Journal of Engineering and Applied Sciences 13 (23): 10068-10079, 2018
ISSN: 1816-949X

A Novel Approach to Classification of Gene Expression Datasets Using Computational Intelligence Techniques

[1]Ramachandro Majji and [2]Bhramaramba Ravi
[1]Department of Computer Science and Engineering,
GMR Institute of Technology, Rajam, Andhra Pradesh (AP), India
[2]Department of Information Technology, GITAM Institute of Technology,
Visakhapatnam, Andhra Pradesh (AP), India

Abstract: The main focus of the proposed system is to handle gene expression datasets of the cancer patients. As per the demography study, the maximum females case suffering with cancer which could be traced by the genes available in their blood, samples. An algorithm is proposed in such a way that which infers in directly predicting the probe set value which indicates at which gene levels, the human is suffering from the research entitled above. The proposed system obtained good result when there is relevant data in the knowledge base, but if failed when there is out loss in data for avoiding this problem, the proposed algorithm is a modified and extended approach of Particle Swarm Optimization (PSO) to find out the exact optimistic gene from which the patient is been lead to cancer at particular levels. An introduced concept namely SIFTS parameter where identification of gene levels are traced at 65-95%. The subsequent outputs which are obtained are giving better results compared with previous research. For this analysis we had considered Poor Differentiated normal cancer (PD) from CPDR datasets supported by IRC and authorized WAMPR, US. The data is consistent and provided better results.

Key words: Gene expression data, prediction, classification, categorization, optimistic, sift parameters, Poor Differentiated normal cancer (PD)

INTRODUCTION

The term bioinformatics (De Souto *et al*., 2012; Jaskowiak *et al*., 2012; Ribeiro *et al*., 2010) was coined by Paulien Hogeweg for the study of informatic processes in bioinformatics stems. It was used in genomics and genetics, particularly in those areas of genomics involving large-scale DNA sequencing. Bioinformatics can be defined as the application of computer technology to the management of biological information. Bioinformatics is the science of storing, extracting, organizing, analyzing, interpreting and utilizing information from biological sequences and molecules. It has been mainly fueled by advances in DNA sequencing and mapping techniques. The primary goal of bioinformatics (Robinson and Speed, 2007) is to increase the understanding of biological processes. Some of the grand area of research in bioinformatics includes:

Sequence analysis of gene: Sequence analysis is the most primitive operation in computational biology and for this proposed approach. This operation consists of finding the part of the biological sequences which are alike and differ during medical analysis and genome mapping processes. The sequence analysis implies subjecting a DNA or peptide, sequence databases, repeated sequence searches or other bioinformatics methods on a computer which is the major study in this research.

Genome annotation: In the context of genomics, annotation is the process of marking the genes and other biological features in a DNA sequence for producing elaborated datasets which are useful in making or creating an occurrence to be happened. A smarter approach is added to the convolution method proposed, so that, it yields annotations results in high.

Analysis of mutations in cancer: In cancer, the genomes of affected cells are rearranged in complex or even unpredictable ways. Massive sequencing efforts are used to identify previously unknown mutations in a variety of genes in cancer. Bioinformaticians continue to produce specialized automated systems to manage the sheer volume of sequence data produced and created new algorithms and software's to compare the sequencing results to the growing collection of human genome

Corresponding Author: Ramachandro Majji, Department of Computer Science and English, GMR Institute of Technology, Rajam, Andhra Pradesh (AP), India

sequences and germline polymorphisms. New physical detection technologies are employed such as oligonucleotide microarrays (Bose *et al*., 2013; Jaskowiak *et al*., 2013) to identify chromosomal gains and losses and single-nucleotide polymorphism arrays to detect known point mutations.

Applications of bioinformatics:

- Genomics/proteomics and X-omics
- Computer aided drug design
- Computational resources
- Applications and databases
- Information technology and management
- Drug design and development

Gene expression data

Gene expression dataset (Costa *et al*., 2002, 2004, 2009): In this module the dataset is defined. Sequence of actions to be carried out on those datasets is defined amongst the file for reading, loading and connecting to dataset repository. The proposed system has been applied to cancer dataset. In addition, the proposed system has been modified, tuned and tested against Center for Prostate Disease Research (CPDR) cancer datasets to boost the confidence in applying the proposed system to different cancer types and to emphasize the suitability of the proposed system to be applied in this sensitive domain.

Preprocessing module: According to the dataset defined in the gene expression dataset module, preprocessing module prepares the dataset to be manipulated. Preparation includes filtering, thres holding, logarithmic transformation and data normalization. Those procedures are essential to be done before actual classification (Lorena *et al*., 2008, 2012) take place.

Gene selection module: In terms of cancer classification (Lorena *et al*., 2012; Ramachandro and Bramaramba, 2015, 2016, 2017), this massive number of microarray data doesn't bring more discriminative power but they degrade the accuracy of the classifier. Thus, features need to be decreased to a feature subset with the most significant features or genes that are capable of discriminating different classes. So, the objective of feature selection is:

- Gene is a stretch of DNA that encodes information
- Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product
- Expression level of genes in an individual that is measured through microarray
- Gene expression is the process by which the information encoded in a gene is used to direct the assembly of a protection molecule

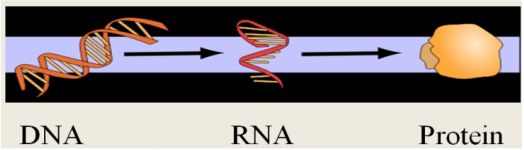


Fig. 1: Converts DNA to RNA and RNA to protein

- Gene expression is explored through a study of protein structure and function, transcription and translation, differentiation and stem cells
- It is the process by which information from a gene is used in the synthesis of functional gene product

Figure 1 shows the DNA makes RNA and RNA makes protein. In higher organisms, the hereditary material, DNA is located in the cell nucleus. The DNA in human cell contains about 100,000 genes, located on 46 chromosomes. The genetic information in the DNA is stored as a sequence of bases called nucleotides.

