आई.ए.एस.आर.आई./पी.आर.- 06/2021
I.A.S.R.I./P.R. - 06/2021
संस्थान परियोजना कोड AGEDIASRISIL201900300149
Institute Project Code: AGEDIASRISIL201900300149

परियोजना रिपोर्ट
PROJECT REPORT

विशेषता विशिष्ट जीन पहचान के लिए कार्यप्रणाली का विकास

Development of Methodology for Trait Specific Genes Identification

डॉ. मोहम्मद समीर फ़ारूक़ी — Dr. Mohammad Samir Farooqi
डॉ. द्विजेश चंद्र मिश्र — Dr. Dwijesh Chandra Mishra
डॉ. कृष्ण कुमार चतुर्वेदी — Dr. Krishna Kumar Chaturvedi
डॉ. सुधीर श्रीवास्तव — Dr. Sudhir Srivastava

कृषि जैव सूचना विज्ञान प्रभाग
Division of Agricultural Bioinformatics
भा.कृ.अनु.प.-भारतीय कृषि सांख्यिकी अनुसंधान संस्थान
लाइब्रेरी एवेन्यू, पूसा, नई दिल्ली – 110012
ICAR-Indian Agricultural Statistics Research Institute
Library Avenue, Pusa, New Delhi - 110012
https://iasri.icar.gov.in/
2021

आमुख

नवीन अनुक्रमण/ नेक्स्ट-जेनेरेशन सीक्वेंसिंग (एन.जी.एस.) और माइक्रोएरे जीन अभिव्यक्ति (एक्स्प्रेशन) डेटा का एक लोकप्रिय स्रोत है। एन.जी.एस प्रौद्योगिकी में तेजी और लागत में पर्याप्त कमी ने जीन अभिव्यक्ति डेटा के बड़ी मात्रा में उत्पादन की गति को तेज कर दिया है। जीन अभिव्यक्ति डेटा की लगातार बढ़ती मात्रा ने वैज्ञानिकों और सांख्यिकीविदों पर ऐसी तकनीक और कार्यप्रणाली विकसित करने पर जोर दिया है जो इस जटिल और बड़ी मात्रा में डेटा के विश्लेषण का समर्थन करती है। कई बार, इन डेटासेट में जीन की संख्या नमूनों की संख्या से बहुत अधिक होती है। इसके अलावा, परिणाम से जुड़े प्रासंगिक सूचनात्मक जीन आमतौर पर डेटा सेट में कम होते हैं। इसके अतिरिक्त, विभिन्न जीनों के बीच जटिल संबंध विश्लेषण को और अधिक कठिन बनाते हैं। सूचनात्मक जीन चयन अनावश्यक और अप्रासंगिक जीन को हटाने में एक बड़ी भूमिका निभाता है और परिणाम की गुणवत्ता में सुधार करता है। विभिन्न सूचनात्मक जीन चयन विधियां मौजूद हैं, और उनका व्यापक रूप से उपयोग किया जा रहा है। इन सभी विधियों का उद्देश्य निरर्थक और अप्रासंगिक जीन को हटाना है, ताकि नए घटना (इंस्टान्स) का वर्गीकरण अधिक सटीक हो सके।

ओपन सोर्स के साथ-साथ वाणिज्यिक कम्प्यूटेशनल और सांख्यिकीय टूल्स के विकास में उच्च प्रदर्शन कंप्यूटिंग, समानांतर (पैरलल) प्रोग्रामिंग, बिग डेटा एनालिटिक्स और डेटा विज़ुअलाइज़ेशन का उपयोग करके जैविक डेटा हैंडलिंग और विश्लेषण के लिए उन्नत तकनीकों से युक्त जैविक जटिलता को उजागर करने का आसान तरीका सक्षम किया है। भारतीय कृषि अनुसंधान परिषद (आई.सी.ए.आर.) ने 2010 में भा.कृ.अनु.प.-भारतीय कृषि सांख्यिकी अनुसंधान संस्थान, नई दिल्ली में जैव सूचना विज्ञान और जैविक डेटा संग्रह के विकास में अनुसंधान शुरू करने के लिए कृषि जैव सूचना विज्ञान केंद्र (केबिन) की स्थापना की है जो संस्थान में एक प्रभाग की हैसियत से है। केंद्र में कंप्यूटर अनुप्रयोग, सांख्यिकी और जीव विज्ञान के शोधकर्ता हैं। कम्प्यूटेशनल जीव विज्ञान और कृषि जैव सूचना विज्ञान के माध्यम से देश में जैव प्रौद्योगिकी अनुसंधान को सहायता प्रदान करने के लिए केंद्र को देश में नोडल एजेंसी के रूप में भी पहचाना जाता है।

इस परियोजना में एक नई पद्धति विकसित करने का प्रयास किया गया है जो जीन अभिव्यक्ति डेटा से अनुमानित (प्रेडिक्टीव) मॉडल बनाता है और विशेषता (ट्रेट) विशिष्ट जीन (सूचनात्मक जीन) का चयन करता है जो अत्यधिक प्रासंगिक हैं। इस पद्धति को दो पारंपरिक मशीन लर्निंग एल्गोरिदम, सपोर्ट वेक्टर मशीन (एस.वी.एम.) और जेनेटिक एल्गोरिदम (जी.ए.) के संयोजन का प्रयोग करके विकसित किया गया है।

लेखकगण

PREFACE

Next-generation sequencing (NGS) and microarrays are popular source of gene expression data. The rapid and substantial cost reduction in NGS technology has significantly accelerated the generation of huge amount of gene expression data. The ever increasing amount of gene expression data has put emphasize on the scientists and statisticians to develop techniques and methodology that supports the analysis of this complex and huge amount of data. Many times, the number of genes in these datasets are much larger than the number of samples. Furthermore, the relevant informative genes associated with the outcome are usually few in the data sets. In addition, the complicated relations among different genes make analysis more difficult. Informative gene selection plays a bigger role in removing redundant and irrelevant genes and improves the quality of result. Various informative gene selection methods exist, and they are being widely used. All these methods aim to remove redundant and irrelevant genes so that classification of new instances is more accurate.

Advances and development of open source as well as commercial computational and statistical tools containing advance techniques for biological data handling and analysis using high performance computing, parallel programming, big data analytics and data visualization enabled the easy way to uncover the biological complexity. Indian Council of Agricultural Research (ICAR) has established a Centre for Agricultural Bioinformatics (CABin) at ICAR-Indian Agricultural Statistics Research Institute, New Delhi in 2010 with a status of a division in the institute to initiate the research in bioinformatics and development of biological data repository. Centre is having researchers from computer application, statistics and biology. Centre is also identified as nodal agency in the country to provide support to biotechnological research in the country through computational biology and agricultural bioinformatics.

In this project, efforts were made to develop a new methodology that builds predictive model from gene expression data and select set of trait specific genes (informative genes) which are highly relevant. This methodology was developed by applying the combination of two conventional machine learning algorithms, support vector machine (SVM) and genetic algorithm (GA).

AUTHORS

Table of Contents

Chapter 1: Introduction

Biological system is being comprehensively profiled by various expression data through high-throughput technologies, such as gene expression data (measured by the microarray or next generation sequencing technology), protein expression (measured by the mass spectrometry-based flow cytometer) and medical imaging (measured by functional magnetic resonance imaging or computerized tomography scan) [1, 2]. Owing to recent technological advances, it is possible nowadays to characterize patients or healthy controls at multiple omics levels. For example, expression of >20000 mRNA transcripts or the methylation status at >400 000 CpG sites in the genome can be measured using microarrays. Next-generation sequencing (NGS) technologies enable even larger numbers of molecules to be quantified. Although different technologies are used for different omics levels, the resulting data sets have several common characteristics making their analysis challenging. The number of variables is often much larger than the number of individuals. Furthermore, the data sets are usually sparse regarding relevant information, i.e. only a small set of variables is associated with the outcome. Additionally, complex correlation patterns are present between the variables. Computational and statistical methods for discovering functional roles of features from expression data are required to have the ability of handling large scale datasets. A straightforward analysis is to carry out statistical tests to identify differentially expressed features between groups of samples [3]. Functional analyses, such as the Gene Set Enrichment Analysis (GSEA) [4], can be followed to discover pathways or biological functions that are over-expressed in the differential feature list. Then the biological semantics of differential features can be explored. Besides differential feature discovery, another important type of analysis is sample classification, in which case samples are classified by characteristics such as disease subtypes and treatment strategies [5]. The classification model constructed from biological expression data can be used for disease diagnosis [6, 7] or clinical outcome prediction [8, 9].

Mutual Information is taken as the basic criterion to find the feature relevance and redundancy. The mutual information between a feature and class labels defines the relevance of that feature. Again, the mutual information among different features defines the correlation i.e., the redundancy among those features. Feature selection is one of the ways to reduce the dimensionality of the data. It is an essential step in successful data mining applications, which can efficiently reduce data dimensionality by removing the irrelevant and redundant features from the original data [10, 11]. At present, there are various kinds of methods to deal with the feature selection problem [12–18]. The feature selection can be supervised or unsupervised. In a supervised scenario [19], the correct class labels of all samples are additionally known and the feature evaluation criterion is based on the known class labels of the samples. In Unsupervised [20–24] case, the feature selection is performed on the basis of some distribution function or clustering in the absence of class label information.

In another context, the feature selection technique can be divided into three categories namely filter, wrapper and embedded. Feature selection methods that make use of a proxy measure to estimate utility are termed as ‘filter’ approaches [16, 17, 24] and feature selection methods that assess feature utility with respect to a given classifier or clustering method, are referred to as ‘wrapper’ [15, 19] approaches. Feature selection methods that select the important features while the model is being trained are termed as embedded methods [30, 31]. Filter-based approaches usually have good generalization properties, but may be less effective at decreasing the dimensionality of the feature space and boosting classification accuracy. Filter-based approaches are computationally cheaper than the wrapper approaches. The real-life data sets frequently contain attributes that are redundant or have a low information content for which the attributes introduce noise and may slow down the classification process gradually. Moreover, they also can introduce high cross-validation errors. Hence selecting the most discriminative attributes [25] may therefore yield significant gains in terms of classification performance. Whatever the way is, the focus of feature selection is to select the features that are most relevant to classification while minimizing the redundancy. But in most of the cases, it has been seen that the basic objective of these methods is either relevance or redundancy.

