multitudes of various phenotypes. Phenotypes are sets of observable properties and characteristics of the living organisms at different levels. For instance, a phenotype at the cellular level can be the type of cell, the shape of the cell, or other observable properties of the cell. A phenotype at a higher level can be the human skin color, eyebrow shape, eye color and so on. A phenotype can also be as complex as being alive. The fact that we are alive at a given moment is a complicated phenotype depending on very many processes, interactions and other simpler phenotypes. Thus, at any given moment, we are a set of a myriad of different phenotypes at different levels. All these phenotypes come about subsequent to interactions of an extraordinarily huge number of proteins interacting with each other in a real time network all over the cells of our bodies. This gigantic network across all the cells is referred as the human interactome.

The human interactome is scientifically believed to be one of the most complex networks containing approximately 25000 protein-encoding genes, and also a high number of unknown and undefined proteins and their interactions. The number of nodes of the network including genes and their products such as proteins is estimated to exceed one hundred thousand. The number of functionally relevant interactions between the components of this network, representing the links of the interactome, is expected to be much larger and remains largely unknown. Unfortunately, phenotypes are not always handy or pretty. They can sometimes be undesirable and even destructive. A phenotype can also be a genetic disease or disorder. The most common reason for genetic disorders is changes or mutations happening in the genes. The altered gene is called a mutated gene. Mutation in the DNA is not a rare incident. As most parts of human DNA are not functionally important, mutations in these parts do not usually lead to harmful consequences. However, if mutation occurs in a functional part of the DNA − inside the genes which handle some functionalities − it may adversely affect some crucial processes required for life, consequently causing genetic diseases.

The most common mutation causing genetic disease is SNP (single nucleotide polymorphism) in which one single nucleotide A, T, C or G is mutated and replaced by another one. If this occurs in a coding area of DNA (genes) it may affect the protein function and consequently lead to genetic disorders.

Disease-gene Association Problem

Associating genes with a specific group of phenotypes, i.e. the genetic diseases is one amongst the key of gene-phenotype association research. Associating genes with genetic diseases and disorders is crucial to recognize the genetic basis of human diseases. The specific research area problem where the genes which are in any way involved in the existence of a given genetic disease are identified is called the disease-gene association problem, or identification of disease genes, or disease gene prediction.

Understanding the complex correlation between genes and proteins requires the processing of a vast amount of data from a wide variety of genomic data sources. Thus, computational tools have become critical for the integration, representation, and visualization of heterogeneous biomedical data. To be extremely general, computational disease-gene association methods apply all the possible and potential findings throughout years of research in related areas, using whatever useful information can be found in the literature to associate genes with diseases. The below Figure 2.7 shows how literature is used to predict diseases using Disease Gene Association.

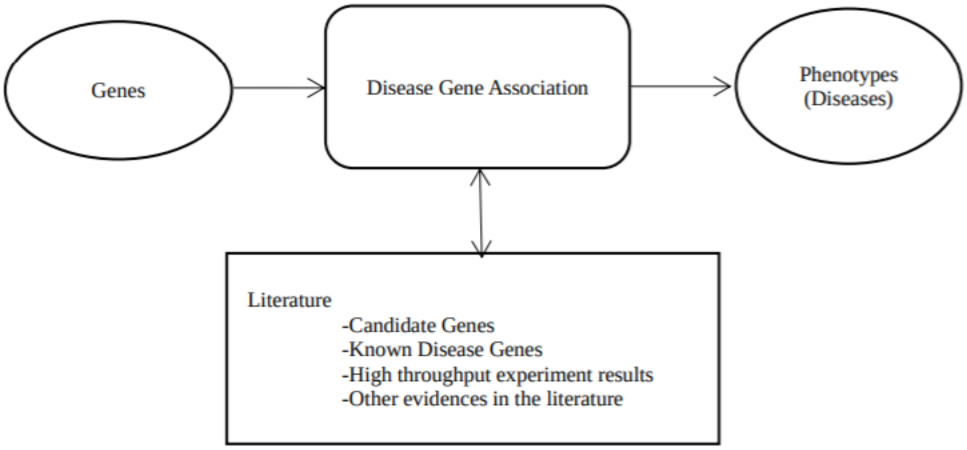


Figure 2.7 Image showing how literature is used to predict diseases using Disease Gene Association

Disease-gene prediction methods choose potential disease genes from a set of candidate genes usually determined by different experimental and other computational methods including Genomewide Association Studies (GWAS) and Linkage Analysis. In Genomewide Association Studies hundreds of thousands of SNPs (Single Nucleotide Polymorphisms) are statistically investigated for association with genetic disease in hundreds of thousands of individuals. Linkage Analysis typically associates certain chromosomal loci (linkage intervals) with a particular disease phenotype and it consists of expensive practical experiments performed in the laboratory to determine loci on chromosomes which have the tendency to be inherited alongside the known disease genes or mutated genes which are believed to be involved in diseases.

Principle of Guilt by Association

Mutations in multiple proteins that form a protein complex may lead to the same disease phenotype. Various genes involved in the same phenotypes work together in a single biological module. This is also called the modular nature of genetic disease.

2.2 Existing solutions

Existing solutions include Text-mining of biomedical literature discovers Gene-disease association by using natural language processing techniques over huge quantities of related written knowledge which are a multitude of biomedical abstracts and studies. Since text mining misses a lot of prominent features in disease gene associations a better computational approach is necessary.

2.3 Related Works

2.3.1 INTEGRO: an algorithm for data-integration and disease-gene association

INTEGRO[1] is an algorithm used for disease gene association and data integration based on the information retrieved from OMIM, NCBI-MedGen, SNOMED-CT, Disease Ontology and DisGeNET. A latest selected version of DisGeNET was used to ensure the reliability and validity of the information.

INTEGRO's pipeline was summarized by the following steps:

1. To establish the relation between the input and the information is annotated in DO to identify a defined term of interest in this the input is referred to as the word Term.
2. To extract attributes related to the Term for example cross-references and DO-id.
3. Then visit the DO's graph to identify the Term with the right attributes contained inside its relationships.
4. Then analyze the cross-references to retrieve external information by OMIM, NCBI-MedGen, SNOMED-CT, Disease Ontology and DisGeNET. And at last, the disease-gene associations and other information such as definition, UMLS codes and treatments are also integrated while excluding redundancies.

DO is a Directed Acyclic Graph which presents terms linked in the hierarchy and interrelated subtypes. In this attribute, "is_a" was used to identify the root node for a given term. In this DO's graph, the terms become more specific while the with depth increased whereas the root nodes are therefore more generic. Here in this graph terms can have multiple values but the term with more similarity and greater depth is chosen discarding others.

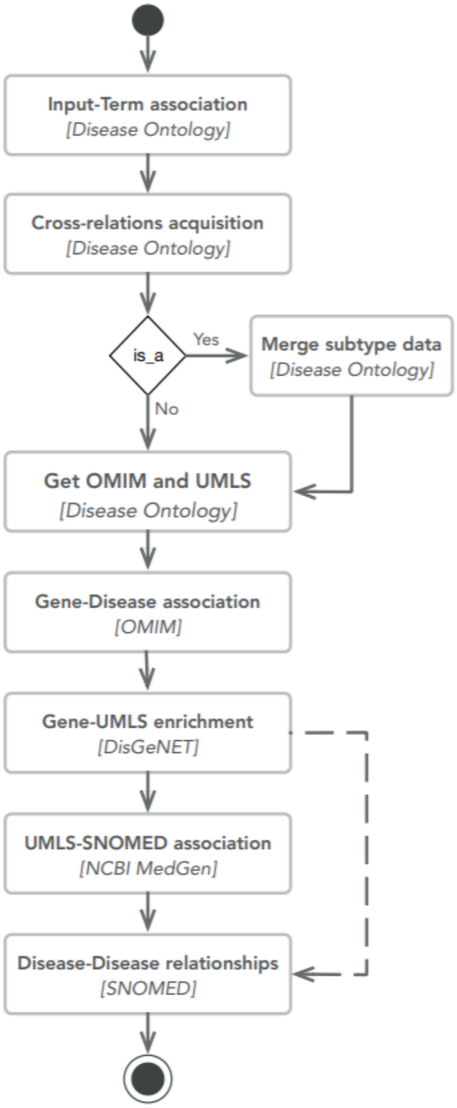


Figure 2.8 Non-exhaustive workflow for the INTEGRO's pipeline

The above Figure 2.8 shows the Non-exhaustive workflow for the INTEGRO's pipeline.

First, the DOID for a given Term is obtained by using best match case and then in next the phase of data analysis and subsequent integration few OMIM identifiers which are also referred as cross-references in DO are used to perform Disease gene association using MorbidMap.

2.3.2 Disease-Gene Association Using a Genetic Algorithm

Authors proposed a disease-gene association using a genetic algorithm[2]. The genetic algorithm can give an effective optimal solution for many NP problems. The genetic algorithm helps us to provide a population of some candidate solutions for a given problem. These candidate solutions are referred to as individuals or chromosomes. And a fitness function is used to evaluate the fitness of the solution obtained from the Genetic algorithm. This fitness is used to measure how good an obtained solution is for a given problem and a process of crossing and mutations are applied to obtain the new off-springs.

The genetic algorithm will follow the below steps. The initial population is generated. This initial population is also considered as the candidate solution. Then we will be measuring the fitness of the solutions. If the fitness obtained is of desired value then the algorithm stops. Even if the number of iterations reaches the number of generations then also algorithm stops. In the next step, we will select the individuals as the parents and apply cross-over followed by mutation to obtain the next generation and flow continues.

After obtaining the best community having highest fitness and then after obtaining the best genes they compared it with the CIPHER results which is one of the best disease-gene association framework.

2.3.3 Gene-disease association through topological and biological feature integration

They developed a learning model [3] which helps them in classifying genes as diseased which are related to diseases based on both biological and topological features. When this model is given a list of genes, it classifies between a disease gene and not a disease gene classes. The topology of the corresponding PPI network is combined with the various sources to discover similarities characterize each class. They achieved an area under the receiver operating characteristic (ROC) curve of 0.941 using Naive Bayes classifier.

Topological and Biological Feature Integration

Topological Features: Mutations caused when two or more proteins interact with each other lead to a similar type of phenotypes. Protein-Protein Interaction network is a network in which in interactions are represented by network and then proteins are represented by nodes and these affect the functional associations among genes. Studying these PPI networks helps a lot in associating the disease with disease-causing genes and below are the topological features used.

Degree: The term degree describes the number of edges which are adjacent to a node.

Eccentricity: Eccentricity means the distance from a node to the furthest node from it in network.

Closeness Centrality: The average distance from a node to all other nodes in the network.

Betweenness Centrality: Number of times a node appears on shortest paths between nodes in the network.

Authority: The value of information stored at a node.

Hub: The quality of a node's links.

Modularity Class: The class reflecting how well a network decomposes into modular communities.

PageRank: The rank of a node by its importance in the network.

Component ID: The number of connected components to a node in the network.

Clustering Coefficient: The completeness of the neighborhood of a node in the network.

Number of triangles: The number of connected triangles including a node.

Eigenvector Centrality: The importance of a node in the network based on its connections.

Biological Features

Many experimental observations can help in differentiating between the disease and non-disease gene. So, they added more attributes. They considered the following biological features.

Sequence Length: Disease genes have longer sequences. So, the number of amino acids in the genes are considered.

Gene Ontology (GO) Terms: GO project provides a set of vocabulary which describe the gene products by their cellular components, molecular functions and biological processes. Cellular components are the parts of a cell or its extracellular environment in which gene exists. Molecular functions explain the elemental activities of a gene product at the molecular level. Biological processes cover the events which are related to functioning of integrated living units which include tissues, cells, organisms and organs.

Pathway: They considered the pathways in which genes participate may potentially direct the association of genes to a disease.

Domain: It is defined as a combination of secondary structures organized into a characteristic 3D structure or folds; protein domains usually correspond to structural domains which fold independently of the rest of the protein chain.

Topological Domains: The topology and compartments of the proteins present in the cell can potentially take part in disease or non-disease gene classification.

Chain: This feature is to considered the polypeptide chain in the protein.