Figure 2 shows the microarray method used to check gene expression and the microarrays of DNA chips use thin glass microscopic slide or silicon chip.

Figure 3 shows the microarray technology is used to learn about genes which are expressed differently in a target sample compared to a control sample.

Microarray: DNA microarrays provide a natural medium for the exploration. The model organisms are the first for which comprehensive genome-wide surveys of gene expression patterns or function are possible. The results can be viewed as maps that reflect the order and logic of the genetic program, rather than the physical order of genes on chromosomes.

Exploration of the genome using DNA microarrays and other genome-scale technologies should narrow the gap in our knowledge of gene function and molecular biology between the current favored model organisms and other species.

Figure 4 gives the gene expression analysis using a DNA microarray. In this illustration, CPDR samples are compared.

Using DNA microarrays to study gene expression on a genomic scale: The study of gene expression on a genomic scale is the most obvious opportunity made possible by complete genome sequences of the model organisms and experimentally the straightest forward.

Three characteristics of the regulation of gene expression at the level of transcript abundance account for the great value and appeal of genome-wide surveys of transcript levels:

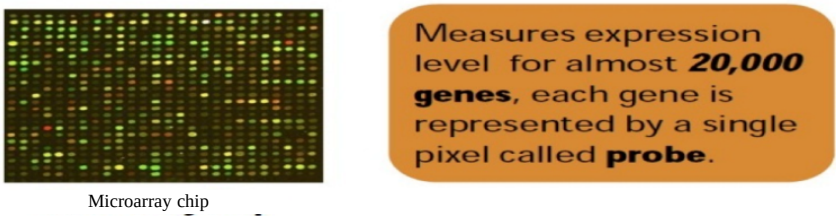


Fig. 2: Microarray chip using gene expression

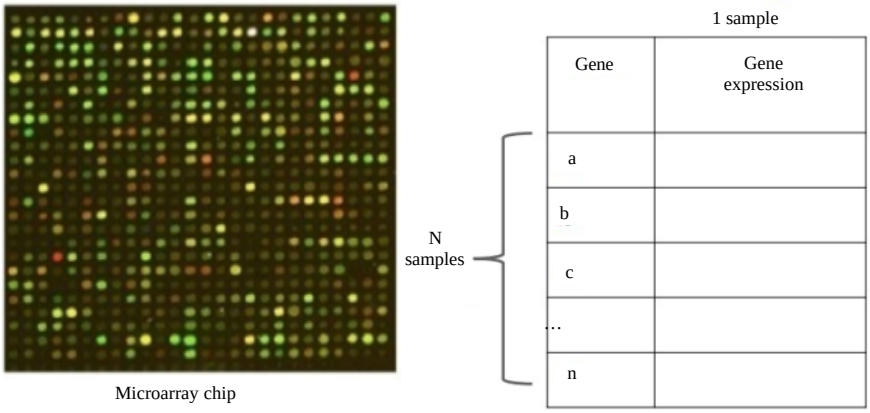


Fig. 3: Microarray chip with gene samples

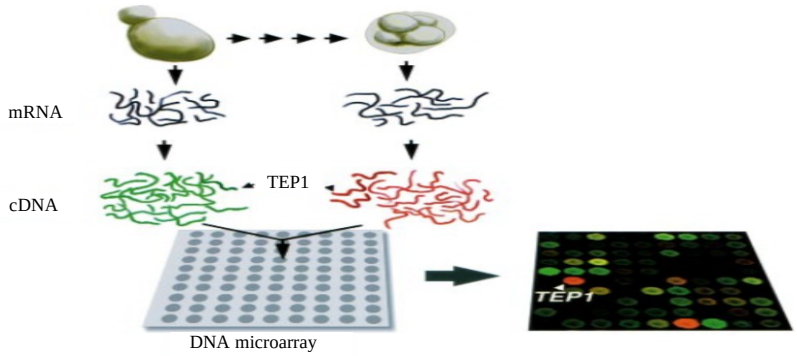


Fig. 4: Conversion using DNA microarray

- Feasible DNA microarrays makes easy to measure the transcripts for every gene at once
- The tight connection between the function of a gene product and its expression pattern
- As a rule, each gene is expressed in the specific cells and under the specific conditions in which its product makes a contribution to fitness

Literature review: A systematic survey of gene expression in 115 human tissue samples representing 35 different tissue types, using microarrays representing approximately 26,000 different human genes. Unsupervised hierarchical cluster analysis (Costa *et al*., 2009; De Souto *et al*., 2010; Maji and Das, 2012; Schonhuth *et al*., 2009) of the gene-expression patterns in these tissues identified clusters of genes with related biological functions and grouped the tissue specimens in a pattern that reflected their anatomic locations, cellular compositions or physiologic functions.

To survey gene expression across normal human tissues, analyzed 115 normal tissue specimens representing 35 different human tissue types, using microarray representing 26,260 different genes. To explore the relationship among samples and underlying features of gene expression an unsupervised two way hierarchical clustering (Balamurugan *et al*., 2014, 2015; Ramachandro

and Bhramaramba, 2016) method using the 5,592 microarray representing 3,960 different uni gene clusters whose expression varied most across samples overall, tissue samples clustered in large part according to their anatomic locations, cellular compositions or physiologic functions.

The gene expression (Babu *et al*., 2015; Ferreira, 2002; Robinson *et al*., 2010) data for a cancer detecting model using incremental fuzzy mining (Jauhari and Rizvi, 2014) based on the study of gene function to determine if a cell or tissue can go cancerous. Though, this test has been performed on each and every cell in the body to ensure a total result.

The gene expression pattern (Babu *et al*., 2015) of a tumor can provide a distinctive molecular portrait recognizable in successive samples over time and in metastases. How distinctive and consistent are the gene expression patterns in individual hepatic cellular carcinomas.