1.1 Knowledge Gap

Dimensionality reduction transforms high-dimensional data into a meaningful representation of reduced dimensionality [26]. Ideally, the reduced representation should have a dimensionality that corresponds to the intrinsic dimensionality of the data. The intrinsic dimensionality of data is the minimum number of parameters needed to account for the observed properties of the data. In other words, the intrinsic dimensionality is the minimum number of dimensions that represent a manifold on which the original data is embedded. Dimensionality reduction reduces the amount of memory and time required by data mining algorithms and it allows the data to be easily visualized. It may also help to eliminate irrelevant features and noise out the data. Dimensionality reduction methods can be subdivided in two subgroups: feature selection when a subset of the original features set is selected or feature extraction when a new set of features is built based on the old feature set.

Feature selection identifies subsets of data that are relevant to the parameters used and is normally called Maximum Relevance. These subsets often contain material which is relevant but redundant. The fundamental problem with redundancy is that the feature set is not a comprehensive representation of the characteristics of the targeted phenotypes. There are two aspects of this problem i.e., Efficiency and Broadness. In efficiency, if a feature set of 50 genes contains quite a number of mutually highly correlated genes, the true "independent" or "representative" genes are therefore much fewer, say 20, We can delete the 30 highly correlated genes without effectively reducing the performance of the prediction; this implies that 30 genes in the set are essentially "wasted". In Broadness, the features are selected according to their

discriminative powers and they are not maximally representative of the original space covered by the entire dataset.

As the dimensionality of the data rises, the amount of data required to provide a reliable analysis grows exponentially. Bellman referred to this phenomenon as the "curse of dimensionality" when considering problems in dynamic optimisation [27]. A popular approach to this problem of high-dimensional datasets is to search for a projection of the data onto a smaller number of variables (or features) which preserves the information as much as possible. Microarray and NGS data are typical of this type of small sample problems. Each data point (sample) can have large number of variables and processing a large number of data points involves high computational cost. When the dimensionality of a dataset grows significantly there is an increasing difficulty in proving the result statistically significant due to the sparsity of the meaningful data in the dataset in question. Large datasets with the so-called "large p, small n" problem (where p is the number of features and n is the number of samples) tend to be prone to over fitting. An over fitted model can mistake small fluctuations for important variance in the data which can lead to classification errors.

NGS and microarrays are a popular source of data for gathering gene expressions. Analysing these can be difficult due to the size of the data. In addition, the complicated relations among the different genes make analysis more difficult and removing excess features can improve the quality of the results. Feature selection plays a bigger role in removing irrelevant features. Many different feature selection and feature extraction methods exist and they are being widely used. All these methods aim to remove redundant and irrelevant features so that classification of new instances will be more accurate.

Many feature selection algorithms (FSA) are introduced in past decade but most of them do not perform well on high-dimensional datasets with a large number of redundant features. These algorithms focus only on the necessary features pertaining to build the efficient and accurate model [28]. There are three different types of feature selection methods named Filter, Embedded and Wrapper method. Apart from this, feature selection can be Univariate or Multivariate. When a Univariate method does not take into account the dependency among the features, a Multivariate method does it [29]. The drawbacks of wrapper and filter method are that the former suffer from high computational cost while the later does not interact with classifiers. Embedded methods can be a solution to this problem that uses classifiers to rank features. SVM was trained with the current set of features and the least performing feature indicated by SVM was removed using a new embedded method of SVM on Recursive Feature Elimination (SVM-RFE) [30]. Further, a new method called kernel-penalized SVM has also been proposed [31].

A particularly well-suited method to tackle the presented challenge is random forest (RF) [32], an ensemble learning method based on decision trees. RF provide variable

importance measures, which can be used to rank variables based on their predictive importance. However, it is difficult to distinguish relevant from irrelevant variables based on their ranking only. Therefore, several variable selection procedures have been proposed that used different criteria and approaches to report the set of truly relevant variables. One popular approach that is also used in combination with other machine learning methods is recursive feature elimination (RFE) [33]. RFE uses the prediction error to select a minimal set of variables needed for a good prediction. Hence, only a limited number of variables need to be measured for further application of the prediction model. A popular alternative to the RF approach for variable selection is penalized regression methods (also called regularized or shrinkage regression methods) such as Least Absolute Shrinkage and Selection Operator [34] or elastic net [35], which have been applied to omics data sets [36, 37]. The general idea is to add a penalty to the loss function so that regression coefficients are shrunken toward zero resulting in a sparse model. The performance of different types of penalized regression methods has been evaluated in several studies, e.g. [38, 39]; however, to the best of our knowledge, no comprehensive and neutral study comparing RF and penalized regression methods has been performed regarding selection of all relevant variables, In a study [40], combined parametric (t test based p value) and non-parametric (fold change value) method with more predictive power has been developed for microarray data. In a similar way, we will apply different techniques and modify the existing methods. Once the dataset is reduced, it will be combined with a classifier to check for accuracy and the best combination will be determined. The ability of an FSA will be measured by testing the accuracy of the classifier trained by using the reduced subset. Different well-known classifiers, such as Random Forest (RF), Decision Tree (J48), k-nearest neighbours (k-NN), Naive Bayes (NB) and Support Vector Machine (SVM) etc. will be used for validating the output of FSAs. Reduced feature subset will be used to train the classifiers and thereby measure its classification ability.

Selection of informative genes from high dimensional gene expression data has emerged as an important research area in transcriptomic. One of the major issues with the RNA-Seq approach in whole genome transcriptome analysis is that, the expression dynamics of various different genes are captured. This result in very high dimensionality in the data, which means the number of genes, is much larger than the number of samples. Therefore, it is important to select most relevant genes related to condition class from thousands of genes with the help of appropriate computational approaches. Most of the existing gene selection methods either fail to identify a list of predictive genes or ignores the spurious relations between genes and trait under study. In this project a new methodology had been developed that builds predictive model from gene expression data and select set of trait specific genes (informative genes) which are highly relevant. This methodology was developed by applying the combination of two conventional machine learning algorithms, support vector machine (SVM) and a genetic algorithm (GA). They are integrated effectively based

on a wrapper approach. GA was used to control and optimize the subset of genes sent to the SVM for classification and evaluation. Using SVM as the classifier performance and the Genetic algorithm for feature selection, a set of informative genes set was obtained. The classification accuracy of the obtained genes set from the developed methodology was compared with the genes set obtained from methods such as Boot-MRMR, MRMR, t-score and F-score of R- package "GSAQ" [81].

1.2 Objectives

To develop the methodology for trait specific genes identification based on gene expression data
To evaluate the developed methodology with the existing methods
To develop R package/web server of developed methodology

This project report is organized into five chapters, as follows
Chapter 1 gives the introduction, problem definition including knowledge gap and it also specifies the objectives of the project.
Chapter 2 provides the review of literature in the area and specify the scope of work.
Chapter 3 in this chapter, the methodology and tools used to develop the algorithm for informative gene selection from gene expression data has been discussed.
Chapter 4 provides the result of data analysis and comparative evaluation of the developed methodology, It also shows the sample report obtained using the developed web tool TSGS.
Finally, the report is concluded in chapter 5 followed by references.

Chapter 2: Review of Literature

Features can be selected in many ways. One scheme is to select features that correlate strongest to the classification variable. This has been called maximum-relevance selection [65, 66]. Many heuristic algorithms can be used, such as the sequential forward, backward, or floating selections. On the other hand, features can also be selected to be mutually far away from each other while still having "high" correlation to the classification variable. This scheme, termed as Minimum Redundancy Maximum Relevance (mRMR) selection has been found to be more powerful than the maximum relevance selection. As a special case, the "correlation" can be replaced by the statistical dependency between variables. Mutual information can be used to quantify the dependency. Minimum redundancy feature selection is an algorithm frequently used in a method to accurately identify characteristics of genes and phenotypes and narrow down their relevance and is usually described in its pairing with relevant feature selection as Minimum Redundancy Maximum Relevance [67, 68]. There are some examples of embedded feature selection methods which achieve the feature selection by imposing regularisation on existing classification methods, such as regularised SVM [41] and sparse logistic regression [42, 43]. The work in [44] develops a Bayesian approach based on a probit regression model with a generalised singular g-prior distribution for regression coefficients.

Traditional feature selection methods include statistics tests to reduce feature space by examining whether the significant values of features of a test pass the predefined threshold. For biological data, there are many advanced feature selection methods being proposed. For example, the binary particle swarm optimisation (BPSO) based model is proposed in [45] for the gene selection of Microarray data. To improve the performance of feature selection, BPSO uses gene-to-class sensitivity (GCS) information in the feature selection process. GCS information is obtained from gene expression data indicating whether a gene is sensitive to sample classes. To evaluate candidate gene subsets selected from BPSO, extreme learning machine (ELM) is used for classification model construction.

There are a large range of machine learning methods to construct classification models. Examples of such methods include deep learning [46, 47], graphical models [48, 49], nonparametric Bayesian models [50, 51], linear discriminant analysis [52], and Naive Bayes [53]. Many tools are particularly designed for biological data. For example, a Python package called Pse-Analysis [54], is developed to automatically generate classifiers for genomics and proteomics datasets. It is based on the framework of LIBSVM [55] and inherits the characteristics of the SVM method. Another classification method, Sparse Bayesian Learning (SBL) [56, 57, 58] is featured in overcoming the dimensionality problem. SBL only uses a small subset of input features for prediction, based on the observation that relevant features are sparse compared to the dimension of whole feature space. Bayesian inference is adopted to obtain solutions for probabilistic classification. SBL is in the same functional form of SVM, but provides probabilistic

classification. SBL uses a fully probabilistic framework and introduces a prior over the model weights governed by a set of hyper parameters. The feature set returned from these methods cannot be easily used to discover predictive pathways or biological functions. Random Forest has been successfully applied in genetic [59], gene expression [60, 61], methylation [62], proteomics [63] and metabolomics studies [64]. It is a flexible approach that can be used to both perform classification, i.e. predicting case-control status, and regression, i.e. predicting quantitative traits.