Protein Family: They assumed that the proteins belonging to same family will share similar functional properties and this may help in associating genes with disease.

They extracted the association data which has 1777 genes linked to 1284 disorders. They preprocessed the dataset by removing the genes which are related to multiple diseases, unclassified diseases, neurological diseases and some other disease genes. After preprocessing the PPI data, they obtained a dataset of 9228 genes. They captured the accurate and consistent information of proteins which include cross-references, biological ontologies and classifications and some other characteristics. And they considered the top features and values describing them.

They then developed a classification model, BT-Classifier, based on Naive Bayesian classifier and used Weka data mining software. They generated results which are based on the 10-fold cross-validation parameters. ROC curve which is an indicator of the classification quality and the classification model corresponds to an AUC of 0.941.

2.3.4 A Novel Disease Gene Prediction Method Based on PPI Network

They describe how to identify the disease-causing gene with a function flow-based model. Function flow-based model[4] consists of two main steps. In the first step, they constructed a weighted protein interaction network from protein interactions between genes, and a map the known disease gene and the candidate disease genes to the protein interaction network. In the second step, candidate genes are ranked based on the function flow model based on the functional similarity between the candidate genes and known disease genes, and the candidate genes with the highest are suspected to be disease genes.

Function Flow Model Algorithm

This algorithm works on the principle of guilt by association. If the candidate genes are connected by physical interactions with the disease genes then they are more likely to causes the disease.

They achieve this by treating each disease gene as a 'source' of 'functional flow' for a particular disease. Then they simulated the spread of the functional flow through the protein-protein interaction network and then each protein obtains the 'functional score' which corresponds to the amount of 'flow' that the protein has received from the source protein.

The function flow model algorithm is done in three steps:

1. Identify the disease gene in the PPI network
2. Then simulate the function flow to get the flow from the source (known disease genes) for each candidate disease gene as a function score. The score obtained by a candidate gene may be zero if the flow did not reach that protein.
3. Rank the genes according to the functional score, the genes with a high score are considered more likely to be disease genes.

The above algorithm is continued for every known disease gene and the score for each particular candidate genes is updated in iterations.

The process of iteration subject to some rules:

1. A node delivers the function flow preserved in its reservoir to its direct neighbors which are proportional to the weight of the edge between two nodes, the functional flow does not cross the capacity it can pass on.
2. When the disease gene node has enough functional flow in its reservoirs to deliver to the candidate gene. They simplified it by capping the maximum flow that can be passed to the candidate gene to 1 in each iteration irrespective of the weight of the edge between them.

As each candidate disease gene receives the function flow from a source node in all iterations. The source node with enough flow and the function flow pass on at the discrete-time, so the nodes closer to the source node (disease gene) receive more flow than the nodes which are far from the source node (disease gene).

Variables

u - a protein in PPI network

$R_t(u)$ - the amount in the reservoir that a protein u has at time t.

$G_t(u,v)$ - the function flow from protein u to protein v at time t.

d - Number of iterations

At time 0, only the disease gene node has a reservoir of function flow, the formula is as follows:

$$R_0(u) = \begin{cases} 1, & \text{if } u \text{ is a known disease gene} \\ 0, & \text{otherwise} \end{cases}$$

In every iteration, the reservoir of each protein will be updated according to the total inflow and total outflow of the node.

$$R_t(u) = R_{t-1}(u) + \sum_{v \in N(u)} (g_t(v,u) - g_t(u,v))$$

As the flow will occur from the nodes with more flow to nodes with less flow at time 0. And the in each iteration the flow is towards downhill it is given by the following formula.

$$g_t(u,v) = \begin{cases} 0, & \text{if } R_{t-1}(u) < R_{t-1}(v) \\ min(w_{u,v}, R_{t-1}(u) \frac{w_{u,v}}{\sum_{(u,y)\in E} w_{u,y}}), & \text{otherwise} \end{cases}$$

Finally, the score can be calculated for node u according to the total amount of flow it preserves over d iterations.

$$f(u) = \sum_{t=1}^{d} \sum_{v \in N(u)} g(v,u)$$

They obtained 8959 proteins and 33528 distinct interactions using this data they constructed a weighted protein interaction network. Their method successfully ranked 102 known disease genes in top1 out of all 723 known disease genes.

2.4 Tools/Technologies

2.4.1 GeneMANIA

GeneMANIA is a multiple association network integration algorithms for predicting the gene function. Using GeneMania we obtained an un-weighted protein-protein interaction network containing the disease genes. We defined the candidate genes to be the first 2000 genes scored by GeneMania in the order of interactions with these disease genes.

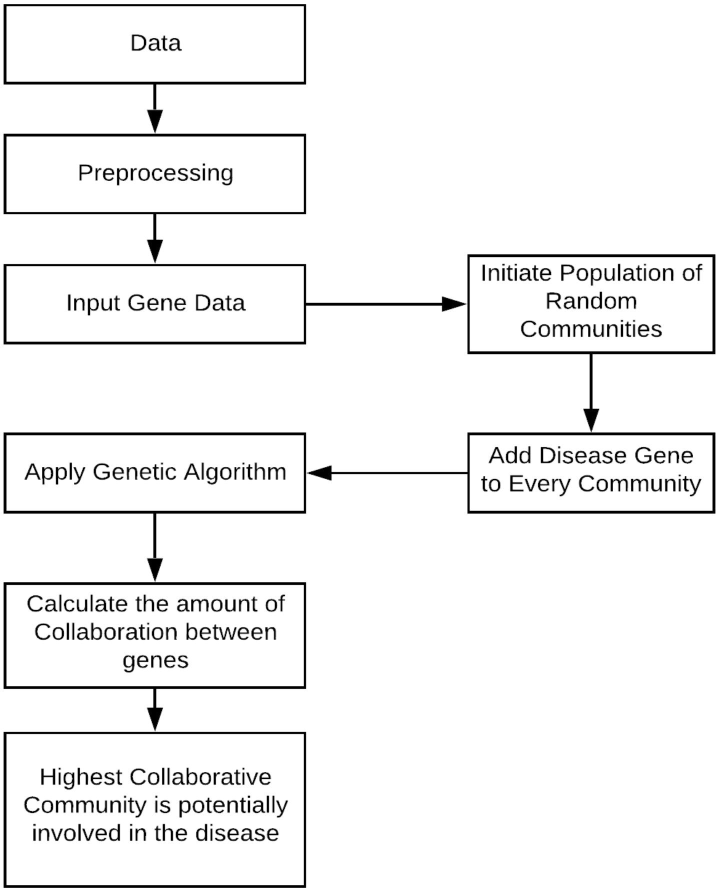
2.4.2 STRING

STRING is a database predicted protein-protein interactions. The interactions include direct and indirect associations or in other terms interactions between physical and functional associations. They stem from computational prediction, from interactions aggregated from other primary databases and from knowledge transfer between organisms.

It also helps to generate a PPI network for a given set of genes. It also offers many visualization facilities.

3 DESIGN OF THE PROPOSED SYSTEM/ALGORITHM

3.1 Block Diagram



Data
Preprocessing
Input Gene Data
Initiate Population of Random Communities
Add Disease Gene to Every Community
Apply Genetic Algorithm
Calculate the amount of Collaboration between genes
Highest Collaborative Community is potentially involved in the disease

Figure 3.1 Block Diagram for computational approach for disease-gene associations

The above Figure 3.1 shows the complete flow of the processes taking place at a high level. The following steps occur:

1. Initially, the data is obtained from GeneCards and GeneMania. The data is generally of two types, one is a PPI network and other is Disease gene association data.

2. In the data pre-processing step, we clean the data before it is being fed to the algorithm. Remove unnecessary genes from the datasets.

3. The data obtained is now given to the Genetic algorithm and a sparse matrix is generated using the PPI network dataset which represents a graph or a network.

4. The communities are randomly generated and added to the initial population. This population of communities is used for the starting population of Generation 0 of genetic algorithm.

5. The disease genes are then added into the communities present in the entire population.

6. Genetic Algorithm is started, it is running for a fixed number of generations.

7. Calculate the collaboration between the genes and check if its the best community of genes obtained so far.

8. At the end the last generation we obtain the candidate genes which are highly associated with breast cancer.

3.2 Activity diagram

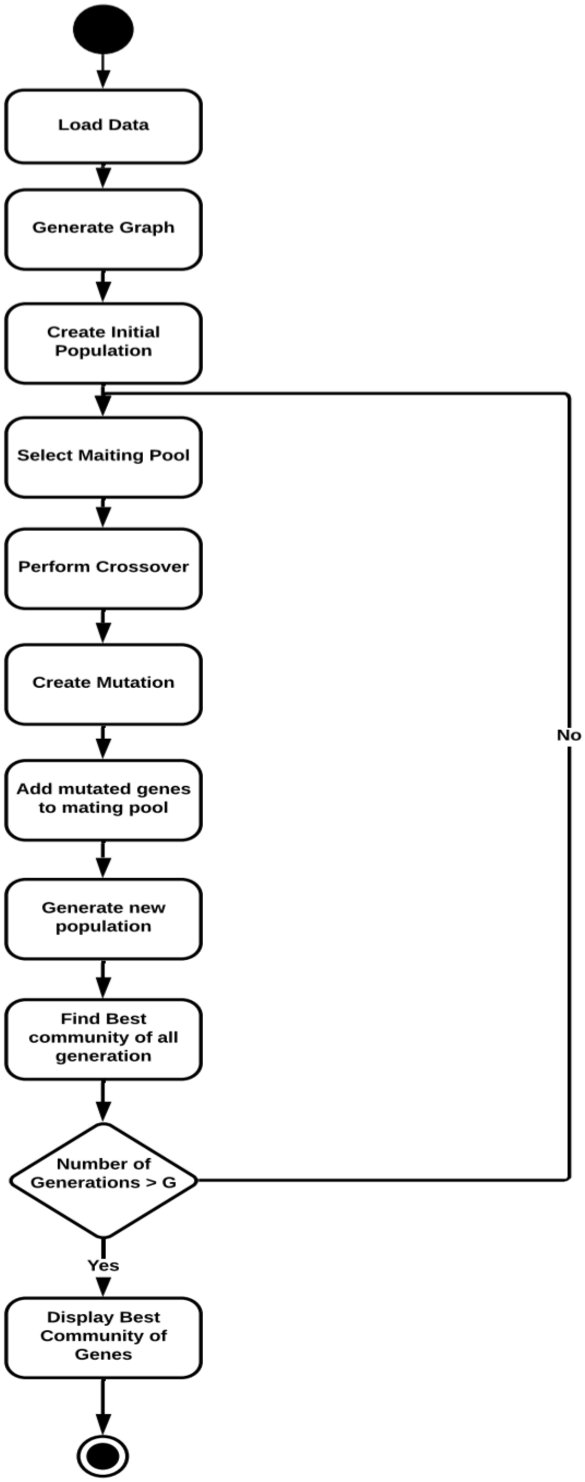

Load Data
Generate Graph
Create Initial Population
Select Maiting Pool
Perform Crossover
Create Mutation
Add mutated genes to mating pool
Generate new population
Find Best community of all generation
Number of Generations > G
No
Yes
Display Best Community of Genes

Figure 3.2 Activity Diagram for computational approach for disease-gene associations

The above Figure 3.2 is the Activity Diagram for this computational approach. Initially, we load both the PPI network data and also disease-gene association data. From the PPI network data, we generate a sparse matrix which represents an undirected graph. Depending on the population size we generate a population of randomly generated communities to serve as an initial population. Then a few genes are selected into the mating pool which are the genes which move forward to the next generation. From the genes present in the mating pool we select parents for performing cross over. After performing cross over the newly generated Genes are added into the mating pool. Then mutation is performed at the given rate and new population is generated. This population is forwarded to the next generation and serves as the current population of the next generation. This process continues till either required fitness is achieved or the number of generations reaches the predefined limit. At each generation, the fitness of every community is calculated and the community with the highest fitness value is considered as the best community.