The expression of many genes can be determined by measuring cancer levels with various techniques such as microarrays, Expressed Sequence Tag (EST) sequencing, Serial Analysis of Gene Expression (SAGE) tag sequencing, Massively Parallel Signature Sequencing (MPSS) or various applications of multiplexed in-situ hybridization, etc. All of these techniques are extremely noise-prone and subject to bias in the biological measurement.

Here, the major research area involves developing statistical tools to separate noise in high-throughput gene expression studies. Gene expression is measured in many ways including protein expression, however, protein expression is one of the best clues of actual gene activity since proteins are usually final catalysts of cell activity.

Protein microarrays and high throughput mass spectrometry can provide a snapshot of the proteins present in a biological sample. Bioinformatics is very much involved in making sense of protein microarray and HT-MS data.

Several methods have been proposed in the literature, however: It is not clear how the different techniques compare with each other with respect to the biological relevance of the clusters as well as with other characteristics such as robustness and sensitivity to noise.

No guidelines concerning the choice of the methods are currently available. There are several options used in MATLAB Software (Prasad *et al*., 2016a, b) to identify the clustering of microarray data, import data, normalization and standardization using clustering techniques but there is no information on visualization of plots based on parallel co-ordination methods.

The analysis is based on the subsets of groups and first to classify the data in clustering method and comparing these two groups by normalized data to standardizing the data

Cancer datasets overview: Cancer is leading cause of death worldwide accounting to 14 million new cases and 8.2 million deaths every year. But today's modern science can boast of know-how to treat cancer but only if detected in early stages. The bottom line is that survivability depends on early detection. There are modern methodologies within medical sciences and information technology together which can detect cancer but only after it has assumed a fatal form.

As most methods rely upon matching known cancer strands with samples, they are capable of detection but fail to predict. Further, they fail to detect any unknown strands of cancer. A study at the gene level can provide sufficient information to predict and diagnose cancer even before it has affected body. Later, the gene study does not rely upon exact pattern matching behavior pattern for prediction and detection, it has a better chance of detecting any previously unknown forms of cancer. Hence, it is of utmost importance while predicting cancer samples are studied at gene level to ensure proper and correct prediction.

Types of cancer:

- Cancer through tumors (for the proposed model)
- Lung cancer
- Breast cancer
- Colon and rectal cancer
- Endometrial cancer
- Pancreatic cancer
- Kidney cancer
- Prostate cancer
- Thyroid cancer
- Leukemia

Features of cancer datasets considered in this approach: These are the micro array data representation of the information collected from Poorly Differentiated (PD) datasets (Balamurugan *et al*., 2014). These datasets in each tuple represents a unique probe set.

The probe sets would be having similar kind of gene bank (Sequence ID) based on the PD attribute values entered.

The most probably the databases are collected from CPDR gene expression data (Vadamodula *et al*., 2018) for tumor affected persons.

CPDR tumor-benign 80 gene chip dataset from 40 patients specimens were obtained under an IRB-approved

Table 1: Feature attributes considered for the design of user interface as shown in study

Abbreviation	Name of the attribute	Database used name	Scientific name as in dataset
HGU133A	Oligonucleotide human genome	S1	21-N-PD
EST	Expressed sequence tag	S2	22-N-PD
HG	Hypothetical genes	S3	23-N-PD
BUTP	Biotinylated UTP	S4	24-N-PD
BCTP	Biotinylated CTP	S5	25-N-PD
QIAGEN	QIAGEN RNeasy	S6	26-N-PD
CRNA	Biotin cRNA	S7	27-N-PD
HHGU133A	Hybridized HG U133A	S8	28-N-PD
COB2	Control oligo B2	S9	29-N-PD
CCRNA	Control cRNA	S10	30-N-PD
FCRNA	Fragmented cRNA	S11	31-N-PD
RSH	Rotisserie hybridization	S12	32-N-PD
FSP	Fluidics station protocol	S13	33-N-PD
SSPET5	0.5 x SSPE-T	S14	34-N-PD
TX	Triton X-100	S15	35-N-PD
RSO	Rotisserie Oven	S16	36-N-PD
SPH	Streptavidin phycoerythrin	S17	37-N-PD
SSPET6	6 X SSPE-T buffer	S18	38-N-PD
SIGMA	Acetylated bovine serum albumin	S19	39-N-PD
GSO	Genearray scanner observation	S20	40-N-PD

protocol from patients treated with Radical Prostatectomy (RP) at Walter Reed Army Medical Center (WRAMC).

From over 300 patients two groups were selected which had prostate tumors with either Well Differentiated (WD) or Poorly Differentiated (PD) after radical RP.

The PD group had Gleason score 8-9, seminal vesicle invasion and poorly differentiated tumor cells, the Well Differentiated (WD) group had Gleason score 6-7, no seminal vesicle invasion and well to moderately differentiated tumor cells. Compatible specimens were selected from age and race (Caucasians) matched PD or WD patients (Lorena *et al*., 2012). Table 1 feature attributes considered for the design of user interface as shown in this study.

Existing and proposed system

Existing system: In recent years, there have been various efforts to overcome the limitations of standard clustering approaches for the analysis of gene expression (Vadamodula *et al*., 2018) data by grouping genes and samples simultaneously. The underlying concept which is often referred to as bi-clustering, allows to identify sets of genes sharing compatible expression patterns across subsets of samples and its usefulness has been demonstrated for different organisms and datasets.

Proposed system: The main objective is to identify the most promising gene from CPDR cancer datasets (Prasad and Rao, 2015) to predict the role of its own increasing the levels of cancer in human cadaver. A software application is developed which involves the below mentioned categories.

To collect the data from consistent repositories, study and represent the same using microarray data. To organize the microarray data into spatial reference using 30 fold cross validation method for obtaining ranks of CPDR cancer attributes based on its probe sets and later the same is allocated to its referenced classes A-E, respectively .

To reduce the dimensionality of the microarray data using existing PSO Algorithm (Cura, 2012). To reduce the dimensionality of the microarray data using the proposed genetic algorithm with SIFT parameter.