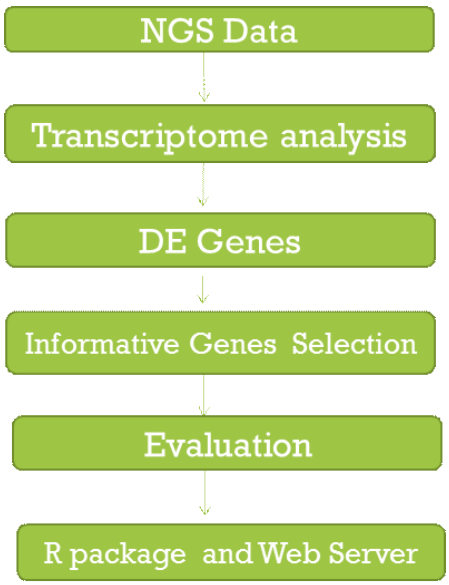
Wrapper method tend to perform better in selecting features since they take the model hypothesis into account by training and testing in the feature space. This leads to the big disadvantage of wrappers, the computational inefficiency which is more apparent as the feature space grows. Unlike filters, they can detect feature dependencies. They are generally categorised into two types randomised and deterministic. Randomized Wrappers use genetic algorithms (GA). Best Incremental Ranked Subset (BIRS) [70] is an algorithm that scores genes based on their value and class label and then uses incremental ranked usefulness based on the Markov blanket to identify redundant genes. Linear discriminant analysis was used in combination with genetic algorithms. Subsets of genes are used as chromosomes and the best 10% of each generation is merged with the previous ones. Part of the chromosome is the discriminant coefficient which indicates the importance of a gene for a class label [71]. Genetic Algorithm-Support Vector Machine (GA-SVM) [72] creates a population of chromosomes as binary strings that represent the subset of features that are evaluated using SVMs.

Next chapter describes the materials used and methodology adopted for obtaining trait specific genes based on feature selection data.

Chapter 3: Materials and Methods

In this study an improvised gene selection approach has been proposed, i.e. SVM-GA wrapper method for selection of informative genes from high dimensional genome expression data. The proposed approach can select informative gene through an optimization procedure that is modelled on the principles of evolution via natural selection, employing a population of individuals that undergo selection in the presence of operators such as mutation and crossover. A genetic algorithm works with a population of individual strings called chromosomes, each representing a possible solution to a given problem. In this case we are using binary chromosomes. In genetic algorithm a fitness value is used to evaluate individual chromosome present in the population. Those chromosomes with the highest fitness values are given more opportunities to reproduce and the offspring share features taken from their parents. This ensures that the selected genes are carried to the next generation. In genetic algorithm a fitness function is used to assign the fitness values to each chromosome. Here we used SVM to define a fitness function.

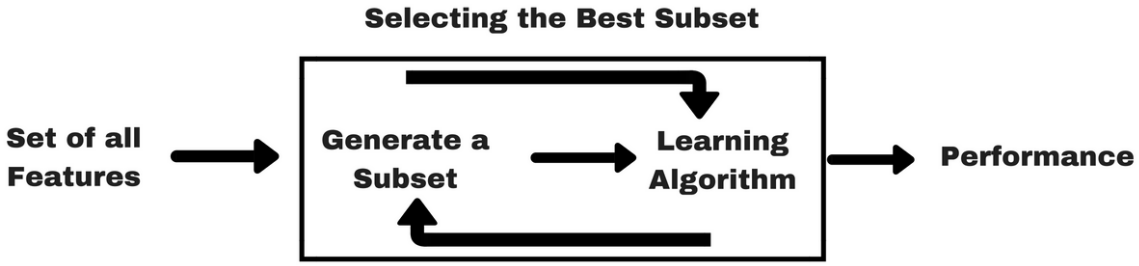
The process flow representation of the developed methodology is shown in following diagram.



The following steps were used to carry out the proposed work

- Select and extract trait specific suitable NGS Data from NCBI
- Simulation of gene expression NGS Data
- Pre-processing of the real data
- Obtaining differentially expressed genes
- Relevant Gene's selection using wrapper methods

Wrapper Methods: Wrapper methods consider the selection of a set of features as a search problem, where different combinations are prepared, evaluated and compared to other combinations. A predictive model is used to evaluate a combination of features and assign a score based on model accuracy. An example of a wrapper method is the recursive feature elimination algorithm.



All the steps to develop the methodology were implemented in R. The major steps of the developed methodology to obtain optimal number of informative genes are as follows:

3.1 Filtering to remove lowly expressed genes

Genes with very low read counts were filtered out across all the samples prior to further steps as these genes provide little evidence and impose problems for differential expression analysis [74]. Further, the removal of genes will reduce the dimension of data, thereby making the program more efficient in terms of time. We selected the genes with at least 10 counts in 2 samples. Then, the genes having the counts per million (cpm) values above a threshold value were retained for next step.

3.2 Data normalization

Data normalization is required to account for the within library and between library variability. For further downstream analysis, normalization by trimmed mean of M-values (TMM) has been used to obtain the effective library sizes [76]. The "calcNormFactors" function normalizes the library sizes by finding a set of scaling factors for the library sizes that minimizes the log fold changes between the samples for most genes. These scale factors use TMM values between each pair of samples.

3.3 Identification of significantly differentially expressed genes

The biological coefficient of variation (BCV) is estimated by using a negative binomial model [77]. We fit a quasi-likelihood negative binomial generalized log-linear model to the count data. Then, we perform statistical test for each gene at a desired level of significance (e.g., $\alpha = 0.05$). Further, we adjust the p-values for multiple testing of genes. We have provided various options of adjusting p-values

such as "bonferroni" (default option), "BH", "holm", "hochberg", "hommel" and "BY". The significantly differentially expressed genes are identified and the count data corresponding to these genes are used for further steps. R Package “edgeR” [75] was used for above steps.

3.4 Obtaining optimal set of informative genes using SVM as the classifier performance and genetic algorithm for gene selection

- Primary screening of the genes was done by identifying the differentially expressed genes. The proposed algorithm consists of a Support Vector Machine (SVM) and a Genetic Algorithm (GA). GA was used to control and optimize the subset of genes sent to the SVM for classification and evaluation.
- Initially, the dataset was split randomly into test and training data.
- To initialize GA, first generation of individuals were created by picking random subsets of genes and training the SVM on those genes.
- The fitness of an individual was then determined by the performance of the SVM on the test data
- Once an initial population of individuals is generated, the GA procedure was then used to evolve a new generation of individuals and the process was repeated for several generations. In this way GA progressively iterates onto a near optimal set of genes.
- Best Individual of each generation was selected and frequency of occurrence of each genes was computed across all the generation.
- To obtain the final subset of genes, the genes that are selected the most often were collected to form the optimal set of genes.
- R package/web server was developed to obtain trait specific genes from gene expression data through developed methodology.

3. 4.1 Genetic Algorithm

Each possible subset of genes corresponds to an “individual” or “chromosome” in the GA algorithm and is represented by a string of bits of length N ; suppose there are 50000 genes , so N = 50000 in each case. Each chromosome is a bit-string of length 50000 with binary representation 1 or 0; 1 means that particular gene is included in the subset of genes to be supplied to the SVM while 0 means that it is excluded.

The initial set of individuals is generated randomly with the restriction that each gene is required to be represented at least once in the initial pool. The pool of individuals which constitutes the first generation is then evolved using the GA which consists of a number of operations.

Each individual is used to train a SVM. Then a fitness score is assigned to each individual based on how well the corresponding SVM classifier classifies the test dataset. The fitness function, f, is as follows;

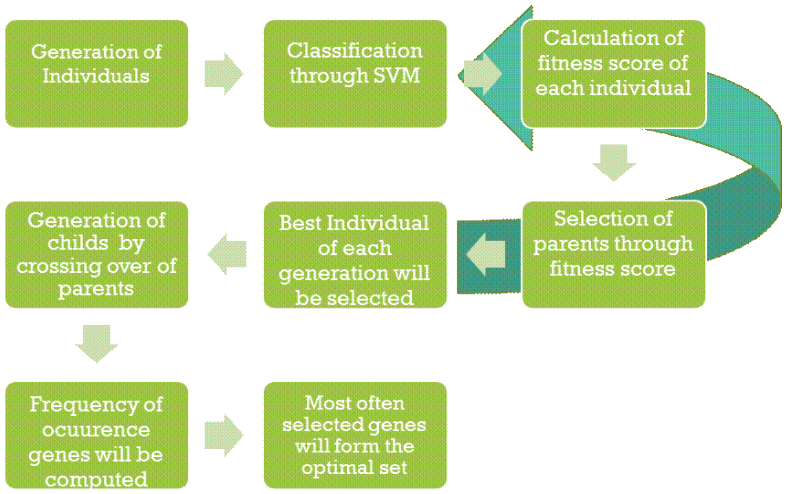
$f = 1/(X–X_0) + 1/ X_1$

where X_0 is the number of samples known to be classified correctly as 0 (control class) and X_1 is the number of samples known to be classified correctly as 1(treatment class) in the test dataset.

Once the fitness has been calculated for all individuals, parent individuals are selected to undergo crossover with a probability proportional to their fitness.

To produce the next generation of individuals, crossover will be performed on two parent individuals. One-point crossover will be used with the crossover point selected at random. To generate a child, data from the two parent-individuals will be swapped, suitable number of individuals will be produced for the next generation. Elitism will be employed which means the best solution from each generation will be copied unchanged to the next generation.

PROCESS Flow of Informative Gene Selection consists of eight step procedure.



3.5 Performance Evaluation of gene selection techniques based on classification

The performance of the proposed and existing gene selection techniques was evaluated based on subject classification accuracy, the number of top ranked genes selected through the proposed and other existing techniques were then used in SVM classifier to discriminate the class labels of samples between samples (stress; +1/ control; -1) on different datasets (real and simulated data).