3.3 Algorithm

Genetic Algorithm

Genetic Algorithms (GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. They are commonly used to generate high-quality solutions for optimization problems and search problems.[7][8]

Genetic algorithms simulate the process of natural selection. Every generation of Genetic Algorithm(GA) consists of a population of individuals and each individual represents a degree in search space and possible solution. Every community/Individual is represented as a string of character/integer/float/bits. This string is analogous to the Chromosome. A Fitness Score is given to every individual which shows the flexibility of an individual to compete. The individual having optimal fitness score (or near-optimal) is sought.

A genetic algorithm (GA) contains the following important elements:

1. A population of candidate solutions for the problem at hand. These candidate solutions are also called chromosomes or individuals.
2. A fitness function that takes a candidate solution and evaluates its fitness value. The perfectness of a solution to a given problem depends on the fitness value.
3. A crossover operator that is a genetic operator applied on two or more individuals (chromosomes) for offspring reproduction.
4. A mutation operator that mimics possible mutations while new individuals are reproduced.

A general recipe for a genetic algorithm can be described as having the following steps:

1. Generate an initial population of candidate solutions. This initial population can be generated in many ways but often it is simply a set of random valid candidate solutions.
2. Fitness evaluation: measure the fitness of each individual in the current generation's population.
3. If there is a solution in the population with the desired fitness value, or if the number of generations created so far has reached the predefined maximum number of generations, then the program stops.
4. Tournament selection: randomly select a number (tournament size) of the individuals and select a number (usually two) of them to be the parents reproducing individuals for the next generation.
5. Generate the population for the next generation: all selected parents in step 4 utilize the crossover operator and possibly the mutation operator to create

offspring which will become the individuals of the next generation (new population). This repeats until the new population becomes full.

6. Go to 2.

Applying a genetic algorithm (GA), we propose a new computational approach to tackle the disease gene association problem.

3.4 Theoretical Foundation

Initially we convert all the n genes present in the PPI network into n by n matrix where each position in the matrix denotes if there is a link present between the two genes. If a link is present between them, then we indicate it by setting the position in the matrix and 0 is there is no link between the genes. We also take an n-length Binary array which represents a community of genes a set bit indicates that gene of the corresponding index is present in the community.

Initial population

The initial population is Randomly generated expect for the 16 genes which are always included in very community at every generation.

Parent selection

Out of the population of communities, we select a few parents which are going to be passed on to the next generation and which participate to generate population for the next generation. we select the parents based on the fitness of the community. The fitness of the community is Calculated using the page rank algorithm we select n genes and apply page rank algorithm over them and sum up the modularity of those n genes. We then select the parents using k tournament selection method.

Modularity

Modularity of a node in a network is given as the ratio of number of nodes connecting that node to another node to the total number of connections in the network.

The modularity function Q for a community C is given as

$$Q\ (Ci) = \sum_{j=1}^{j=n} E(i,j)$$

Where

C_i is the ith Gene in the community C

$\sum E(i,j)$ is the number of edges from node i to the remaining nodes j

Crossover population

We then apply a single point cross over on the selected parents and generate new genes.

Mutation (mutation rate)

We mutate the genes based on the mutation rate by flipping the random genes present in the community.

4 IMPLEMENTATION OF THE PROPOSED SYSTEM

4.1 Pseudo Code/Algorithm

Genetic Algorithm

Algorithm: Genetic Algorithm

```
numberOfGenerations ← ɵ;
bestFitness ← 0;
while generation $\leq$ numberOfGenerations do
    parentGenes ← selectMatingPool(population);
    crossOverPopulation ← crossover(parentGenes);
    mutatedPopulation ← mutation(crossOverPopulation);
    population ← mutatedPopulation;
    fitness ← calculateFitness(population);
    if fitness $\geq$ bestFitness then
        fitness ← bestFitness;
end
```

The Genetic algorithm takes the number of generations to be performed, the population size and also the mutation rate. Initially, we generate a population of randomly generated communities to serve as the initial population. Then a few genes are selected into the mating pool which are the genes which move forward to the next generation. From the genes present in the mating pool we select parents for performing cross over. After performing cross over the newly generated Genes are added into the mating pool. Then mutation is performed at the given rate and new population is generated. This population is forwarded to the next generation and serves as the current population of the next generation. This process continues till either required fitness is achieved or the number of generations reaches the predefined limit. At each generation, the fitness of every community is calculated and the community with the highest fitness value is considered as the best community.

4.2 Data Set description

Disease-Gene Association Data

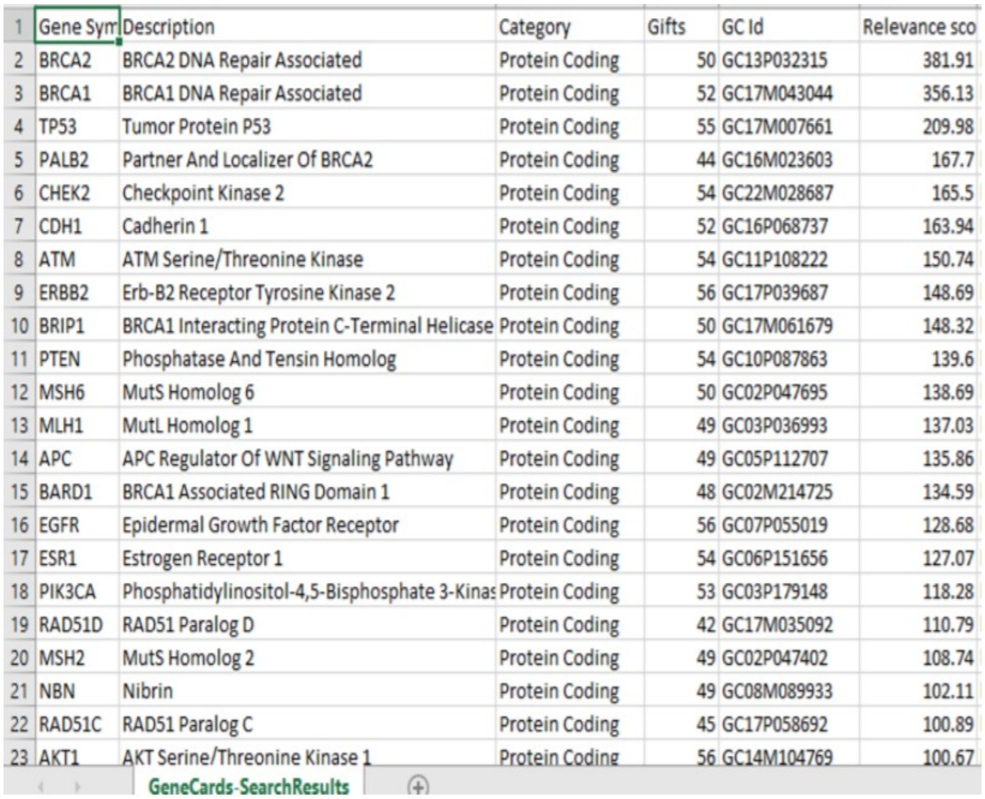
In order to find the disease genes association data, we used GeneCards. GeneCards is a Human Gene Database that provides all predicted human genes which integrates genetic data from 150 web sources including clinical data. The genes are ranked according to disease relevance.

Protein-protein Interaction Data

These above Disease gene data is given as input to GeneMANIA which helps us in obtaining the Protein-Protein Interaction network (PPI). GeneMANIA is an interaction database of homo sapiens(human) which records Co-localization, Genetic Interactions, Pathway, Shared protein domains, Co-expression, Attributes. This database also contains proteomics and genomics data from different sources. GeneMANIA mainly relies on GEO, BioGRID, EMBL-EBI, Pfam, Ensembl, NCBI, MGI, I2D, InParanoid, Pathway Commons.

Data Collection

There are two types of data required to be given as input to our model. The first one being the Disease gene data and the other one is the data which helps in constructing the PPI network. The Disease-gene data is obtained from the GeneCards website which provides us with the genes associated with a specified disease.

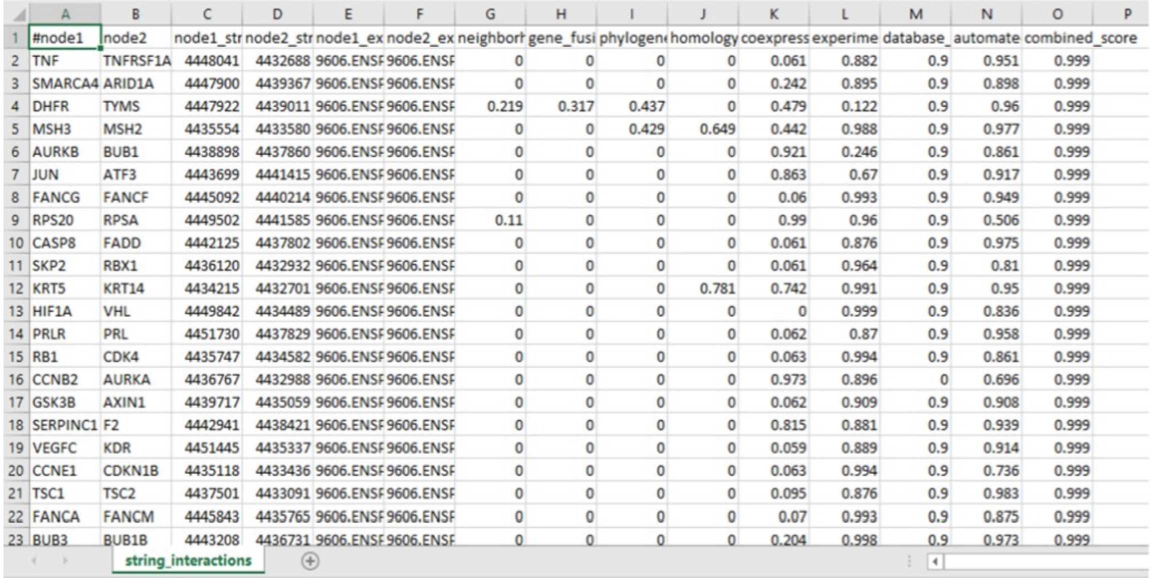


1	Gene Sym	Description	Category	Gifts	GC Id	Relevance sco
2	BRCA2	BRCA2 DNA Repair Associated	Protein Coding	50	GC13P032315	381.91
3	BRCA1	BRCA1 DNA Repair Associated	Protein Coding	52	GC17M043044	356.13
4	TP53	Tumor Protein P53	Protein Coding	55	GC17M007661	209.98
5	PALB2	Partner And Localizer Of BRCA2	Protein Coding	44	GC16M023603	167.7
6	CHEK2	Checkpoint Kinase 2	Protein Coding	54	GC22M028687	165.5
7	CDH1	Cadherin 1	Protein Coding	52	GC16P068737	163.94
8	ATM	ATM Serine/Threonine Kinase	Protein Coding	54	GC11P108222	150.74
9	ERBB2	Erb-B2 Receptor Tyrosine Kinase 2	Protein Coding	56	GC17P039687	148.69
10	BRIP1	BRCA1 Interacting Protein C-Terminal Helicase	Protein Coding	50	GC17M061679	148.32
11	PTEN	Phosphatase And Tensin Homolog	Protein Coding	54	GC10P087863	139.6
12	MSH6	MutS Homolog 6	Protein Coding	50	GC02P047695	138.69
13	MLH1	MutL Homolog 1	Protein Coding	49	GC03P036993	137.03
14	APC	APC Regulator Of WNT Signaling Pathway	Protein Coding	49	GC05P112707	135.86
15	BARD1	BRCA1 Associated RING Domain 1	Protein Coding	48	GC02M214725	134.59
16	EGFR	Epidermal Growth Factor Receptor	Protein Coding	56	GC07P055019	128.68
17	ESR1	Estrogen Receptor 1	Protein Coding	54	GC06P151656	127.07
18	PIK3CA	Phosphatidylinositol-4,5-Bisphosphate 3-Kinas	Protein Coding	53	GC03P179148	118.28
19	RAD51D	RAD51 Paralog D	Protein Coding	42	GC17M035092	110.79
20	MSH2	MutS Homolog 2	Protein Coding	49	GC02P047402	108.74
21	NBN	Nibrin	Protein Coding	49	GC08M089933	102.11
22	RAD51C	RAD51 Paralog C	Protein Coding	45	GC17P058692	100.89
23	AKT1	AKT Serine/Threonine Kinase 1	Protein Coding	56	GC14M104769	100.67

Figure 4.1 The dataset containing disease genes and their relevance score

The above Figure 4.1 almost consists of 13000 records of gene and their associated relevance score. The relevance score is a measure of how prominent a gene is in causing the disease.