Comparative study to predict the most unfair gene targets for cancer and to identified the path way involvement of the resultant genes and validate their role in causing the cancer disease using generic parameter, that is efficiency, accuracy and precision.

MATERIALS AND METHODS

The net framework: The net framework is a revolutionary platform that helps to write the applications in Windows. The net framework applications are multi-platform applications. The framework has been designed in such a way that it can be used from any of the following languages: C#, C++, Visual Basic, Jscript, COBOL, etc. All these languages can access the framework as well as communicate with each other.

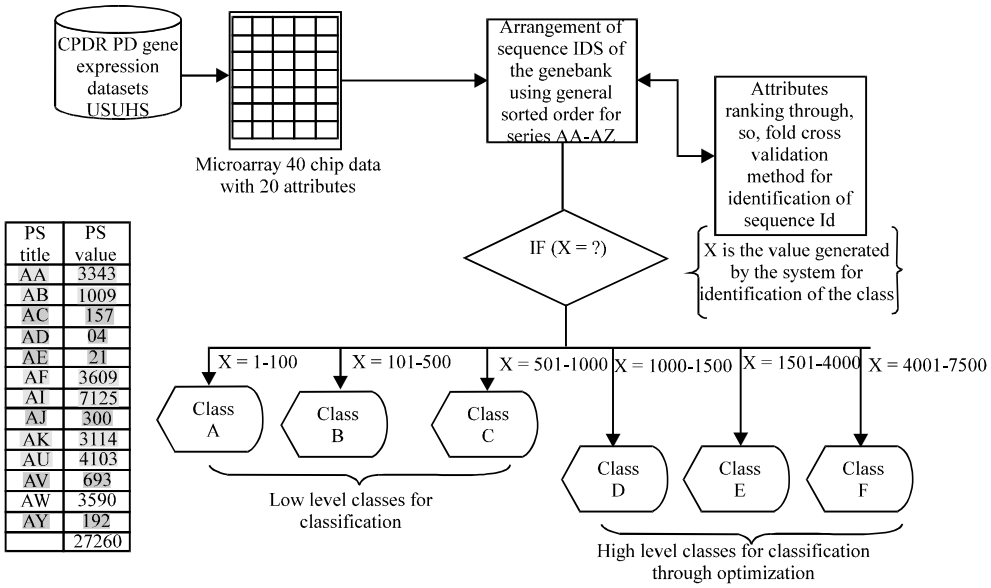
The net framework consists of an enormous library of codes used by the client languages such as C#. Integrated Development Environment (IDE) for C#.

Microsoft provides the following development tools for C# programming:

- Visual Studio 2010 (VS)
- Visual C# 2010 Express (VCE)
- Visual Web Developer

The above tools are considered in developing the software tool proposed. The representation of the same is mentioned in study.

Introduction to Particle Swarm Optimization (PSO): PSO is a heuristic technique (Balamurugan *et al*., 2014) with the latest evolutionary and population-based optimization algorithms (Balamurugan *et al*., 2015; Ramachandro and Bhramaramba, 2016 Vadamodula *et al*., 2018) which can simulate bird flocking or fish schooling behavior. PSO searches for the optimum solution in the search space. A problem-specific fitness function is employed to determine the next search step. Each particle's movement is the composition of a velocity and two randomly



PS title	PS value
AA	3343
AB	1009
AC	157
AD	04
AE	21
AF	3609
AI	7125
AJ	300
AK	3114
AU	4103
AV	693
AW	3590
AY	192
	27260

Fig. 5: Schematic view of the proposed system on pd gene expression datasets for classification

weighted influences. The two randomly-weight influences are individuality or the tendency to return to its best previous position and sociality or the tendency to move towards its neighborhood's best previous position. When compared to many of the other population-based approaches such as other well-known genetic algorithms, the convergence rate of the population is much slower for PSO. Thus, there is no need for any extra mechanism such as the mutation operator in genetic algorithms to diversify different areas of the search space.

PSO algorithm: The PSO algorithm written in pseudo code as follows for the proposed research.

Algorithm 1; Parameters:

A: Population of events. pi: position of agent ai in solution space
f: Objective function. vi: Velocity of agents ai
$V(a_i)$: Neighborhood of agent a_i (fixed)

```
[x*] = PSO()
p = Particle-Initialization();
    For i = 1 to it-max
                For each particle p in P do
                        fp = f (p)
                        if fp beter than f(pbest)
                        pbest = p
            End
    End
            Continued on Next Slide¬
    gbest = best p in P
    For each particle p in P do
                vi = vi+c1*rand()*(pbest-p)+c2*rand()*(gbest-p)
                p = p+v
    End
End algorithm
Particle update rule
```

```
    p = p+v
with
    v = v+c1*rand()*(pbest-p)+c2*rand()(gbest-p)
```

where
p: particle position v: path direction c1: cognitive learning constant
c2: social learning constant (Continued on Next Slide) pbest: best position of the particle gbest: best position of the swarm
rand: a random variable

Figure 5 shows the complete implementation procedure of the 30 fold cross validation method at caring for high and low level classes for classification.

Genetic algorithms: Genetic Algorithm (GA) is a probabilistic algorithm simulating the mechanism of natural selection of living organisms and is often used to solve problems having expensive solutions. In GA, the search space is composed of candidate solutions to the problem each represented by a string is termed as a chromosome. Each chromosome has an objective function value called fitness. A set of chromosomes together with their associated fitness is called the population. This population at a given iteration of the genetic algorithm is called a generation.

Selection rules: Select the individuals called parents which contribute to the population at the next generation.

Crossover rules combine two parents to form children for the next generation. Mutation rules apply random changes to individual parents to form children.