An SVM learn to discriminate between the members and non-members of a given functional class based on expression data. Having learned the expression features of the class, the SVM could recognize new genes as members or non-members of the class based on their expression data. Leave-One-Out Cross-Validation (LOOCV) method was used to assess the classifying ability of the developed system. The LOOCV procedure works as by dividing all samples into K subsets randomly, where K is the total number of samples. Then K - 1 subsets are used to train the model and the remaining Kth sample is used for testing and the same is repeated for K times such that each sample is given a chance for testing the performance.

In the SVM classifier, three basic kernel functions, i.e. linear (SVM-LBF), radial (SVM-RBF) and polynomial (SVM-PBF) were used to compute the classification accuracy. Further, the techniques which provide maximum discrimination between the two groups through classification will be the better technique for informative gene selection and vice-versa. The performance of these techniques was adjudged on the basis of classification accuracy.

The other criteria viz. sensitivity, specificity, False Discovery Rate (FDR), False Positive Rate (FPR), False Negative Rate (FNR), Accuracy (ACC), and F1-Score were also used in this performance evaluation.

3.6 R programming and R packages

R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used for statistics and data mining for developing statistical software. R and its libraries implement a wide variety of statistical and graphical techniques, including linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, and others. R is easily extensible through functions and extensions, and the R community is noted for its active contributions in terms of packages. The packages used for this study are as follows:

CARET

The caret package short for Classification And REgression Training (CARET) contains functions to streamline the model training process for complex regression and classification problems. The package utilizes a number of R packages but tries not to load them all at package start-up by removing formal package dependencies, the package startu33p time can be greatly decreased. The package “suggests” field includes 30 packages. caret loads packages as needed and assumes that they are installed.

Caret has several functions that attempt to streamline the model building and evaluation process, as well as feature selection and other techniques. One of the primary tools in the package is the "train" function which is used to• evaluate, using resampling, the effect of model tuning parameters on performance• choose the optimal model across these parameters• estimate model performance from a training set

The functions used from this package are "createDataPartition" which is used to divide the data into training and testing subsets. The model was trained on the training dataset using the "train" function. The "predict" function was used to predict the classes of the testing datasets. Calculation a cross-tabulation of observed and predicted classes with associated statistics was done by “confusionMatrix" function from where we obtain the accuracy of the model.

edgeR

This R package is available for Differential expression analysis of RNA-seq expression profiles with biological replication. Implements a range of statistical methodology based on the negative binomial distributions, including empirical Bayes estimation, exact tests, generalized linear models and quasi-likelihood tests. This package was applied in our work for primary screening of genes, normalization of gene expression data and obtaining differentially expressed genes which were used with SVM and GA to obtain the informative genes.

Genalg

The package has R based genetic algorithm for binary and floating point chromosomes. In this package we use "rbga.bin" which is a R based genetic algorithm that optimizes, using a user set evaluation function and a binary chromosome which can be used for variable selection. The optimum is the chromosome for which the evaluation value is minimal. It requires a "evalFunc" method to be supplied that takes as argument the binary chromosome, a vector of zeros and ones. Additionally, the GA optimization can be monitored by setting a "monitorFunc" that takes a rbga object as argument. Results can be visualized with "plot.rbga" .

SimSeq

SimSeq performs data based simulation **of RNA-Seq** data creating a dataset with a known list of DE and EE genes. The core function that implements of the methodology of SimSeq is the SimData function. The vector of read counts simulated for a given experimental unit has a joint distribution that closely matches the distribution of a source RNA-seq dataset provided by the user. Users control the proportion of genes simulated to be differentially expressed (DE) and can provide a vector of weights to control the distribution of effect sizes. The algorithm requires a matrix of RNA-seq read counts with large sample sizes in at least two treatment groups.

CompcodeR

It is an R package that provides extensive functionality for comparing results obtained by different methods for differential expression analysis of (mainly) RNAseq data. It also contains functions for simulating count data and interfaces to several packages for performing the differential expression analysis.

3.7 Data collection

For real balanced dataset, we used KIRC RNA-seq dataset (The version of the KIRC dataset unc.edu_KIRC.IlluminaHiSeq_RNASeqV2.Level_3.1.5.0 accessed from Simseq package of R) containing 20,531 genes and 72 paired columns of data with rows corresponding to genes and columns corresponding to replicates; replic vector specifies replicates and treatment vector specifies non-tumour and tumour group samples respectively within replicate (The Cancer Genome Atlas Research Network, 2013). The

GE experimental datasets [73] of UV stress on Arabidopsis thaliana were collected from Gene Expression Omnibus database of NCBI (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GPL19290). GEO series file GSE64870, the response of Arabidopsis thaliana accessions to UV radiation stress was obtained and analysed. The data was for the platform GPL17639.

For real unbalanced dataset, we used TCGA re-processed RNA-Seq data from 9264 tumor samples and 741 normal samples across 24 cancer types available via GSE62944 from GEO. We have used an R package "ExperimentHub" (82) to obtain the count data of TCGA's Low Grade Glioma (LGG) samples. There are total 97 samples corresponding to two groups: (i) IDH1 mutant (70 samples) and (ii) IDH1 wild (27 samples). The total number of genes is 23368. The data is unbalanced as it has unequal number of samples in each group

Further R codes were also written for generating simulated dataset and testing model accuracy and classification. We used synthetic and real RNA-Seq datasets. The synthetic dataset following parametric distribution was generated using compcodeR() package (Soneson, 2014). The simulation was performed following the description by Soneson and Delorenzi (2013). The count dataset contained 15,000 genes for two groups of 15 samples each, where 10% of the genes are simulated to be deferentially expressed between the two groups (equally distributed between up- and down regulated in group 2 compared to group 1). Furthermore, the counts for all genes were simulated from a Negative Binomial distribution with the same dispersion in the two sample groups. For simulating dataset following non parametric distribution, we used package SimSeq (Benidt and Nettleton, 2015), the generated count dataset contained 15,000 genes for two groups of 35 samples each, where 10% of the genes were simulated to be deferentially expressed between the two groups.

Sl. No.	Description	Source	Genes	Samples	Class
1	KIRC RNA-seq dataset	unc.edu_KIRC.IlluminaHiSeq_RNASeqV2.Level_3.1.5.0	20531	144	2
2	UV stress on Arabidopsis thaliana	GSE64870	24185	22	2
3	TCGA (LGG) RNA-Seq data	GSE62944	23368	97	2
4	Simulated Data 1	SimSeq package of R	15000	70	2
5	Simulated Data 2	compcodeR package	15000	30	2

Table 3.1: Gene expression data used

3.8 R Codes

a. Code for the GA-SVM feature selection

```
 featureSelect <- function(X, y, p = 20, n.iter = 5, alpha = 0.05, p.adj.method =
"bonferroni"){

 countData1 <- X

 geneNames1 <- rownames(X)

 Labels <- as.numeric(y)

 group <- unique(y)

 z <- factor(Labels, levels = group, labels = group)

 n1 <- length(which(z == group[1]))

 n2 <- length(which(z == group[2]))

 n <- n1+n2

 ## Filtering to remove low count reads: [74].

 LS <- colSums(countData1)

 LS.CPM <- LS/10^6

 t <- round(10/min(LS.CPM), 1) # Threshold

 y <- DGEList(counts = countData1, genes = geneNames1)

 keep <- rowSums(cpm(y) > t) >=2

 y <- y[keep, , keep.lib.sizes=FALSE]

 countData <- y$counts

 geneNames <- y$genes

 nGenes <- nrow(countData)

 y <- calcNormFactors(y)

 design <- model.matrix(~z)

 y <- estimateDisp(y, design = design)
```

```
fit <- glmQLFit(y, design)
qlf <- glmQLFTest(fit, coef=2)
res2 <- topTags(qlf, n=nGenes)
res.tab <- res2$table
ind1 <- which(res.tab$PValue < alpha)
adj.pval <- p.adjust(res.tab$PValue, method = p.adj.method)
res.tab$`Adjusted PValue` <- adj.pval
ind <- which(adj.pval < alpha)
geneNames.sel <- res.tab$genes[ind]

res.f <- res.tab[ind,]
log.counts <- cpm(y$counts, log = TRUE)
countData.f <- log.counts[ind,] # Final log cpm data for feature selection
data <- t(countData.f)
s <- data.frame(data)
t <- z
eval_funct <- function(indices){
  evl_df <- cbind(s[,indices==1],t)
  evl_trng <- createDataPartition(evl_df$t, p=0.60,list = FALSE)
  evl_test <- evl_df[-evl_trng,]
  evl_train <- evl_df[evl_trng,]
  evl_svm <- train(t~.,data=evl_train,method="svmRadial",preProc=c("zv"),
                   trControl=trainControl(method = "cv",number = 5),savePredictions = "all")
  evl_cls <- predict(evl_svm,newdata=evl_test)
```

```
  evl_tbl <- confusionMatrix(evl_cls,evl_test$t)
  tbl <- evl_tbl$table
  tp <- tbl[1,1]
  tn <- tbl[2,2]
  t <- tbl[1,1]+tbl[1,2]+tbl[2,1]+tbl[2,2]
  result <- -(tn+t-tp)/((tn*t)-(tp*tn))
  return(result)
}
monitor <- function(obj) {
  minEval = min(obj$evaluations);
  filter = obj$evaluations == minEval;
  bestObjectCount = sum(rep(1, obj$popSize)[filter]);
  # ok, deal with the situation that more than one object is best
  if (bestObjectCount > 1) {
    bestSolution = obj$population[filter,][1,];
  } else {
    bestSolution = obj$population[filter,];
  }
  outputBest = paste(obj$iter, " #selected=", sum(bestSolution),
                     " Best (Error=", minEval, "): ", sep="");
  for (var in 1:length(bestSolution)) {
    outputBest = paste(outputBest,
                       bestSolution[var], " ",
                       sep="");
```

```
  }
  outputBest = paste(outputBest, "\n", sep="");
  cat(outputBest);
 }


 woppa <- rbga.bin(size=ncol(s),popSize=p,iters=n.iter, mutationChance=0.30,
zeroToOneRatio=20,
                evalFunc=eval_funct, verbose=TRUE, monitorFunc=monitor)
 bestSolution <- woppa$population[which.min(woppa$evaluations),]
 result <- cbind(data[,bestSolution==1],z)
 feature.selected <- res.f[bestSolution ==1,1]
 logcpm.feature.selected <- t(result)
 ind.m <- fmatch(as.character(feature.selected), as.character(res.f[,1]))
 result.pval <- res.f[ind.m, ]
 list(`InformativeGenes` = feature.selected,
     `LogCPM` = logcpm.feature.selected,
     `DEA_Result` = result.pval)
}
```

b. R function for evaluation of classification method:

We used “svmRadial” method and “LOOCV” resampling method for building the training model and validation using “caret” R package.

```
eval_funct<- function(data){
 s<-data[,-ncol(data)]
 t<-data[,ncol(data)]
  evl_df<-cbind(s,t)
  evl_svm<-train(t~.,data=evl_df,method="svmRadial",
```

```
            trControl=trainControl(method = "LOOCV" ),savePredictions = "all")
  evl_cls<-predict(evl_svm,newdata=evl_df)
  evl_tbl<-confusionMatrix(as.factor(evl_cls), as.factor(evl_df$t))
  return(evl_tbl)
 }
```

c. Code for generating simulated data following parametric distribution

```
library(compcodeR)
simdata <- generateSyntheticData(dataset = "B_625_625", n.vars = 15000,
samples.per.cond = 15, n.diffexp = 1500,
relmeans = "auto", dispersions = "auto",
repl.id = 1, seqdepth = 1e7,
fraction.upregulated = 0.5,
between.group.diffdisp = TRUE,
filter.threshold.total = 1,
filter.threshold.mediancpm = 0,
fraction.non.overdispersed = 0,
random.outlier.high.prob = 0, random.outlier.low.prob =0,
single.outlier.high.prob = 0, single.outlier.low.prob = 0,
output.file = "B_625_625_5spc_repl1.rds")
write.csv(simdata@count.matrix, file=”filepath”)
```

d. Code for generating simulated data following non parametric distribution

```
library(SimSeq)
data(kidney)
```

```
counts <- kidney$counts # Matrix of read counts from KIRC dataset

replic <- kidney$replic # Replic vector indicating paired columns

treatment <- kidney$treatment # Treatment vector indicating Non-Tumor or Tumor columns

nf <- apply(counts, 2, quantile, 0.75)

library(fdrtool)

sort.list <- SortData(counts = counts, treatment = treatment, replic = replic,

sort.method = "paired", norm.factors = nf)

counts <- sort.list$counts

replic <- sort.list$replic

treatment <- sort.list$treatment

nf <- sort.list$norm.factors

probs <- CalcPvalWilcox(counts, treatment, sort.method = "paired",

sorted = TRUE, norm.factors = nf, exact = FALSE)

weights <- 1 - fdrtool(probs, statistic = "pvalue", plot = FALSE, verbose = FALSE)$lfdr

data.sim <- SimData(counts = counts, replic = replic, treatment =treatment,

sort.method ="paired", k.ind = 35, n.genes = 15000, n.diff = 1500,

weights = weights, norm.factors = nf)

write.csv(data.sim$counts, file=”path”)
```

e. R function to filter and transform RNA-Seq count data:

The function is used to filter lowly expressed genes and transform the RNA-Seq count data to log counts per million (log cpm) values. The resulting log cpm values have also been used for feature selection using different methods given in “GSAQ” R package.

```
datalogcpm <- function(X, y){

  countData1 <- X

  geneNames1 <- rownames(X)

  Labels <- as.numeric(y)
```

```
 group <- unique(y)
 z <- factor(Labels, levels = group, labels = group)
 n1 <- length(which(z == group[1]))
 n2 <- length(which(z == group[2]))
 n <- n1+n2
  ## Filtering to remove low count reads: [74]
 LS <- colSums(countData1)
 LS.CPM <- LS/10^6
 t <- round(10/min(LS.CPM), 1) # Threshold
 y <- DGEList(counts = countData1, genes = geneNames1)
 keep <- rowSums(cpm(y) > t) >=2
 y <- y[keep, , keep.lib.sizes=FALSE]
 countData <- y$counts
 geneNames <- y$genes
 nGenes <- nrow(countData)
 log.cpm <- cpm(countData, log = TRUE)
 return(as.data.frame(log.cpm))
}
```

Chapter 4: RESULTS & DISCUSSION

This section describes the experimental results obtained by applying the developed algorithms to the data sets. For experimentation, two read count data sets were obtained. The present study divided the data into 10 folds where 1 fold was for testing and 9 folds were for training for the 10-fold crossover validation for training the SVM classifier for the genetic algorithm. A population size of 100 was created for 100 generations in the genetic algorithm. The default mutation probability of 0.30 and an elitism probability of 0.10 was applied in the algorithm. An informative gene set of size 350 was selected from GSE64870 gene expression dataset and gene set of size 551 was selected from Kidney data using developed algorithm.

4.1 Analysis Results

a. Real Balanced Data

1. UV stress on Arabidopsis thaliana data with population size=100 and Iterations =100 and 22 variables

	Reference	
Prediction	Control	Treatment
Control	11	0
Treatment	0	11

Table 4.1: Confusion Matrix for UV stress on Arabidopsis thaliana data

Method	**Accuracy**	**Sensitivity**	**Specificity**	**PPV**	**NPV**	**Precision**
TSGS	1	1	1	1	1	1

Table 4.2: Statistics for UV stress on Arabidopsis thaliana data

95% CI : (0.8316, 1)

2. KIRC RNA-seq data with population size=100 and Iterations =100 and 144 variables

	Reference	
Prediction	Control	Treatment
Control	71	1
Treatment	1	71

Table 4.3: Confusion Matrix for KIRC RNA-seq data

Method	Accuracy	Sensitivity	Specificity	PPV	NPV	Precision
TSGS	98.61	98.61	98.61	98.61	98.61	98.61

Table 4.4: Statistics for KIRC RNA-seq data

95% CI : (0.9507, 0.9983)

b. Real Unbalance data

TCGA **(LGG)**RNA-Seq data with population size=100 and Iterations =100 and 97 variables

	Reference	
Prediction	Control	Treatment
Control	20	0
Treatment	7	70

Table 4.5: Confusion Matrix for TCGA (LGG) RNA-seq data

Method	Accuracy	Sensitivity	Specificity	PPV	NPV	Precision
TSGS	0.9278	0.7407	1.00	1.00	0.9091	1.00

Table 4.6: Statistics for TCGA RNA-seq data

95% CI: (0.857, 0.9705)

c. Simulated data 1 with population size=100 and Iterations =100 and 35 variables

	Reference	
Prediction	Control	Treatment
Control	35	1
Treatment	0	34

Table 4.7: Confusion Matrix for simulated data 1

Method	Accuracy	Sensitivity	Specificity	PPV	NPV	Precision
TSGS	98.57	1.00	97.14	97,22	1.00	98.57

Table 4.8: Statistics for simulated data 1

95% CI : (0.923, 0.9996)

d. Simulated data 2 with population size=100 and Iterations =100 and 15 variables

	Reference	
Prediction	Control	Treatment
Control	15	0
Treatment	0	15

Table 4.9: Confusion Matrix for simulated data 2

Method	Accuracy	Sensitivity	Specificity	PPV	NPV	Precision
TSGS	1	1	1	1	1	1

Table 4.10: Statistics for simulated data 2

95% CI: (0.8843, 1)

4.2 Performance Analysis

We got 551 genes selected using developed method TSGS. Here, we have used "DESeq2" for the differential expression analysis. For comparison with other methods, we sorted the genes according to "BH" adjusted p-value with 0.05 level of significance and then selected top 100 genes for determining the accuracy of our developed methodology.

Our developed method TSGS was compared with "t-score", "F-score", "MRMR" and "bootMRMR" methods available in "GSAQ" R package. Informative gene selection using each of these methods was performed. Since, "MRMR" and "boot-MRMR" methods are for smaller dataset they were unable to select 100 genes using the complete dataset. Therefore, for these two methods, we performed differential expression analysis and selected top 1000 genes. We apply the filtering criteria: "BH" adjusted p-value < 0.05 and log fold change cut off of ±1 (logFC > 1 and logFC < -1). We used data corresponding to 100 genes selected using these four methods We used "svmRadial" method and "LOOCV" resampling method for building the training model and validation using "caret" R package. We performed the validation 100 times. The mean of various measures of accuracies (in %) obtained using each method are shown in Table 4.2.1:

	Accuracy	Sensitivity	Specificity	PPV	NPV	Precision	Recall	F1
TSGS	98.61	98.61	98.61	98.61	98.61	98.61	98.61	98.61
t-score	97.30	95.99	98.61	98.57	96.13	98.57	95.99	97.25

F-score	97.81	97.00	98.61	98.59	97.06	98.59	97.00	97.78
MRMR	98.06	97.5	98.61	98.60	97.53	98.60	97.50	98.04
bootMRMR	98.15	97.69	98.61	98.60	97.72	98.60	97.69	98.14

Table 4.2.1: Evaluation of TSGS with other methods

The performance analysis showed that the developed methodology TSGS selects informative genes which are more biologically relevant. The developed methodology TSGS is also found to be quite competitive with the existing techniques with respect to subject classification accuracy. Our results also showed that under the multiple criteria decision-making setup, the proposed technique is better for informative gene selection over the above compared methods.