The PPI network data is obtained from GeneMania which is a real-time multiple association network integration algorithms for predicting gene function. Using GeneMania we obtained an un-weighted protein-protein interaction network containing the disease genes. We defined the candidate genes to be the first 2000 genes scored by GeneCards in the order of interactions with these disease genes. These candidate genes are given into Genemania to obtain the PPI network.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	#node1	node2	node1_str	node2_str	node1_ex	node2_ex	neighborh	gene_fusi	phylogen	homology	coexpress	experime	database_	automate	combined_score	
2	TNF	TNFRSF1A	4448041	4432688	9606.ENSP	9606.ENSP	0	0	0	0	0.061	0.882	0.9	0.951	0.999	
3	SMARCA4	ARID1A	4447900	4439367	9606.ENSP	9606.ENSP	0	0	0	0	0.242	0.895	0.9	0.898	0.999	
4	DHFR	TYMS	4447922	4439011	9606.ENSP	9606.ENSP	0.219	0.317	0.437	0	0.479	0.122	0.9	0.96	0.999	
5	MSH3	MSH2	4435554	4433580	9606.ENSP	9606.ENSP	0	0	0.429	0.649	0.442	0.988	0.9	0.977	0.999	
6	AURKB	BUB1	4438898	4437860	9606.ENSP	9606.ENSP	0	0	0	0	0.921	0.246	0.9	0.861	0.999	
7	JUN	ATF3	4443699	4441415	9606.ENSP	9606.ENSP	0	0	0	0	0.863	0.67	0.9	0.917	0.999	
8	FANCG	FANCF	4445092	4440214	9606.ENSP	9606.ENSP	0	0	0	0	0.06	0.993	0.9	0.949	0.999	
9	RPS20	RPSA	4449502	4441585	9606.ENSP	9606.ENSP	0.11	0	0	0	0.99	0.96	0.9	0.506	0.999	
10	CASP8	FADD	4442125	4437802	9606.ENSP	9606.ENSP	0	0	0	0	0.061	0.876	0.9	0.975	0.999	
11	SKP2	RBX1	4436120	4432932	9606.ENSP	9606.ENSP	0	0	0	0	0.061	0.964	0.9	0.81	0.999	
12	KRT5	KRT14	4434215	4432701	9606.ENSP	9606.ENSP	0	0	0	0.781	0.742	0.991	0.9	0.95	0.999	
13	HIF1A	VHL	4449842	4434489	9606.ENSP	9606.ENSP	0	0	0	0	0	0.999	0.9	0.836	0.999	
14	PRLR	PRL	4451730	4437829	9606.ENSP	9606.ENSP	0	0	0	0	0.062	0.87	0.9	0.958	0.999	
15	RB1	CDK4	4435747	4434582	9606.ENSP	9606.ENSP	0	0	0	0	0.063	0.994	0.9	0.861	0.999	
16	CCNB2	AURKA	4436767	4432988	9606.ENSP	9606.ENSP	0	0	0	0	0.973	0.896	0	0.696	0.999	
17	GSK3B	AXIN1	4439717	4435059	9606.ENSP	9606.ENSP	0	0	0	0	0.062	0.909	0.9	0.908	0.999	
18	SERPINC1	F2	4442941	4438421	9606.ENSP	9606.ENSP	0	0	0	0	0.815	0.881	0.9	0.939	0.999	
19	VEGFC	KDR	4451445	4435337	9606.ENSP	9606.ENSP	0	0	0	0	0.059	0.889	0.9	0.914	0.999	
20	CCNE1	CDKN1B	4435118	4433436	9606.ENSP	9606.ENSP	0	0	0	0	0.063	0.994	0.9	0.736	0.999	
21	TSC1	TSC2	4437501	4433091	9606.ENSP	9606.ENSP	0	0	0	0	0.095	0.876	0.9	0.983	0.999	
22	FANCA	FANCM	4445843	4435765	9606.ENSP	9606.ENSP	0	0	0	0	0.07	0.993	0.9	0.875	0.999	
23	BUB3	BUB1B	4443208	4436731	9606.ENSP	9606.ENSP	0	0	0	0	0.204	0.998	0.9	0.973	0.999	

Figure 4.2 Dataset representing a PPI network containing breast cancer candidate genes

The above Figure 4.2 dataset consists of about 76670 records. The visualization of the above network is done using STRING database which is a biological database and web resource of known and predicted protein-protein interactions. The STRING database contains information from various sources, which include experimental data, computational methods, prediction methods and text collections. It also provides functionality to visualize the PPI network. The visualization of the above PPI network looks something like this. The below Figure 4.3 shows the visualization of the PPI network.

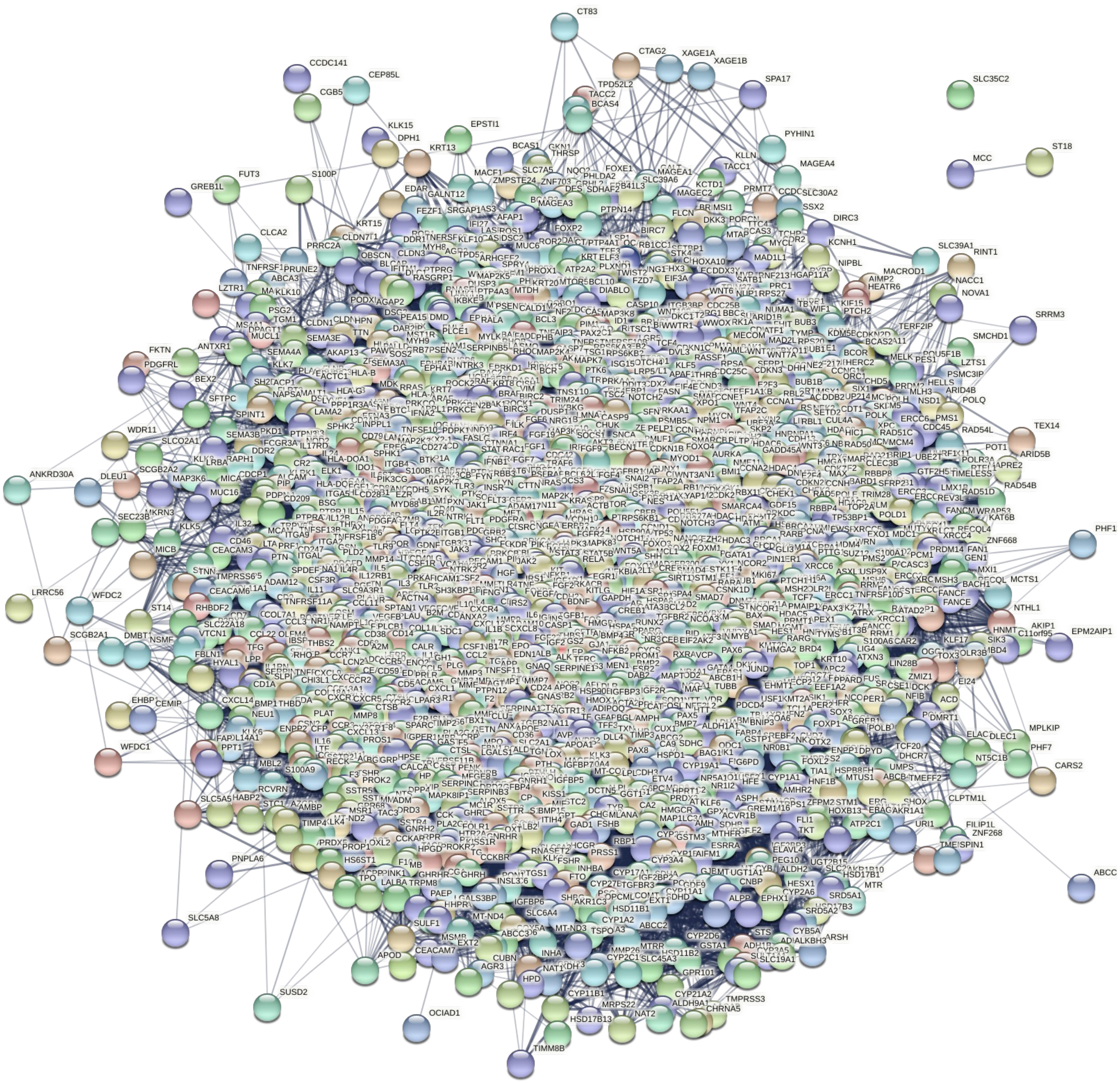
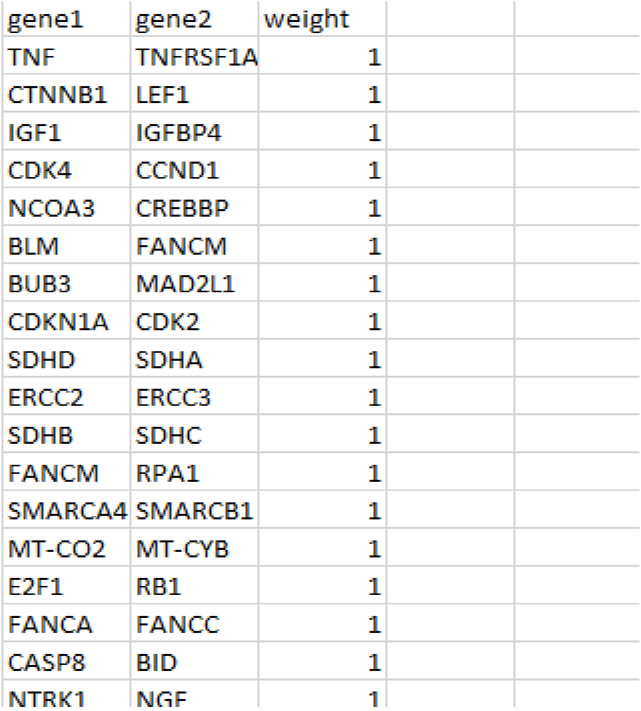


Figure 4.3 Visualization of PPI network containing breast cancer candidate genes

Data Preprocessing

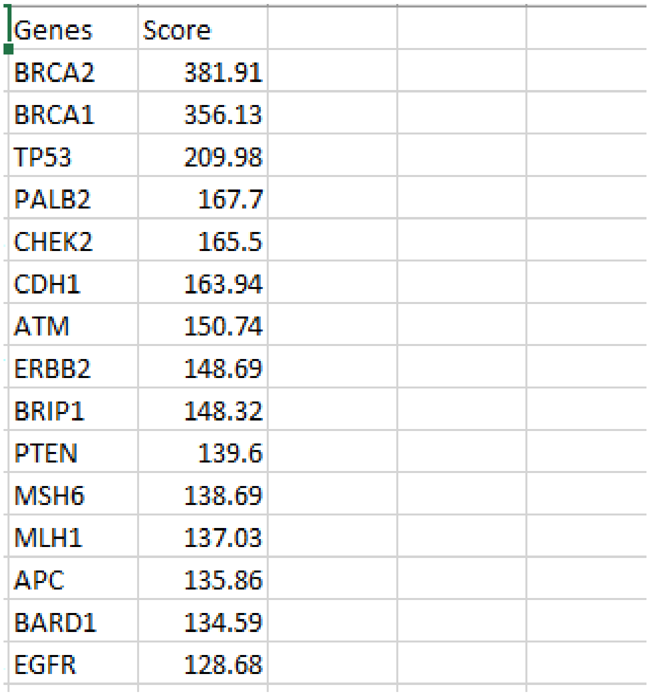
The PPI network not only consists of those 2000 genes but also some other genes with which these 2000 genes may or may not interact. So, we need to cross-verify both the dataset and retain the genes which are present in both the dataset. In the dataset on PPI network, we only take the gene pairs and their scores which denotes the link or interactivity between them. Whereas in disease gene dataset we consider the gene and their corresponding Relevance Score. After performing data preprocessing and removing the features that are not necessary dataset is obtained.



gene1	gene2	weight
TNF	TNFRSF1A	1
CTNNB1	LEF1	1
IGF1	IGFBP4	1
CDK4	CCND1	1
NCOA3	CREBBP	1
BLM	FANCM	1
BUB3	MAD2L1	1
CDKN1A	CDK2	1
SDHD	SDHA	1
ERCC2	ERCC3	1
SDHB	SDHC	1
FANCM	RPA1	1
SMARCA4	SMARCB1	1
MT-CO2	MT-CYB	1
E2F1	RB1	1
FANCA	FANCC	1
CASP8	BID	1
NTRK1	NGF	1

Figure 4.4 PPI network after Data preprocessing

The above Figure 4.4 shows the final dataset of PPI network obatined after Data prepocessing.