Algorithm 2; implementation of proposed genetic Algorithms Rsearch:

```
/*… Pseudo Code for Genetic Proposed…*/
namespace Simple Genetic Algorithm
{
 public delegate int Fitness Delegate (object chromosome Representation)
 public delegate Chromosome Pair Selection Delegate ()
 public struct Chromosome Pair
        {public Chromosome Parent 1
         public Chromosome Parent 2}
 /*…Pseudo Code for representing Chromosome…*/
public class Chromosome
 {
  public string Chromosome String {set, get}
     public int Calculate Fitness (Fitness Delegate fitness Function)
            {return fitnessFunction(Chromosome String)}
      public static int SumStringCharacter(object chromosomestring)
            {
              int sum = 0; foreach (char c in (string)chromosomestring)
            {
              if (c == '1') { sum++; }
             else if (c == '0') { }
             else
                   {}
             } return sum
   }
/*…Pseudo Code for representing Chromosome…*/
public Chromosome()
{Random r = new Random (Convert. To Int 32 (Date Time. Now. Ticks
% Int16.Max Value)) this.ChromosomeString = String.Empty
             for (int i = 0; i<8; i++)
            {this.ChromosomeString += "" + r.Next() % 2}
            Thread.Sleep(1)
                        }
      }
 /*…Pseudo Code for representing Chromosome Entry…*/
 public Chromosome Mutate(Chromosome entry)
   {
    Random r = new Random (Convert. To Int 32 (Date Time. Now. Ticks
% Int16.MaxValue))
   bool do Mutation = (((r.Next()%100))<Mutation Probability)
              Chromosome ret = new Chromosome()
         ret.ChromosomeString = entry.ChromosomeString; if (doMutation)
    {
        if (entry.ChromosomeString.IndexOf('0') >= 0)
     {
byte[] tmp = get Byte Array From String (entry. Chromosome String) tmp
[entry. Chromosome String. Index Of ('0')] = 1
        ret.ChromosomeString = get String From Byte Array (tmp)
       }
     } return ret
   }
  } }
/*…Pseudo Code for representing Class of Genetic…*/
class Genetic
       {
 public void algorithm()
 {
  Generation Manager manager = new GenerationManager()
   Console.WriteLine("First gen")
   Chromosome[] Gen = manager.CurrentGen
   Console.WriteLine("First Generation Before mutation Childrens")
   List<Chromosome> nextgen = new List<Chromosome>()
                  foreach (ChromosomePair p in pairs)
   {
    Chromosome Pair pair = manager.Crossover (p)
    if (pair.parent2 == null)
{Console. Write Line (pair. parent 1. Chromosome String) nextgen. Add
(pair.parent1)}
  else
   {
     nextgen.Add(pair.parent1)
      nextgen.Add(pair.parent2)
 } } } }
```

RESULTS AND DISCUSSION

Resultant outcomes: Table 2 shows slots representing ther input parameter entry through the user interface developedand the slots clearly differentiates that there is a change in each and every attribute for the change of case which is required for comparitive study (Fig. 6 and 7).

Table 3 shows slots representing ther input parameter entry through the user interface developedand the slots clearly differentiates that there is a change in each and every attribute for the change of case which is required for comparitive study. Color red indicates the change in

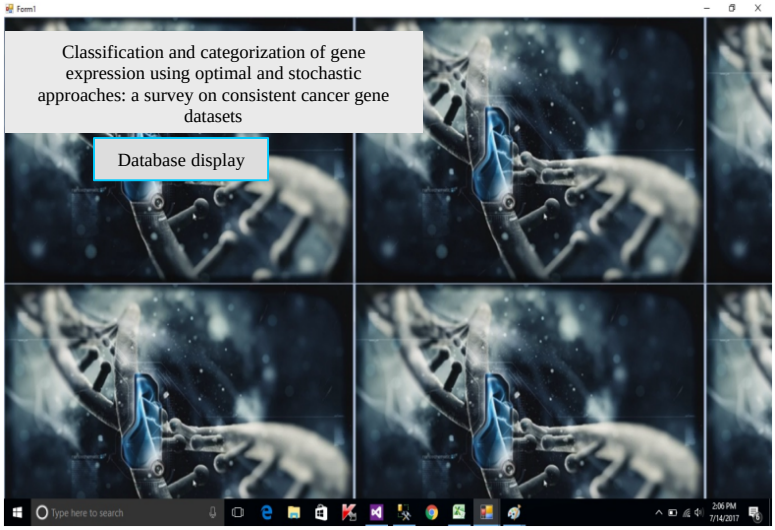


Fig. 6: The home page of the proposed software tool on a standalone system

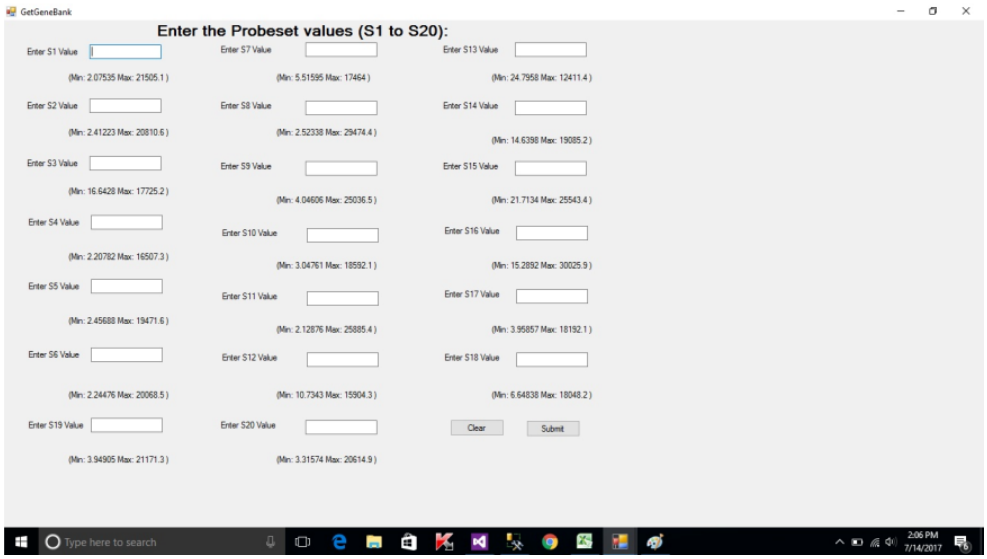


Fig. 7: The above home pages shows the to enter the input values through the keyboard of 20 attributes