Initially, we have used "DESeq2" for the differential expression analysis in our method. Later on, based on various suggestions, we used "edgeR" package for differential expression analysis. Furthermore, we used a portion of our program to get the expression data and used this data as input to the different methods of "GSAQ" package. After modifying our program, 622 genes got selected using our method. For comparison with other methods, we selected the genes according to "BH" adjusted p-value with 0.05 level of significance. Our proposed method was compared with "t-score", "F-score", "MRMR" and "bootMRMR" methods available in "GSAQ" package. We performed gene selection using each of these methods. We used the expression data obtained from our method as input data for these methods. However, "MRMR" and "boot-MRMR" methods were unable to select any genes using the complete dataset. This is one of the limitations of "GSAQ" R package. Therefore, we discarded these two methods from further comparison. We selected 622 genes using the methods "t-score" and "F-score". We used "svmRadial" method and "LOOCV" resampling method for building the training model and validation using "caret" R package. We performed the validation 100 times. The mean of various measures of accuracies (in %) obtained using each method are shown in Table 4.2.2:

	Accuracy	Sensitivity	Specificity	PPV	NPV	Precision	Recall	F1
TSGS	97.65	98.61	96.69	96.8	98.58	97.69	97.65	98.61
t-score*	97.73	98.61	96.85	96.94	98.59	97.76	97.73	98.61
F-score*	98.01	98.61	97.42	97.46	98.59	98.03	98.01	98.61
MRMR*	NA	NA	NA	NA	NA	NA	NA	NA
bootMRMR*	NA	NA	NA	NA	NA	NA	NA	NA

Table 4.2.2: Evaluation of TSGS using KIRC RNA-seq data , TSGS was used to get the expression data which was then used as input to the different methods of "GSAQ" package.

From the above table, we observe that "F-score" method has more accuracy as compared to other methods. However, the results are comparable. Thus, we can say that after including a portion of our method TSGS, the accuracy of other methods also increases. Same conclusion was drawn when we used simulated data (Benidt and Nettleton, 2015) consisting of 15000 genes and 35 samples in each of the two classes (Please see section 3.7 & 3.8 d). We got 67 genes selected using developed method TSGS. The mean of various measures of accuracies (in %) obtained using each method are shown in Table 4.2.3:

	Accuracy	Sensitivity	Specificity	PPV	NPV	Precision	Recall	F1
TSGS	98.5	99.89	97.11	97.2	99.89	98.52	98.5	99.89
t-score*	98.41	99.89	96.94	97.04	99.89	98.44	98.41	99.89
F-score*	98.57	100	97.14	97.22	100	98.59	98.57	100
MRMR*	NA	NA	NA	NA	NA	NA	NA	NA
bootMRMR*	NA	NA	NA	NA	NA	NA	NA	NA

Table 4.2.3: Evaluation of TSGS using simulated data1, TSGS was used to get the expression data which was then used as input to the different methods of "GSAQ" package.

Comparison using TCGA's Low Grade Glioma (LGG) unbalanced Dataset

We applied our method "TSGS" with population size 100, number of iterations 100, level of significance 0.05, "bonferroni" method of adjusting p-values. We got 53 genes selected using our method "TSGS". We have used "svmRadial" method and "LOOCV" resampling method for building the training model and validation using "caret" R package. TSGS was compared with "t-score", "F-score", "MRMR" and "bootMRMR" methods available in "GSAQ" R package. Gene selection using each of these methods was performed. However, "t-score" and "F-score" methods were unable to select any genes as it cannot handle unbalanced data. Similarly, we also tried "MRMR" and "boot-MRMR" methods, but these methods also failed to select any genes. This is one of the limitations of "GSAQ" R package that it cannot handle unbalanced data whereas TSGS worked for unbalanced data also.

4.3 Development of Web tool:

We have developed a user-friendly tool TSGS (Trait Specific Gene Selection) for gene selection based on RNA-Seq expression data. We have implemented all the steps in R [77] and used “shiny” package [79] for developing the web application. Besides these, we have also used various Bioconductor packages [80]. The tool is available at https://icar-iasri.shinyapps.io/tsgs/

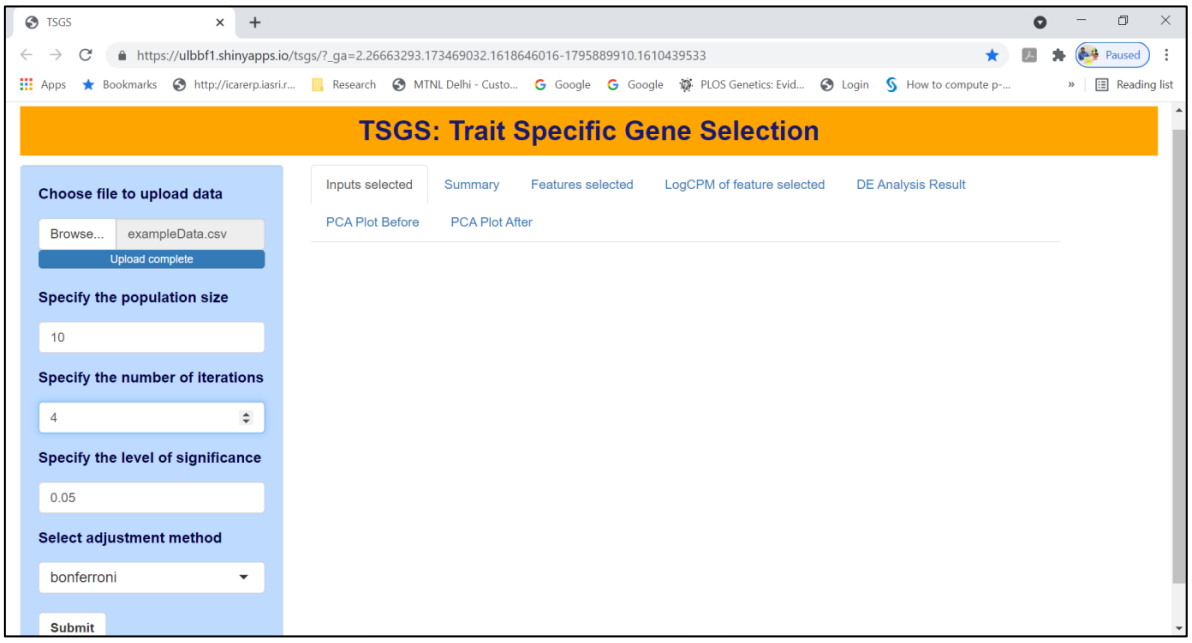


Fig 4.1 Home Page of TSGS

The user has to upload data in a specified format either in csv, tsv, txt, xls or xlsx format (Please see Figure 4.2). The sample names are specified in the first row starting from second position of row. The classes of each samples are specified in the second row corresponding to the sample names in first row. From third row and onwards, we have to put the RNA-Seq count data corresponding to genes in the first column and samples of the first row. A portion of KIRC RNA-Seq count data with 72 samples in each of the two classes is shown below:

	A	B	C	D	BT	BU	BV	BW	BX	BY	EO	EP	EQ
1		S1	S2	S3	...	S71	S72	S73	S74	S75	...	S143	S144
2	Class	0	0	0	...	0	0	1	1	1	...	1	1
3	?100130426	0	0	0	...	0	0	1	0	0	...	0	0
4	?100133144	6	3	3	...	40	16	21	26	37	...	26	2
5	?100134869	6	7	3	...	53	5	3	10	28	...	21	7
6	?10357	140	144	61	...	240	156	196	197	171	...	263	169
7	?10431	2042	1764	1766	...	3127	2456	1856	2280	2082	...	1963	1808
8	?136542	0	0	0	...	0	0	0	0	0	...	0	0
9	?155060	207	105	66	...	426	146	359	736	542	...	556	173
10	?26823	3	0	0	...	4	2	1	9	2	...	1	3
11	?280660	0	0	0	...	0	0	0	0	0	...	0	0
12	?317712	0	0	0	...	0	0	0	0	0	...	0	0
13	?340602	4	0	0	...	1	0	5	6	0	...	0	17
14	?388795	1	2	1	...	0	1	5	4	6	...	0	1
15	?390284	22	13	5	...	23	10	20	20	7	...	10	20
16	?391343	0	0	0	...	0	0	1	0	0	...	0	0
17	?391714	1	2	0	...	0	0	3	0	2	...	2	1
18	?404770	0	0	0	...	0	0	0	0	0	...	0	0
19	?441362	0	0	0	...	0	0	0	0	0	...	0	0
20	?442388	0	0	0	...	0	0	0	0	0	...	0	0
21	?553137	1812	979	803	...	3241	1492	1823	1944	1139	...	4126	886
22	?57714	998	1006	520	...	4835	2316	3048	2930	1597	...	7048	437
23	?645851	101	35	14	...	19	90	53	35	20	...	33	7

Fig 4.2 A portion of RNA-Seq count data

Then, the user has to specify other parameters, namely, population size, number of iterations, level of significance (default value 0.05) and method of adjusting p-values for multiple testing of genes. We have provided the following options for adjusting p-values: “BH”, “bonferroni”, “holm”, “Hochberg”, “hommel” and “BY”. The method “bonferroni” is the default adjustment method.