Genes	Score
BRCA2	381.91
BRCA1	356.13
TP53	209.98
PALB2	167.7
CHEK2	165.5
CDH1	163.94
ATM	150.74
ERBB2	148.69
BRIP1	148.32
PTEN	139.6
MSH6	138.69
MLH1	137.03
APC	135.86
BARD1	134.59
EGFR	128.68

Figure 4.5 Disease gene association data after Data preprocessing

The above Figure 4.5 shows the Disease gene association data after Data preprocessing.

4.3 Testing Process

We have tested our computational approach on 16 known breast cancer genes taken from similar experiments performed by Xuebing et al [5] to test the performance of algorithm CIPHER which is a well-known disease gene prioritization framework. The below Table 4.1 shows the 16 known breast cancer genes used by Xuebing et al.

Gene	**NCBI ID**
BRCA1	672
BRCA2	675
TP53	7157
AR	367
ATM	472
CHEK2	11200
STK11	6794
RAD51	5888
PTEN	5728
BARDI	580
RB1CC1	9821
NCOA3	8202
PIK3CA	5290
PPMID	8493
CASP8	841
TGFI	281527

Table 4.1 Known Disease Genes for Breast Cancer

We have carried out three types of testing processes

A. Normal Cross Validation

In the first process, we have initially generated a random initial population which may or may not consist of the disease gene. After running the algorithm over several generations few candidate genes are generated and these candidate genes are cross verified against the 16 Genes used to test the performance of CIPHER.

B. Leave-One-Out Cross-Validation

To assess the performance of our GA-base method, we used the leave-one-out cross-validation procedure. This procedure is the most frequently used procedure to assess many of the state-of-the-art disease-gene associations approaches. According to the leave-one-out cross-validation procedure, for every time performing the experiment, one of the known disease genes is removed from the known disease genes set, to observe whether or not the method can again recover this gene as a high associative gene. Given a set of candidate genes K and a set of known disease genes D a gene $g \in D$ is left out and all genes in $K \cup \{g\}$ are given as the input to the algorithm to check if the left-out gene d is regenerated. This process is repeated for every gene in the known disease genes set.

C. Validation of genes provided in Dataset

The dataset provided for this system is first validated as given below.

The genes that are provided for the system are compared with the genes that are extracted from various biomedical websites which have integrated, searchable, database of human genes.

5.RESULTS AND DISCUSSIONS

Parameter Tuning

The technique of adjusting the parameter of our models such that the behavior of the model changes so that we obtain better results is called parameter tuning. The various parameters present in our model are Population size, Number of Generations, mutation rate. In order to determine the best population size and number of generations for our model we have run our algorithm with different number of generations and population sizes.

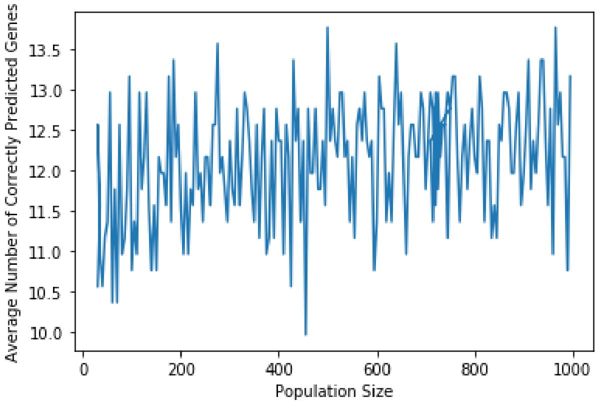


Figure 5.1 Graph showing the trend of correctly predicted genes as population size increases

The above Figure 5.1 is plotted against Average number of correctly predicted genes and population size. By keeping the number of generations constant initially and increasing the size of the population from 30 to 1000 by increasing it by an interval of 5 a graph is plotted against the average number of correctly predicted genes and population size. We run the model ten times for every size of the population and take an average of the predicted values. We can observe a peek at a population size of 965 with an average of 13.7.

The below Figure 5.2 is plotted against Average number of correctly predicted genes and number of generations. On keeping the population size as 965 and changing the number of generations by 50 through 0-1000 we obtained a peek at 400th generation with an average of 13.9.

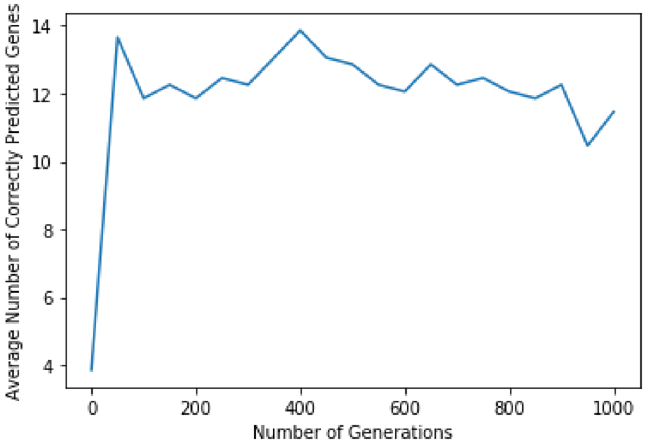


Figure 5.2 Graph showing the trend of correctly predicted genes as Number of Generations increases

Genetic Algorithm Parameters

Parameter	Value
Number of Generations	400
Population Size	950
Mutation rate	5
Tournament Size	4
Fitness function	Q (C)

Table 5.1 Genetic Algorithm Parameters

Using the cross-validation method, we have run the algorithm with the parameters in the above Table 5.1 for 20 different times. We have compared the candidate genes obtained with the known disease gene set and it is observed that on an average 14 out of the 16 genes are successfully predicted. And by using the leave one out cross-validation we are able to successfully regenerate 12 out of 16 disease genes.

For every Generation, we are given all the gene communities present within the population and also their corresponding score. At the end of all the generations, we are given a community which has the highest score calculated by the fitness function. This community with the highest score contains the disease genes and also the candidate genes. These Candidate genes are most likely to be involved in breast cancer. The output of the algorithm at the end of the 400th Generation looks like as follow.

```
=======Community With best fitness at the end of 400th Generation=======


['BRCA1', 'TP53', 'PALB2', 'CHEK2', 'PTEN', 'MSH6', 'MLH1',
'PIK3CA', 'NBN', 'CCND1', 'NF1', 'CDKN2A', 'AR', 'VEGFA',
'CDK4', 'MDM2', 'POLE', 'CDKN1A', 'CYP19A1', 'CDKN1B',
'TERT', 'HRAS', 'TGFBR2', 'RAD51', 'STAT3', 'BAX', 'TNF',
'RAD50', 'MTOR', 'CASP8', 'IL6', 'PLAU', 'PMS2', 'IGF2',
'FGFR1', 'NOTCH1', 'FAS', 'KIT', 'SRC', 'ALK', 'FHIT',
'RAF1', 'AURKA', 'AKT2', 'PPM1D', 'TOP2A', 'BMPR1A', 'PRLR',
'NME1', 'TGFBR1', 'EZH2', 'SPP1', 'IDH1', 'MMP1', 'KLLN',
'PTGS2', 'KRT8', 'CASP3', 'TWIST1', 'GNRH1', 'FLT1', 'WWOX',
'KRT5', 'CXCL12', 'MMP9', 'GLI1', 'ABCG2', 'INS', 'IGF1',
'GATA3', 'AKT3', 'MMP2', 'FASLG', 'CYP1B1', 'MAPK1',
'SLC22A18', 'BUB1B', 'ABCB1', 'NOS2', 'CTSD', 'BIRC5',
'NRAS', 'AHR', 'RAD54L', 'FGF8', 'ERBB3', 'TFF1', 'GJA1',
'PHB', 'RARA', 'ING1', 'WNT1', 'NTRK1', 'ETS1', 'FGFR3',
'ABCC1', 'TACC1', 'PPARG', 'MGMT', 'TGFA', 'STS', 'RET',
'HFE', 'RRAS2', 'BCAR3', 'CDK1', 'CDK2', 'CYP1A1', 'TFAP2A',
'CXCL8', 'BCAR1', 'CASP9', 'BCL2L1', 'STAT5B', 'SEMA3A',
'CBFB', 'MEN1', 'NRG1', 'TSC2', 'WT1', 'MAPK3', 'KRT19',
'IL1B', 'CAV1', 'RASSF1', 'CEACAM5', 'GSTM1', 'BRMS1',
'PTHLH', 'ATR', 'CTNNA1', 'NFKB1', 'NAT2', 'CCNB1',
'TNFRSF10B', 'IFNG', 'CCNE1', 'FN1', 'SP1', 'BMP6', 'CCNA2',
'GSK3B', 'TOP1', 'MMP14', 'SDHB', 'SLC2A1', 'PRL',
'RPS6KB1', 'TIMP1', 'SMAD2', 'NQO2', 'CCND2', 'KRT18',
'SHBG', 'FLT4', 'PMS1', 'NF2', 'KLK10', 'MAPK14', 'HDAC1',
'GADD45A', 'AXL', 'GPER1', 'BCAS1', 'SERPINE1', 'PIK3CG',
'TSC1', 'HDAC4', 'ITGB1', 'SKP2', 'AREG', 'TNFSF11',
'HOXB13', 'ICAM1', 'TSG101', 'ERCC4', 'XBP1', 'RAD51B',
'CDH2', 'PAK1', 'PIP', 'MTHFR', 'COMT', 'ANKRD30A',
'DNMT3B', 'CDH3', 'MSH3', 'MMP11', 'CASP7', 'MUCL1', 'ECT2',
'AGR3', 'CAT', 'CA9', 'S100A4', 'FGF7', 'SCGB2A2',
'ALDH1A1', 'CSF1R', 'CTLA4', 'IDH2', 'TG', 'FGF1', 'CD40LG',
'HBEGF', 'SRA1', 'BECN1', 'CDK7', 'CASP10', 'TEK', 'CREB1',
'POMC', 'CCND3', 'CDC25A', 'MAP3K1', 'ANXA1', 'ITGA6',
'MMP7', 'KRT20', 'KLF6', 'DDR1', 'KDM5B', 'NCOA1', 'APOD',
'VDR', 'MUC16', 'GPX1', 'BLM', 'MAGED2', 'TFAP2C', 'NCOA6',
'GRN', 'HEATR6', 'HMGA2', 'CSNK1D', 'AGR2', 'PALLD', 'CCL2',
'SFN', 'SULT1E1', 'TIMELESS', 'SMARCB1', 'MITF', 'IGFBP5',
'FBLN1', 'CDC25B', 'OPCML', 'TEX14', 'THBS1', 'RHOBTB2',
'NOTCH4', 'SOX10', 'CDC27', 'TGFB3', 'SLC9A3R1', 'SUSD2',
'ETV6', 'MAPK12', 'MST1R', 'PTPN11', 'SOX9', 'CHGA',
'PDGFA', 'ADIPOQ', 'BIRC3', 'ERCC3', 'TP53BP1', 'CEP85L',
'KITLG', 'MED12', 'LOXL2', 'FH', 'SRRM3', 'MC1R', 'ANXA5',
'MCC', 'GRP', 'NPM1', 'POT1', 'CSK', 'INSR', 'CCNB2',
'VEGFB', 'RAC1', 'MUC4', 'PRDM14', 'TIMP4', 'DCTN5',
'MAP2K5', 'AKAP13', 'FOXP3', 'PAX5', 'TNFRSF10A', 'DPH1',
'CTSB', 'XPC', 'NR5A1', 'SYP', 'AKIP1', 'FLCN', 'TYR',
'IRF1', 'IL4', 'IL7', 'KIF15', 'NR0B1', 'HLA-A', 'BDNF',
'TUBB', 'MAP3K6', 'FANCM', 'IFNA1', 'PSCA', 'IKBKG',
'ERCC5', 'TBX3', 'FBXO11', 'PRSS1', 'CDC6', 'DNMT3A',
'CYP2E1', 'PRC1', 'SOD1', 'ARID1A', 'MSLN', 'DLC1', 'PXN',
'VWF', 'RUNX3', 'F3', 'RECQL', 'ZFHX3', 'GAST', 'SETBP1',
'SDHA', 'BCR', 'HLA-B', 'SNAI1', 'FANCD2', 'CD34', 'PROM1',
'S100B', 'CEACAM3', 'SRD5A1', 'AMHR2', 'SSTR2', 'MYD88',
'KRT10', 'RNASEL', 'CDKN2C', 'IL5', 'NOTCH2', 'KMT2D',
'ILK', 'PTPRF', 'ADH1C', 'CD28', 'NR1H2', 'FLNA', 'NCAM1',
'IL24', 'CLDN4', 'CRP', 'BGLAP', 'PROX1', 'HSD11B2',
'FEZF1', 'MAP2K4', 'IFNB1', 'RECQL4', 'ATF1', 'ANXA2',
'HPRT1', 'ANGPT2', 'BAD', 'CCL3', 'PPP2R1B', 'BIRC2',
'G6PD', 'BCL6', 'SOS1', 'KLK6', 'FOXP1', 'PPP2R1A', 'NGFR',
'S100A8', 'SMARCE1', 'CCR7', 'NES', 'FSHB', 'BCAS2', 'TF',
'SIRT1', 'SATB1', 'DSP', 'BMP2', 'KLK4', 'FGF23', 'POU1F1',
'CISH', 'WNT6', 'LDHA', 'PDPN', 'FBN1', 'SLPI', 'CADM1',
'APEX1', 'SLC6A4', 'ANTXR1', 'ARSH', 'ELAC2', 'COL1A1',
'PECAM1', 'POU5F1', 'SPDEF', 'NOTCH3', 'GHR', 'F2',
'MAGEA1', 'PTCH2', 'CD24', 'SMARCA2', 'HMOX1', 'F8',
'TNFRSF11B', 'UHRF1', 'PDCD4', 'NOS3', 'LRP5', 'GZMB',
'POSTN', 'CLDN3', 'ZEB2', 'MYH11', 'SCGB2A1', 'MYLK',
'APOE', 'FOXL2', 'CCL11', 'PTTG1', 'FAP', 'MAP2K2',
'CYP3A4', 'GNA11', 'CTNND1', 'CR2', 'PRKAA2', 'IL7R',
'COL1A2', 'GRHL2', 'CXCR3', 'MS4A1', 'RUNX1', 'GDF15',
'EIF4EBP1', 'SHH', 'PSG2', 'IGF2BP3', 'PSEN2', 'HSPA4',
'EML4', 'INSL3', 'SLC29A1', 'PAX8', 'FOLR1', 'GNAQ',
'CYP11A1', 'XRCC5', 'WNT3', 'IL13', 'MAX', 'NCOR1', 'HLA-
```