Table 2: Entry of user interface elements representing the elements 21-30

Case	S1-(21-CaseN-PD)	S2 (22-N-PD)	S3 (23-N-PD)	S4 (24-N-PD)	S5 (25-N-PD)	S6 (26-N-PD)	S7 (27-N-PD)	S8 (28-N-PD)	S9 (29-N-PD)	S10 (30-N-PD)
1	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741
2	**2.38152**	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741
3	3.38152	**5.53348**	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741
4	3.38152	6.53348	**21.0793**	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741
5	3.38152	6.53348	22.0793	**2.83505**	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741
6	3.38152	6.53348	22.0793	3.83505	**3.68055**	5.35682	7.41108	5.68163	12.4103	6.93741
7	3.38152	6.53348	22.0793	3.83505	4.68055	**4.35682**	7.41108	5.68163	12.4103	6.93741
8	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	**6.41108**	5.68163	12.4103	6.93741
9	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	**4.68163**	12.4103	6.93741
10	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	**11.4103**	6.93741
11	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	**5.93741**
12	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741
13	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741
14	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741
15	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741
16	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741
17	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741
18	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741
19	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741
20	3.38152	6.53348	22.0793	3.83505	4.68055	5.35682	7.41108	5.68163	12.4103	6.93741

Table 3: Entry of user interface elements representing the elements 31-40

Case	S11 (31-N-PD)	S12 (32-N-PD)	S13 (33-N-PD)	S14 (34-N-PD)	S15 (35-N-PD)	S16 (36-N-PD)	S17 (37-N-PD)	S18 (38-N-PD)	S19 (39-N-PD)	S20 (40-N-PD)
1	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
2	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
3	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
4	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
5	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
6	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
7	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
8	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
9	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
10	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
11	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
12	**2.96624**	13.9652	31.0702	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
13	3.96624	**12.9652**	31.0702	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
14	3.96624	13.9652	**30.0702**	22.53	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278
15	3.96624	13.9652	31.0702	**21.53**	49.0503	18.7165	7.15865	11.8721	5.38809	6.01278

Table 3: Continue

Case	S11 (31-N-PD)	S12 (32-N-PD)	S13 (33-N-PD)	S14 (34-N-PD)	S15 (35-N-PD)	S16 (36-N-PD)	S17 (37-N-PD)	S18 (38-N-PD)	S19 (39-N-PD)	S20 (40-N-PD)
16	3.96624	13.9652	31.0702	22.53	**48.0503**	18.7165	7.15865	11.8721	5.38809	6.01278
18	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	**6.15865**	11.8721	5.38809	6.01278
19	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	6.15865	**10.8721**	5.38809	6.01278
20	3.96624	13.9652	31.0702	22.53	49.0503	18.7165	6.15865	11.8721	**4.38809**	6.01278

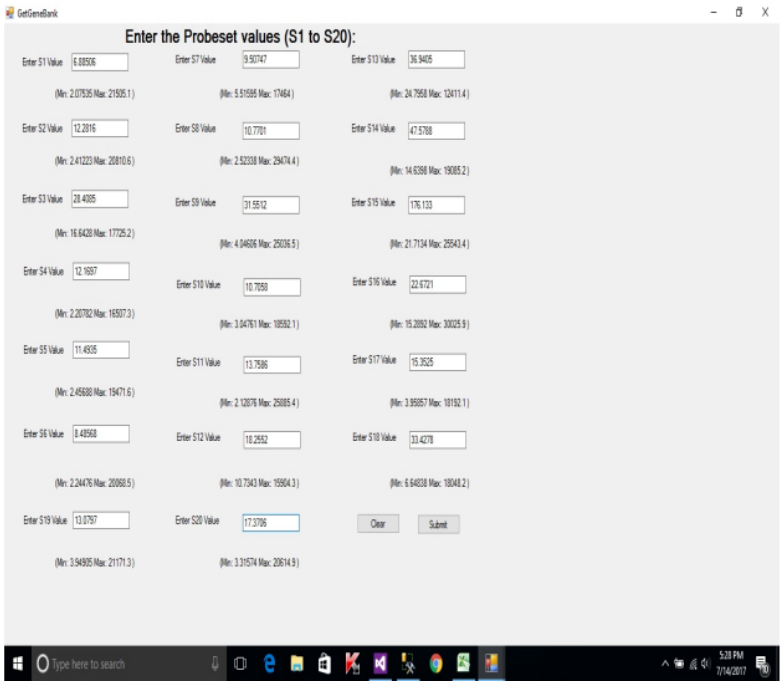


Fig. 8: The above pictures shows the after enter the input values through the keyboard of 20 attributes

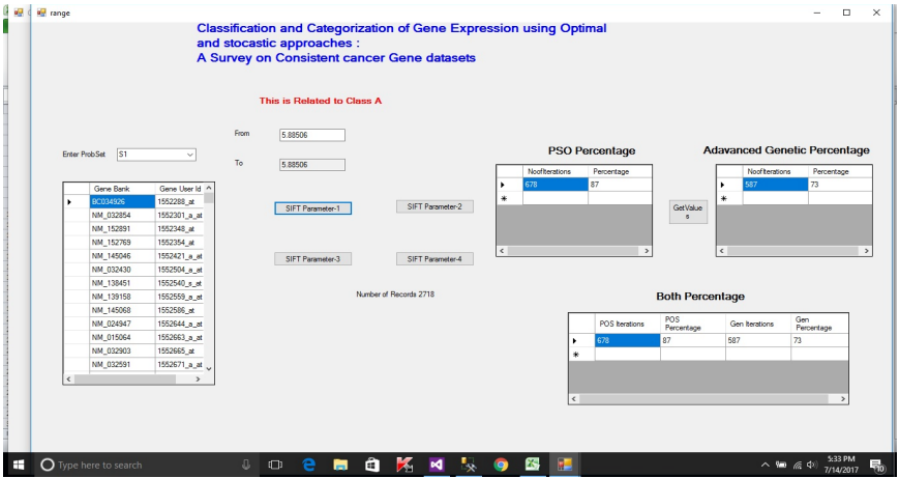


Fig. 9: The above screen shot shows the to change the SIFT parameter values of the giving the minimum values to maximimum value from-to fields

the value which is majorly required for prediction of the conceren classes. Figure 8-10 discribes the user interface approach of Table 2 and 3 as represented.