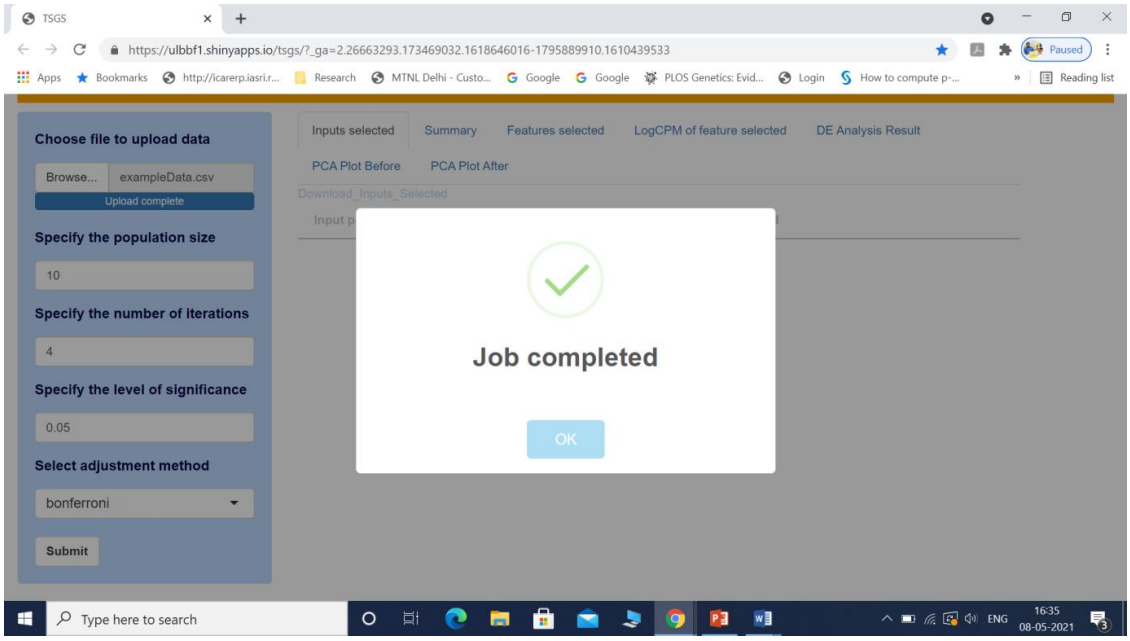


Fig 4.3 Status of Job

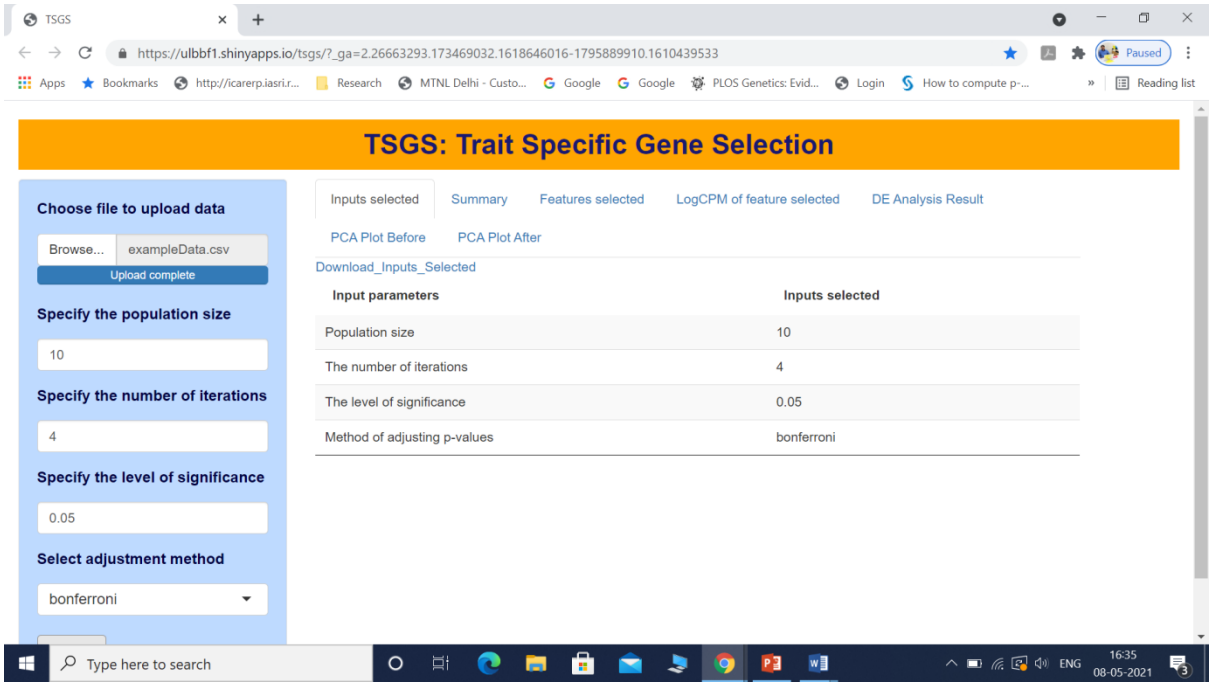


Fig 4.4 Parameters Selected

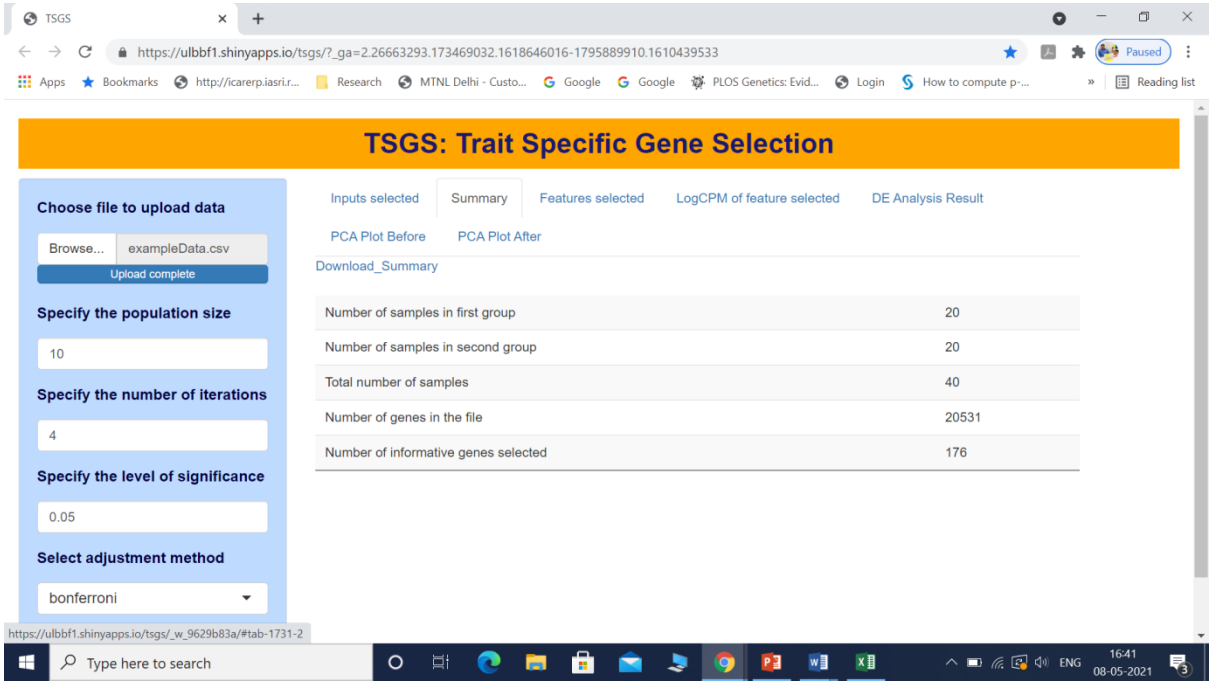


Fig 4.5 Summary

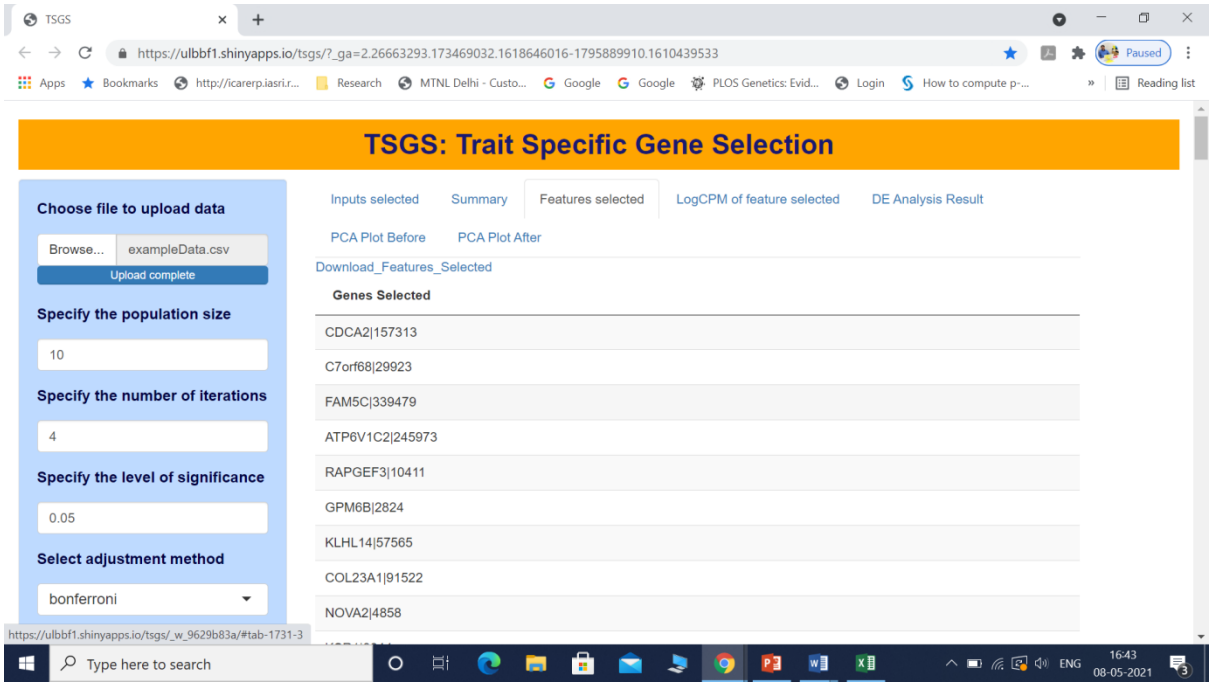


Fig 4.6 List of Selected Genes

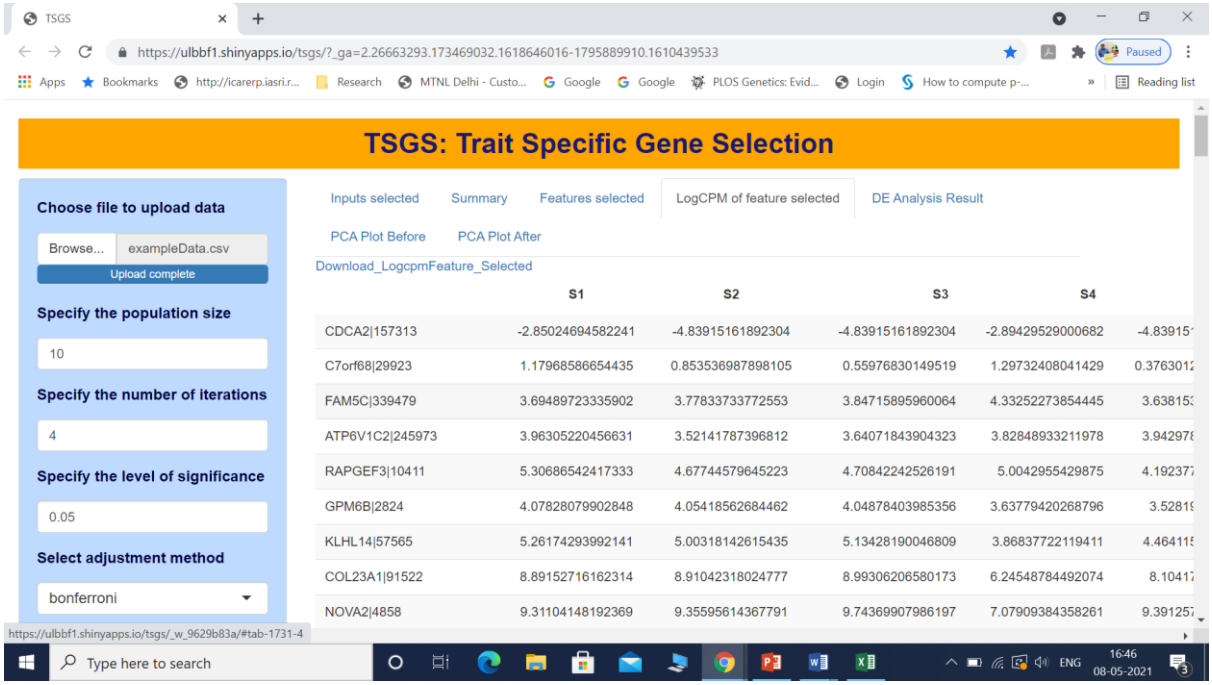


Fig 4.7 LogCPM of selected Genes

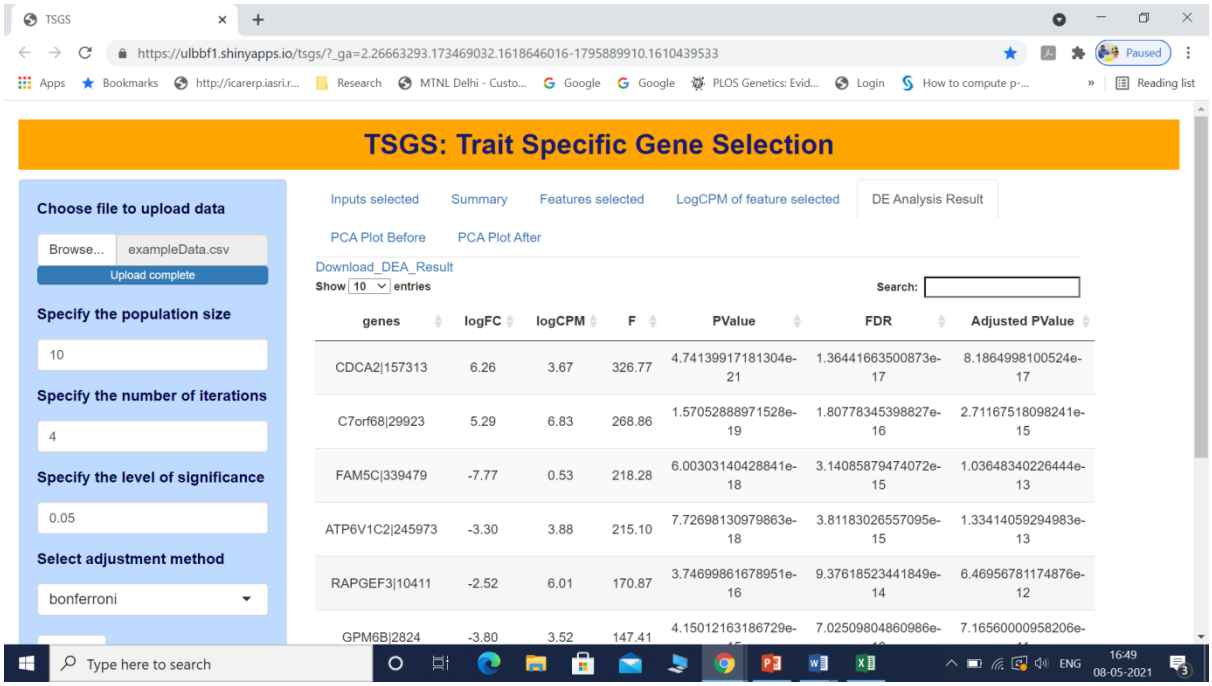
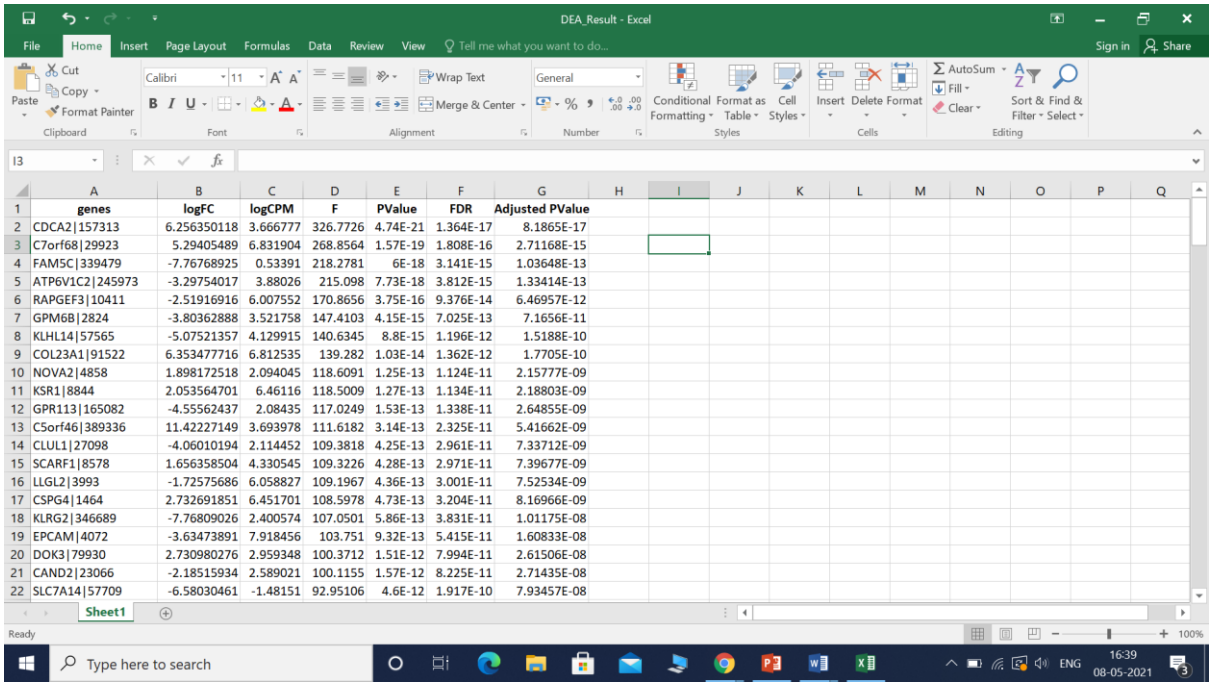


Fig 4.7a Differential Analysis Result



	genes	logFC	logCPM	F	PValue	FDR	Adjusted PValue
2	CDCA2|157313	6.256350118	3.666777	326.7726	4.74E-21	1.364E-17	8.1865E-17
3	C7orf68|29923	5.29405489	6.831904	268.8564	1.57E-19	1.808E-16	2.71168E-15
4	FAM5C|339479	-7.76768925	0.53391	218.2781	6E-18	3.141E-15	1.03648E-13
5	ATP6V1C2|245973	-3.29754017	3.88026	215.098	7.73E-18	3.812E-15	1.33414E-13
6	RAPGEF3|10411	-2.51916916	6.007552	170.8656	3.75E-16	9.376E-14	6.46957E-12
7	GPM6B|2824	-3.80362888	3.521758	147.4103	4.15E-15	7.025E-13	7.1656E-11
8	KLHL14|57565	-5.07521357	4.129915	140.6345	8.8E-15	1.196E-12	1.5188E-10