DQB1', 'AURKB', 'LOX', 'TET2', 'DHFR', 'TACR3', 'KLK5', 'HABP2', 'LALBA', 'WNT2B', 'TLR9', 'ACPP', 'EI24', 'OCA2', 'TNFRSF11A', 'LZTR1', 'HSPD1', 'EGR1', 'FMR1', 'F13A1', 'CD79A', 'SULT1A1', 'APC2', 'MXI1', 'TNFRSF6B', 'GREB1L', 'PIN1', 'FANCA', 'KAT6B', 'PDPK1', 'ATAD2', 'PAWR', 'PIK3R2', 'DPP4', 'HNF4A', 'SPIN1', 'TXN', 'PIK3R3', 'HLA-DRB1', 'CDH13', 'AVP', 'HTR2A', 'HPSE', 'ETV5', 'SUFU', 'NTRK2', 'ACTA2', 'KCTD1', 'THPO', 'DIRAS3', 'CDKN1C', 'LZTS1', 'IL15', 'NCOA2', 'CHRNA5', 'S100A9', 'WNT4', 'F5', 'TTN', 'WWTR1', 'SLC6A3', 'PLCG1', 'FLNB', 'EXT1', 'EPAS1', 'CXCR5', 'CTSL', 'EPB41L3', 'MDM4', 'XDH', 'ACE', 'GPR101', 'FABP4', 'FKTN', 'HDAC2', 'EPSTI1', 'SLC5A5', 'ERCC8', 'CDKN2D', 'HSD17B3', 'ELF3', 'LCT', 'FEN1', 'AMBP', 'HOXA11', 'LAMC2', 'SPTAN1', 'MRPS22', 'IRS2', 'CTAG2', 'PHF7', 'GHRL', 'SERPINC1', 'GSTM3', 'IL11', 'SSTR5', 'MT-ND4', 'C11orf95', 'IL32', 'RAPH1', 'MAF', 'RRAS', 'TKT', 'SOCS1', 'E2F4', 'NEU1', 'ANPEP', 'ENO1', 'TYRP1', 'PENK', 'FILIP1L', 'TNC', 'GRPR', 'PRUNE2', 'GPNMB', 'PDGFRL', 'MUC6', 'TGM1', 'OLFM4', 'TRPV6', 'SSTR1', 'HIC1', 'HMGCR', 'NUP214', 'PIM1', 'STIM1', 'PRKCE', 'NDRG1', 'ORC1', 'DRD3', 'CCAR2', 'XPO1', 'CXCL1', 'FUS', 'SULF1', 'MTR', 'TK1', 'RIPK1', 'GPR68', 'CD247', 'MECOM', 'LGALS3BP', 'DMRT1', 'PRDM2', 'ADA', 'TNFRSF8', 'NUMA1', 'BIRC7', 'E2F2', 'NCOR2', 'TACC2', 'MELK', 'ARHGEF2', 'SF3B1', 'GREM1', 'RHOB', 'HTR3A', 'IDO1', 'TFRC', 'SUZ12', 'VCAM1', 'SPA17', 'RAD54B', 'GAB1', 'CCDC141', 'WNT3A', 'SEMA3E', 'NANOG', 'MYH9', 'PRKDC', 'TNFAIP3', 'EPHB6', 'TRIM37', 'F2R', 'SIK1', 'BMP10', 'BGN', 'GNB1', 'PTPN1', 'CDH5', 'SH3KBP1', 'NFKB2', 'NKX3-1', 'ASXL1', 'TGM2', 'FLI1', 'VTCN1', 'MMP10', 'ZMIZ1', 'WEE1', 'MAML2', 'THRB', 'MPLKIP', 'SCT', 'PPARD', 'ARAF', 'CD1A', 'INHBA', 'CCDC170', 'RHO', 'OTX2', 'CRH', 'CD9', 'PRKCB', 'RNF213', 'SOCS3', 'NFIB', 'AGER', 'TIAM1', 'FGF10', 'PSENEN', 'SPINT1', 'ZNF268', 'CYP2C19', 'OBSCN', 'PPARA', 'HNRNPA1', 'ATP2C1', 'HSP90B1', 'PRKCI', 'TLR3', 'MFGE8', 'XRCC6', 'IGFBP7', 'BTK', 'MAD2L1', 'RHBDF2', 'CX3CR1', 'IQGAP1', 'ROR1', 'APOBEC3G', 'TMPRSS6', 'CCL4', 'GAD1', 'SLC5A8', 'SHOX', 'SPHK1', 'ACTN4', 'TMEFF2', 'APPL1', 'TRPM8', 'CCR3', 'ACACA', 'GEN1', 'SFTPC', 'PRMT7', 'FZD7', 'IL16', 'ID1', 'JUND', 'NR3C2', 'RAC2', 'ADAM17', 'PLCB1', 'CSNK2A1', 'FLRT3', 'NR4A1', 'DDX3X', 'TNFSF13B', 'OSM', 'DNTT', 'NIPBL', 'ZNF217', 'PPT1', 'CEACAM1', 'ABCC3', 'ABCA3', 'KDM6A', 'ARID5B', 'HNRNPA2B1', 'HNRNPK', 'CASP1', 'PEBP1', 'CLDN1', 'BRINP1', 'IL12RB1', 'CASC3', 'CDCP1', 'IKBKE', 'CCR2', 'ISG15', 'PTH', 'BTC', 'PRKACB', 'TCF3', 'MDK', 'TTC4', 'MBP',

```
'PES1', 'ITGA2', 'GPC6', 'LSP1', 'MICA', 'ELN', 'CEACAM6',
'KL', 'MCM4', 'NUP107', 'ADAM10', 'DLL4', 'PHLDA2',
'ZMPSTE24', 'RPS27A', 'CYP21A2', 'SIX1', 'ANGPT1', 'FAN1',
'SLC39A1', 'PLXND1', 'BMP7', 'JUNB', 'PIK3CD', 'AIFM1',
'SULF2', 'FGF19', 'TPX2', 'CSF3R', 'KRT13', 'GHRH', 'GDNF',
'CUX1', 'CD1D', 'SFRP2', 'RHEB', 'CCKAR', 'BACH1', 'HIPK2',
'NR1I2', 'ZFPM2', 'UMPS', 'BCAS4', 'PRDX5', 'CUL4A', 'HDGF',
'CRTC1', 'TPD52', 'CCR4', 'DUSP1', 'DMBT1', 'PIAS1', 'CNBP',
'LAMA2', 'CT83', 'XAGE1A', 'CSE1L', 'XRCC4', 'GSTA1',
'KCNH1', 'HSD11B1', 'ATRX', 'ENPP2', 'SH2B3', 'TBL1XR1',
'USF1', 'ATF3', 'SMO', 'CXCR2', 'CXCL14', 'MCM5', 'MSI1',
'GALT', 'CUBN', 'ROCK2', 'SSTR4', 'TRAF6', 'KLF17', 'ABCB4',
'BUB3', 'ROR2', 'CD7', 'AKAP9', 'MT-ND2', 'IGF2BP2',
'ALDOA', 'FYN', 'PAX3', 'DHH', 'TMEM97', 'TRAF3', 'PRDM16',
'SLC45A3', 'PHLPP1', 'USF2', 'MFN2', 'RBL1', 'MAP1LC3A',
'IGFBP6', 'DDX5', 'ITGA9', 'LAMTOR5', 'NT5C1B', 'GHRHR',
'INPPL1', 'ARHGAP11A', 'LYN', 'RPS27', 'GAS6', 'AFAP1',
'MICB', 'SLC30A2', 'THRSP', 'WASF3', 'ATP7A', 'CXCR1',
'MDC1', 'TRH', 'LPP', 'TRIM29', 'CD80', 'SLCO2A1', 'RTEL1',
'PRMT1', 'PMAIP1', 'PTPRA']
```