Table 4 shows slots representing the output parameters obtained through input marked. The records traced vary from one level of SIFT to other levels.

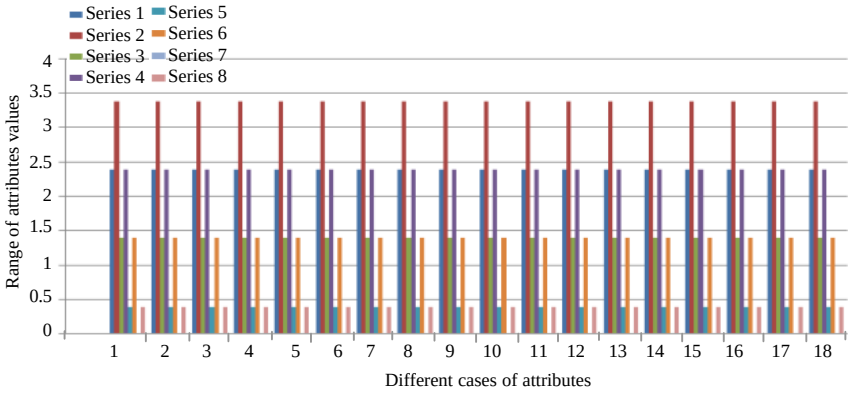


Fig. 10: Reprsentation of the plots obtained evolved in the SIFT parameter chat

Table 4: Change in SIFT parameters

S. No.	No. of record sift-1	No. of record sift-2	No. of record sift-3	No. of record sift-4
1	7958	7948	7933	7918
	•	•	•	•
	•	•	•	•
	•	•	•	•
2	7958	7948	7933	7918
3	7958	7948	7933	7918
4	7958	7948	7933	7918
5	7958	7948	7933	7918
6	7941	7948	7933	7918
7	7941	7948	7933	7918
8	7941	7948	7933	7918
9	7941	7926	7933	7896
10	7941	7926	7933	7896
11	7941	7926	7911	7896
12	7941	7948	7911	7896
	•	•	•	•
	•	•	•	•
	•	•	•	•
13	7941	7926	7911	7896
14	7941	7926	7911	7896
15	7936	7926	7911	7896
16	7936	7926	7911	7896
17	7936	7921	7911	7896
18	7936	7921	7911	7896
Total records traced	**434**	**424**	**409**	-----

CONCLUSION

The proposed research had given surprisingly better solutions for the attributes entered not <50% accurately, towards the provided user interface. As mentioned the accuracy of entering data into User interfaces. When increased to 80 or 90% the data in the knowledge base is not available and the resultant output screen displayed that the number of non matches are max compared with matches. Here, the probe set with higher matches is target and further the probe set is classified into its relevant class. In later stage, the classifier considered the non monotonic value, i.e., by default S1 for tracing the SIFT parameters values obtained at the levels 95, 85, 75 and 65%, respectively. The observations are noted in such a way that the optimistic values either or nor giving better result at stages SIFT-1 or 2 parameters, maximum cases failed at SIFT-3. A levels due the non occurrence of the entered values in the same scope or range. The predictions turned out in such a way that the optimistic values proposed by this algorithms.

RECOMMENDATIONS

The proposed research had given its mile stone outcomes at various scenarios but the same research could be extended in the below said factors. The attributes and the features varies from location to location and hence, it can resource a geographical setting. An evolutionary model of this tool can be prepared in such a way that it could research in any feasible environment. The applicable levels of attainment of the outcomes can be studied by considering not only PSO and Genetic but also with advanced deep learning and machine learning algorithms (Prasad *et al*., 2014, 2016a, b) too. The tool developed could act as a framework which could solve many hypothetical issues developed or generated because of this cancer datasets.

REFERENCES

Babu, K.U., R. Rajeswari and G. GunaSekaran, 2015. A survey on data mining of gene expression data for gene function prediction. Intl. J. Innovative Res. Comput. Commun. Eng., 3: 3445-3451.

Balamurugan, R., A.M. Natarajan and K. Premalatha, 2014. Comparative study on swarm intelligence techniques for biclustering of microarray gene expression data. Intl. J. Comput. Control Quantum Inf. Eng., 8: 323-329.

Balamurugan, R., A.M. Natarajan and K. Premalatha, 2015. Stellar-mass black hole optimization for biclustering microarray gene expression data. Appl. Artif. Intell., 29: 353-381.

Bose, S., C. Das, A. Chakraborty and S. Chattopadhyay, 2013. Effectiveness of different partition based clustering algorithms for estimation of missing values in microarray gene expression data. Proceedings of the 2nd International Conference on Advances in Computing and Information Technology, July 13-15, 2012, Springer, Chennai, India, ISBN:978-3-642-31551-0, pp: 37-47.

Costa, I.G., A.C. Lorena, L.R.Y. Peres and M.C. De Souto, 2009. Using supervised complexity measures in the analysis of cancer gene expression data sets. Proceedings of the 4th Brazilian Symposium on Bioinformatics, July 29-31, 2009, Springer, Porto Alegre, Brazil, ISBN:978-3-642-03222-6, pp: 48-59.