Output for Generation 0-10

```
======================= Generation 0 =======================


 Generation 0 Best Fitness 16723.0

['BRCA2', 'BRCA1', 'TP53', 'PALB2', 'CHEK2', 'CDH1', 'ATM',
'ERBB2', 'BRIP1', 'BARD1', 'ESR1', 'PIK3CA', 'NBN',
'RAD51C', 'KRAS', 'CCND1', 'STK11', 'SMAD4', 'NF1',
'CDKN2A', 'VEGFA', 'BRAF', 'CDKN1A', 'CYP19A1', 'TERT',
'HRAS', 'MET', 'RB1', 'TGFBR2', 'STAT3', 'RAD50', 'MTOR',
'CASP8', 'IGF2', 'PGR', 'FGFR1', 'FAS', 'KIT', 'AXIN2',
'ALK', 'FHIT', 'CXCR4', 'RAF1', 'AURKA', 'MUC1', 'AKT2',
'PIK3R1', 'PPM1D', 'PDGFRB', 'BMPR1A', 'TYMP']

======================= Generation 1  ======================


 Generation 1 Best Fitness 16723.0

['BRCA2', 'BRCA1', 'TP53', 'PALB2', 'CHEK2', 'CDH1', 'ATM',
'ERBB2', 'BRIP1', 'BARD1', 'ESR1', 'PIK3CA', 'NBN',
'RAD51C', 'KRAS', 'CCND1', 'STK11', 'SMAD4', 'NF1',
'CDKN2A', 'VEGFA', 'BRAF', 'CDKN1A', 'CYP19A1', 'TERT',
'HRAS', 'MET', 'RB1', 'TGFBR2', 'STAT3', 'RAD50', 'MTOR',
'CASP8', 'IGF2', 'PGR', 'FGFR1', 'FAS', 'KIT', 'AXIN2',
'ALK', 'FHIT', 'CXCR4', 'RAF1', 'AURKA', 'MUC1', 'AKT2',
'PIK3R1', 'PPM1D', 'PDGFRB', 'BMPR1A', 'TYMP']

======================= Generation 2 =======================


 Generation 2 Best Fitness 16945.0

['BRCA2', 'BRCA1', 'TP53', 'PALB2', 'CHEK2', 'CDH1', 'ATM',
'ERBB2', 'BRIP1', 'BARD1', 'ESR1', 'PIK3CA', 'NBN',
'RAD51C', 'KRAS', 'CCND1', 'STK11', 'SMAD4', 'NF1',
'CDKN2A', 'VEGFA', 'BRAF', 'CDKN1A', 'CYP19A1', 'TERT',
'HRAS', 'MET', 'RB1', 'TGFBR2', 'STAT3', 'RAD50', 'MTOR',
'CASP8', 'IGF2', 'PGR', 'FGFR1', 'FAS', 'KIT', 'AXIN2',
'ALK', 'FHIT', 'CXCR4', 'RAF1', 'AURKA', 'MUC1', 'AKT2',
'PIK3R1', 'PPM1D', 'PDGFRB', 'BMPR1A', 'TYMP']
```

```
======================= Generation 3 =======================
```

Generation 3 Best Fitness 16945.0

```
['BRCA2', 'BRCA1', 'TP53', 'PALB2', 'CHEK2', 'CDH1', 'ATM',
'ERBB2', 'BRIP1', 'BARD1', 'ESR1', 'PIK3CA', 'NBN',
'RAD51C', 'KRAS', 'CCND1', 'STK11', 'SMAD4', 'NF1',
'CDKN2A', 'VEGFA', 'BRAF', 'CDKN1A', 'CYP19A1', 'TERT',
'HRAS', 'MET', 'RB1', 'TGFBR2', 'STAT3', 'RAD50', 'MTOR',
'CASP8', 'IGF2', 'PGR', 'FGFR1', 'FAS', 'KIT', 'AXIN2',
'ALK', 'FHIT', 'CXCR4', 'RAF1', 'AURKA', 'MUC1', 'AKT2',
'PIK3R1', 'PPM1D', 'PDGFRB', 'BMPR1A', 'TYMP']
```

```
=======================Generation 4  =======================
```

Generation 4 Best Fitness 17713.0

```
['BRCA2', 'BRCA1', 'TP53', 'PALB2', 'CHEK2', 'CDH1', 'ATM',
'ERBB2', 'BRIP1', 'BARD1', 'ESR1', 'PIK3CA', 'NBN',
'RAD51C', 'KRAS', 'CCND1', 'STK11', 'SMAD4', 'NF1',
'CDKN2A', 'VEGFA', 'BRAF', 'CDKN1A', 'CYP19A1', 'TERT',
'HRAS', 'MET', 'RB1', 'TGFBR2', 'STAT3', 'RAD50', 'MTOR',
'CASP8', 'IGF2', 'PGR', 'FGFR1', 'FAS', 'KIT', 'AXIN2',
'ALK', 'FHIT', 'CXCR4', 'RAF1', 'AURKA', 'MUC1', 'AKT2',
'PIK3R1', 'PPM1D', 'PDGFRB', 'BMPR1A', 'TYMP']
```

```
=======================Generation 5  =======================
```

Generation 5 Best Fitness 17823.0

```
['BRCA2', 'BRCA1', 'TP53', 'PALB2', 'CHEK2', 'CDH1', 'ATM',
'ERBB2', 'BRIP1', 'BARD1', 'ESR1', 'PIK3CA', 'NBN',
'RAD51C', 'KRAS', 'CCND1', 'STK11', 'SMAD4', 'NF1',
'CDKN2A', 'VEGFA', 'BRAF', 'CDKN1A', 'CYP19A1', 'TERT',
'HRAS', 'MET', 'RB1', 'TGFBR2', 'STAT3', 'RAD50', 'MTOR',
'CASP8', 'IGF2', 'PGR', 'FGFR1', 'FAS', 'KIT', 'AXIN2',
'ALK', 'FHIT', 'CXCR4', 'RAF1', 'AURKA', 'MUC1', 'AKT2',
'PIK3R1', 'PPM1D', 'PDGFRB', 'BMPR1A', 'TYMP']
```

```
=======================Generation 6  =======================


 Generation 6 Best Fitness 17927.0

['BRCA2', 'BRCA1', 'TP53', 'PALB2', 'CHEK2', 'CDH1', 'ATM',
'ERBB2', 'BRIP1', 'BARD1', 'ESR1', 'PIK3CA', 'NBN',
'RAD51C', 'KRAS', 'CCND1', 'STK11', 'SMAD4', 'NF1',
'CDKN2A', 'VEGFA', 'BRAF', 'CDKN1A', 'CYP19A1', 'TERT',
'HRAS', 'MET', 'RB1', 'TGFBR2', 'STAT3', 'RAD50', 'MTOR',
'CASP8', 'IGF2', 'PGR', 'FGFR1', 'FAS', 'KIT', 'AXIN2',
'ALK', 'FHIT', 'CXCR4', 'RAF1', 'AURKA', 'MUC1', 'AKT2',
'PIK3R1', 'PPM1D', 'PDGFRB', 'BMPR1A', 'TYMP']

=======================Generation 7  =======================


 Generation 7 Best Fitness 17927.0

['BRCA2', 'BRCA1', 'TP53', 'PALB2', 'CHEK2', 'CDH1', 'ATM',
'ERBB2', 'BRIP1', 'BARD1', 'ESR1', 'PIK3CA', 'NBN',
'RAD51C', 'KRAS', 'CCND1', 'STK11', 'SMAD4', 'NF1',
'CDKN2A', 'VEGFA', 'BRAF', 'CDKN1A', 'CYP19A1', 'TERT',
'HRAS', 'MET', 'RB1', 'TGFBR2', 'STAT3', 'RAD50', 'MTOR',
'CASP8', 'IGF2', 'PGR', 'FGFR1', 'FAS', 'KIT', 'AXIN2',
'ALK', 'FHIT', 'CXCR4', 'RAF1', 'AURKA', 'MUC1', 'AKT2',
'PIK3R1', 'PPM1D', 'PDGFRB', 'BMPR1A', 'TYMP']


=======================Generation 8  =======================


 Generation 8 Best Fitness 17927.0

['BRCA2', 'BRCA1', 'TP53', 'PALB2', 'CHEK2', 'CDH1', 'ATM',
'ERBB2', 'BRIP1', 'BARD1', 'ESR1', 'PIK3CA', 'NBN',
'RAD51C', 'KRAS', 'CCND1', 'STK11', 'SMAD4', 'NF1',
'CDKN2A', 'VEGFA', 'BRAF', 'CDKN1A', 'CYP19A1', 'TERT',
'HRAS', 'MET', 'RB1', 'TGFBR2', 'STAT3', 'RAD50', 'MTOR',
'CASP8', 'IGF2', 'PGR', 'FGFR1', 'FAS', 'KIT', 'AXIN2',
'ALK', 'FHIT', 'CXCR4', 'RAF1', 'AURKA', 'MUC1', 'AKT2',
'PIK3R1', 'PPM1D', 'PDGFRB', 'BMPR1A', 'TYMP']
```

```
======================= Generation 9 ========================


 Generation 9 Best Fitness 17927.0

['BRCA2', 'BRCA1', 'TP53', 'PALB2', 'CHEK2', 'CDH1', 'ATM',
'ERBB2', 'BRIP1', 'BARD1', 'ESR1', 'PIK3CA', 'NBN',
'RAD51C', 'KRAS', 'CCND1', 'STK11', 'SMAD4', 'NF1',
'CDKN2A', 'VEGFA', 'BRAF', 'CDKN1A', 'CYP19A1', 'TERT',
'HRAS', 'MET', 'RB1', 'TGFBR2', 'STAT3', 'RAD50', 'MTOR',
'CASP8', 'IGF2', 'PGR', 'FGFR1', 'FAS', 'KIT', 'AXIN2',
'ALK', 'FHIT', 'CXCR4', 'RAF1', 'AURKA', 'MUC1', 'AKT2',
'PIK3R1', 'PPM1D', 'PDGFRB', 'BMPR1A', 'TYMP']

======================= Generation 10 ======================


 Generation 10 Best Fitness 17927.0

['BRCA2', 'BRCA1', 'TP53', 'PALB2', 'CHEK2', 'CDH1', 'ATM',
'ERBB2', 'BRIP1', 'BARD1', 'ESR1', 'PIK3CA', 'NBN',
'RAD51C', 'KRAS', 'CCND1', 'STK11', 'SMAD4', 'NF1',
'CDKN2A', 'VEGFA', 'BRAF', 'CDKN1A', 'CYP19A1', 'TERT',
'HRAS', 'MET', 'RB1', 'TGFBR2', 'STAT3', 'RAD50', 'MTOR',
'CASP8', 'IGF2', 'PGR', 'FGFR1', 'FAS', 'KIT', 'AXIN2',
'ALK', 'FHIT', 'CXCR4', 'RAF1', 'AURKA', 'MUC1', 'AKT2',
'PIK3R1', 'PPM1D', 'PDGFRB', 'BMPR1A', 'TYMP']
```

Output for Generation 390-400

```
=======================Generation 390=======================


 Generation 390 Best Fitness 18214.0

['BRCA2', 'BRCA1', 'PALB2', 'CHEK2', 'CDH1', 'ERBB2',
'PTEN', 'MSH6', 'MLH1', 'APC', 'ESR1', 'PIK3CA', 'RAD51D',
'MSH2', 'NBN', 'RAD51C', 'KRAS', 'CCND1', 'STK11', 'CTNNB1',
'SMAD4', 'NF1', 'CDKN2A', 'VEGFA', 'BRAF', 'MYC', 'CDK4',
'POLE', 'CDKN1A', 'CYP19A1', 'CDKN1B', 'HRAS', 'TGFBR2',
'EGF', 'STAT3', 'BAX', 'TNF', 'ESR2', 'MTOR', 'CASP8',
'IL6', 'PLAU', 'PMS2', 'FGFR1', 'NOTCH1', 'FAS', 'KIT', 'EP-
CAM', 'SRC', 'ALK', 'EP300']

=======================Generation 391=======================


 Generation 391 Best Fitness 18586.0

['BRCA2', 'BRCA1', 'PALB2', 'CHEK2', 'CDH1', 'ERBB2',
'PTEN', 'MSH6', 'MLH1', 'APC', 'ESR1', 'PIK3CA', 'RAD51D',
'MSH2', 'NBN', 'RAD51C', 'KRAS', 'CCND1', 'STK11', 'CTNNB1',
'SMAD4', 'NF1', 'CDKN2A', 'VEGFA', 'BRAF', 'MYC', 'CDK4',
'POLE', 'CDKN1A', 'CYP19A1', 'CDKN1B', 'HRAS', 'TGFBR2',
'EGF', 'STAT3', 'BAX', 'TNF', 'ESR2', 'MTOR', 'CASP8',
'IL6', 'PLAU', 'PMS2', 'FGFR1', 'NOTCH1', 'FAS', 'KIT', 'EP-
CAM', 'SRC', 'ALK', 'EP300']

=======================Generation 392=======================


 Generation 392 Best Fitness 18586.0

['BRCA2', 'BRCA1', 'PALB2', 'CHEK2', 'CDH1', 'ERBB2',
'PTEN', 'MSH6', 'MLH1', 'APC', 'ESR1', 'PIK3CA', 'RAD51D',
'MSH2', 'NBN', 'RAD51C', 'KRAS', 'CCND1', 'STK11', 'CTNNB1',
'SMAD4', 'NF1', 'CDKN2A', 'VEGFA', 'BRAF', 'MYC', 'CDK4',
'POLE', 'CDKN1A', 'CYP19A1', 'CDKN1B', 'HRAS', 'TGFBR2',
'EGF', 'STAT3', 'BAX', 'TNF', 'ESR2', 'MTOR', 'CASP8',
'IL6', 'PLAU', 'PMS2', 'FGFR1', 'NOTCH1', 'FAS', 'KIT', 'EP-
CAM', 'SRC', 'ALK', 'EP300']
```

```
=======================Generation 393=======================
```

Generation 393 Best Fitness 18586.0

```
['BRCA2', 'BRCA1', 'PALB2', 'CHEK2', 'CDH1', 'ERBB2',
'PTEN', 'MSH6', 'MLH1', 'APC', 'ESR1', 'PIK3CA', 'RAD51D',
'MSH2', 'NBN', 'RAD51C', 'KRAS', 'CCND1', 'STK11', 'CTNNB1',
'SMAD4', 'NF1', 'CDKN2A', 'VEGFA', 'BRAF', 'MYC', 'CDK4',
'POLE', 'CDKN1A', 'CYP19A1', 'CDKN1B', 'HRAS', 'TGFBR2',
'EGF', 'STAT3', 'BAX', 'TNF', 'ESR2', 'MTOR', 'CASP8',
'IL6', 'PLAU', 'PMS2', 'FGFR1', 'NOTCH1', 'FAS', 'KIT', 'EP-
CAM', 'SRC', 'ALK', 'EP300']
```

```
=======================Generation 394=======================
```

Generation 394 Best Fitness 18586.0

```
['BRCA2', 'BRCA1', 'PALB2', 'CHEK2', 'CDH1', 'ERBB2',
'PTEN', 'MSH6', 'MLH1', 'APC', 'ESR1', 'PIK3CA', 'RAD51D',
'MSH2', 'NBN', 'RAD51C', 'KRAS', 'CCND1', 'STK11', 'CTNNB1',
'SMAD4', 'NF1', 'CDKN2A', 'VEGFA', 'BRAF', 'MYC', 'CDK4',
'POLE', 'CDKN1A', 'CYP19A1', 'CDKN1B', 'HRAS', 'TGFBR2',
'EGF', 'STAT3', 'BAX', 'TNF', 'ESR2', 'MTOR', 'CASP8',
'IL6', 'PLAU', 'PMS2', 'FGFR1', 'NOTCH1', 'FAS', 'KIT', 'EP-
CAM', 'SRC', 'ALK', 'EP300']
```

```
=======================Generation 395=======================
```

Generation 395 Best Fitness 18986.0

```
['BRCA2', 'BRCA1', 'PALB2', 'CHEK2', 'CDH1', 'ERBB2',
'PTEN', 'MSH6', 'MLH1', 'APC', 'ESR1', 'PIK3CA', 'RAD51D',
'MSH2', 'NBN', 'RAD51C', 'KRAS', 'CCND1', 'STK11', 'CTNNB1',
'SMAD4', 'NF1', 'CDKN2A', 'VEGFA', 'BRAF', 'MYC', 'CDK4',
'POLE', 'CDKN1A', 'CYP19A1', 'CDKN1B', 'HRAS', 'TGFBR2',
'EGF', 'STAT3', 'BAX', 'TNF', 'ESR2', 'MTOR', 'CASP8',
'IL6', 'PLAU', 'PMS2', 'FGFR1', 'NOTCH1', 'FAS', 'KIT', 'EP-
CAM', 'SRC', 'ALK', 'EP300']
```

=======================Generation 396=======================

Generation 396 Best Fitness 18986.0

['BRCA2', 'BRCA1', 'PALB2', 'CHEK2', 'CDH1', 'ERBB2', 'PTEN', 'MSH6', 'MLH1', 'APC', 'ESR1', 'PIK3CA', 'RAD51D', 'MSH2', 'NBN', 'RAD51C', 'KRAS', 'CCND1', 'STK11', 'CTNNB1', 'SMAD4', 'NF1', 'CDKN2A', 'VEGFA', 'BRAF', 'MYC', 'CDK4', 'POLE', 'CDKN1A', 'CYP19A1', 'CDKN1B', 'HRAS', 'TGFBR2', 'EGF', 'STAT3', 'BAX', 'TNF', 'ESR2', 'MTOR', 'CASP8', 'IL6', 'PLAU', 'PMS2', 'FGFR1', 'NOTCH1', 'FAS', 'KIT', 'EPCAM', 'SRC', 'ALK', 'EP300']

=======================Generation 397=======================

Generation 397 Best Fitness 18986.0

['BRCA2', 'BRCA1', 'PALB2', 'CHEK2', 'CDH1', 'ERBB2', 'PTEN', 'MSH6', 'MLH1', 'APC', 'ESR1', 'PIK3CA', 'RAD51D', 'MSH2', 'NBN', 'RAD51C', 'KRAS', 'CCND1', 'STK11', 'CTNNB1', 'SMAD4', 'NF1', 'CDKN2A', 'VEGFA', 'BRAF', 'MYC', 'CDK4', 'POLE', 'CDKN1A', 'CYP19A1', 'CDKN1B', 'HRAS', 'TGFBR2', 'EGF', 'STAT3', 'BAX', 'TNF', 'ESR2', 'MTOR', 'CASP8', 'IL6', 'PLAU', 'PMS2', 'FGFR1', 'NOTCH1', 'FAS', 'KIT', 'EPCAM', 'SRC', 'ALK', 'EP300']

=======================Generation 398=======================

Generation 398 Best Fitness 18986.0

['BRCA2', 'BRCA1', 'PALB2', 'CHEK2', 'CDH1', 'ERBB2', 'PTEN', 'MSH6', 'MLH1', 'APC', 'ESR1', 'PIK3CA', 'RAD51D', 'MSH2', 'NBN', 'RAD51C', 'KRAS', 'CCND1', 'STK11', 'CTNNB1', 'SMAD4', 'NF1', 'CDKN2A', 'VEGFA', 'BRAF', 'MYC', 'CDK4', 'POLE', 'CDKN1A', 'CYP19A1', 'CDKN1B', 'HRAS', 'TGFBR2', 'EGF', 'STAT3', 'BAX', 'TNF', 'ESR2', 'MTOR', 'CASP8', 'IL6', 'PLAU', 'PMS2', 'FGFR1', 'NOTCH1', 'FAS', 'KIT', 'EPCAM', 'SRC', 'ALK', 'EP300']

=======================Generation 399========================

Generation 399 Best Fitness 18986.0

['BRCA2', 'BRCA1', 'PALB2', 'CHEK2', 'CDH1', 'ERBB2', 'PTEN', 'MSH6', 'MLH1', 'APC', 'ESR1', 'PIK3CA', 'RAD51D', 'MSH2', 'NBN', 'RAD51C', 'KRAS', 'CCND1', 'STK11', 'CTNNB1', 'SMAD4', 'NF1', 'CDKN2A', 'VEGFA', 'BRAF', 'MYC', 'CDK4', 'POLE', 'CDKN1A', 'CYP19A1', 'CDKN1B', 'HRAS', 'TGFBR2', 'EGF', 'STAT3', 'BAX', 'TNF', 'ESR2', 'MTOR', 'CASP8', 'IL6', 'PLAU', 'PMS2', 'FGFR1', 'NOTCH1', 'FAS', 'KIT', 'EPCAM', 'SRC', 'ALK', 'EP300']

All Generations Best Fitness 18986.0
All Generations Best Fitness Community

['BRCA2', 'BRCA1', 'PALB2', 'CHEK2', 'CDH1', 'ERBB2', 'PTEN', 'MSH6', 'MLH1', 'APC', 'ESR1', 'PIK3CA', 'RAD51D', 'MSH2', 'NBN', 'RAD51C', 'KRAS', 'CCND1', 'STK11', 'CTNNB1', 'SMAD4', 'NF1', 'CDKN2A', 'VEGFA', 'BRAF', 'MYC', 'CDK4', 'POLE', 'CDKN1A', 'CYP19A1', 'CDKN1B', 'HRAS', 'TGFBR2', 'EGF', 'STAT3', 'BAX', 'TNF', 'ESR2', 'MTOR', 'CASP8', 'IL6', 'PLAU', 'PMS2', 'FGFR1', 'NOTCH1', 'FAS', 'KIT', 'EPCAM', 'SRC', 'ALK', 'EP300']

6.CONCLUSION AND FUTURE WORK

6.1 Conclusion

Our method tries to evolve a set of candidate genes with as much collaboration with the whole set of known disease genes as possible. It starts from randomly created communities of genes and optimizes the communities over generations, based on the modularity of the community. The idea behind scoring system that we used is that the more frequently a gene is chosen to be in the optimizing communities, i.e. the more a gene is selected for the more optimized communities of the higher generations, the more it is associated with the known disease genes, and therefore should have a higher score compared to others that are selected less often. These candidates Genes obtained at the end are the genes which are most likely to be causing breast cancer.

Although protein-protein interaction networks have been shown to be among the most powerful pieces of evidence for disease-gene association, there are still major concerns about the amount, accuracy, and quality of the available data, and there are still a considerable amount of interactions that are not well-studied. To obtain more accurate disease genes communities one may use a combination of different network-based data resources along with PPI. It should be noted nonetheless that this is a proof of concept and that much further study remains. Most importantly, this method should be applied to other diseases.

Genes associated with a specific disease may act in separate communities which work with one another or separate communities which overlap. We intuitively believe that our GA-based computation is capable of finding disease genes working in different communities or in overlapped communities for the following reasons. First, as the evolving populations contain thousands of different communities, different communities which work with the known disease genes would have the chance to evolve and be in the population at the same time. Genes in all such communities will get high scores as they are often selected in a number of the population's communities over generations, thereby they can increase their scores. Secondly, potential disease genes which are present in more than one community working with disease genes are selected more frequently as they can have a major chance to be in many of the communities of the populations.

6.2 Future Work

We have tested our computational approach on 16 known breast cancer genes taken from similar experiments performed by Xuebing et al [5] to test the performance of algorithm CIPHER which is a well-known disease gene prioritization framework. The results are satisfactory, however, there should be specific and more accurate strategies to determine the known disease gene set with regarding different diseases and choosing appropriate GA parameters. Concisely speaking, appropriate strategies and different parameter values may vary significantly from one disease to another.

REFERENCES

[1] Pietro Cinaglia, Pietro H Guzzi, Pierangelo Veltri (2018), “INTEGRO: an algorithm for data-integration and disease-gene association”, 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM).

[2] Koosha Tahmasebipour, Sheridan Houghten (2014) “Disease-Gene Association Using a Genetic Algorithm”, 14th International Conference on Bioinformatics and Bioengineering.

[3] Eileen Marie, Hanna Nazar, M. Zaki (2015) “Gene-disease association through topological and biological feature integration”, 11th International Conference on Innovations in Information Technology.

[4] Junmin Zhao, Tingting He, Xiaohua Hu, Yan Wang, Xianjun Shen, Minghong Fang, Jie Yuan (2014) “A Novel Disease Gene Prediction Method Based on PPI Network”, IEEE International Conference on Bioinformatics and Biomedicine (BIBM).

[5] Xuebing Wu, Rui Jiang, Michael Q Zhang, and Shao Li(2008) “Networkbased global inference of human disease genes” Molecular systems biology, 4(1).

[6] Bioinformatics. https://en.wikipedia.org/wiki/Bioinformatics [Online; Accessed on April 2020]

[7] Genetic Algorithm Introduction https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3 [Online; Accessed on December 2019]

[8] Genetic Algorithms https://www.geeksforgeeks.org/genetic-algorithms/ [Online; Accessed on December 2019]