Costa, I.G., F.A.D. Carvalho and Mareilio C.P. de souto, 2004. Comparative analysis of clustering methods for gene expression time course data. Gene. Mol. Biol., 27: 623-631.

Costa, I.G., F.D.A. De Carvalho and M.C.P. De Souto, 2002. A symbolic approach to gene expression time series analysis. Proceedings of the 7th Brazilian Symposium on Neural Networks (SBRN'02), November 11-14, 2002, IEEE, Pernambuco, Brazil, ISBN:0-7695-1709-9, pp: 25-30.

Cura, T., 2012. A particle swarm optimization approach to clustering. Expert Syst. Appl., 39: 1582-1588.

De Souto, M.C., A.C. Lorena, N. Spolaor and I.G. Costa, 2010. Complexity measures of supervised classifications tasks: A case study for cancer gene expression data. Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN'10), July 18-23, 2010, IEEE, Barcelona, Spain, ISBN:978-1-4244-6916-1, pp: 1-7.

De Souto, M.C., A.L. Coelho, K. Faceli, T.C. Sakata and V. Bonadia *et al*., 2012. A comparison of external clustering evaluation indices in the context of imbalanced data sets. Proceedings of the 2012 Brazilian Symposium on Neural Networks (SBRN'12), October 20-25, 2012, IEEE, Curitiba, Brazil, ISBN:978-1-4673-2641-4, pp: 49-54.

Ferreira, C., 2002. Gene Expression Programming in Problem Solving. In: Soft Computing and Industry, Roy, R., M. Koppen, S. Ovaska, T. Furuhashi and F. Hoffmann (Eds.). Springer, London, England, UK., ISBN:978-1-4471-1101-6, pp: 635.

Jaskowiak, P.A., R.J. Campello and I.G. Costa, 2012. Evaluating correlation coefficients for clustering gene expression profiles of cancer. Proceedings of the 7th Brazilian Symposium on Bioinformatics (BSB'12), August 15-17, 2012, Springer, Campo Grande, Brazil, ISBN:978-3-642-31926-6, pp: 120-131.

Jaskowiak, P.A., R.J. Campello and I.G. Costa, 2013. Proximity measures for clustering gene expression microarray data: A validation methodology and a comparative analysis. IEEE. ACM. Trans. Comput. Boil. Bioinf., 10: 845-857.

Jauhari, S. and S.A.M. Rizvi, 2014. Mining gene expression data focusing cancer therapeutics: A digest. IEEE. ACM. Trans. Comput. Biol. Bioinf., 11: 533-547.

Lorena, A.C., I.G. Costa and M.C. De Souto, 2008. On the complexity of gene expression classification data sets. Proceedings of the 8th International Conference on Hybrid Intelligent Systems (HIS'08), September 10-12, 2008, IEEE, Barcelona, Spain, ISBN:978-0-7695-3326-1, pp: 825-830.

Lorena, A.C., I.G. Costa, N. Spolaor and M.C. De Souto, 2012. Analysis of complexity indices for classification problems: Cancer gene expression data. Neurocomput., 75: 33-42.

Maji, P. and C. Das, 2012. Relevant and significant supervised gene clusters for microarray cancer classification. IEEE. Trans. Nanobiosci., 11: 161-168.

Prasad, V. and T.S. Rao, 2015. Implementation of regularization method ridge regression on specific medical datasets. Intl. J. Res. Comput. Appl. Inf. Technol., 3: 25-33.

Prasad, V., T.S. Rao and M.S.P. Babu, 2016b. Thyroid disease diagnosis via hybrid architecture composing rough data sets theory and machine learning algorithms. Soft Comput., 20: 1179-1189.

Prasad, V., T.S. Rao and P.P. Reddy, 2016a. Improvised prophecy using regularization method of machine learning algorithms on medical data. Pers. Med. Universe, 5: 32-40.

Prasad, V., T.S. Rao, A.V. Reddy and B. Chaitanya, 2014. Health diagnosis expert advisory system on trained data sets for hyperthyroid. Intl. J. Comput. Appl., 102: 40-46.

Ramachandro, M. and R. Bhramaramba, 2015. Comparing clustering techniques for gene expression data analysis. Intl. J. Appl. Eng. Res., 10: 10397-10399.

Ramachandro, M. and R. Bhramaramba, 2016. Implementation of classification rule discovery on biological dataset using ant colony optimization. Intl. J. Eng. Technol. Manage., 3: 1-12.

Ramachandro, M. and R. Bramaramba, 2017. Clustering approaches for evaluation and analysis on formal gene expression cancer datasets. Intl. J. Recent Innovation Trends Comput. Commun., 5: 228-235.

Ribeiro, C., T.F. De Assis and I.G. Costa, 2010. Semi-supervised approach for finding cancer sub-classes on gene expression data. Proceedings of the 5th Brazilian Symposium on Bioinformatics, August 31-September 3, 2010, Springer, Rio de Janeiro, Brazil, ISBN:978-3-642-15059-3, pp: 25-34.

Robinson, M.D. and T.P. Speed, 2007. A comparison of Affymetrix gene expression arrays. BMC. Bioinf., 8: 449-1-449-16.

Robinson, M.D., D.J. McCarthy and G.K. Smyth, 2010. Edger: A Bioconductor package for differential expression analysis of digital gene expression data. Bioinf., 26: 139-140.

Schonhuth, A., I.G. Costa and A. Schliep, 2009. Semi-supervised clustering of yeast gene expression data. Proceedings of the Two German-Japanese Workshops on Cooperation in Classification and Data Analysis, May 5, 2009, Springer, Berlin, Germany, ISBN:978-3-642-00667-8, pp: 151-159.

Vadamodula, P., M.P. Rao, V.H. Kumar, S. Radhika and K. Vahini *et al*., 2018. Scrutiny of data sets through procedural algorithms for categorization. Proceedings of the 2018 International Conference on Data Engineering and Intelligent Computing, June 1, 2018, Springer, Singapore, ISBN:978-981-10-3222-6, pp: 437-444.