9	COL23A1|91522	6.353477716	6.812535	139.282	1.03E-14	1.362E-12	1.7705E-10
10	NOVA2|4858	1.898172518	2.094045	118.6091	1.25E-13	1.124E-11	2.15777E-09
11	KSR1|8844	2.053564701	6.46116	118.5009	1.27E-13	1.134E-11	2.18803E-09
12	GPR113|165082	-4.55562437	2.08435	117.0249	1.53E-13	1.338E-11	2.64855E-09
13	C5orf46|389336	11.42227149	3.693978	111.6182	3.14E-13	2.325E-11	5.41662E-09
14	CLUL1|27098	-4.06010194	2.114452	109.3818	4.25E-13	2.961E-11	7.33712E-09
15	SCARF1|8578	1.656358504	4.330545	109.3226	4.28E-13	2.971E-11	7.39677E-09
16	LLGL2|3993	-1.72575686	6.058827	109.1967	4.36E-13	3.001E-11	7.52534E-09
17	CSPG4|1464	2.732691851	6.451701	108.5978	4.73E-13	3.204E-11	8.16966E-09
18	KLRG2|346689	-7.76809026	2.400574	107.0501	5.86E-13	3.831E-11	1.01175E-08
19	EPCAM|4072	-3.63473891	7.918456	103.751	9.32E-13	5.415E-11	1.60833E-08
20	DOK3|79930	2.730980276	2.959348	100.3712	1.51E-12	7.994E-11	2.61506E-08
21	CAND2|23066	-2.18515934	2.589021	100.1155	1.57E-12	8.225E-11	2.71435E-08
22	SLC7A14|57709	-6.58030461	-1.48151	92.95106	4.6E-12	1.917E-10	7.93457E-08

Fig 4.7b Differential Analysis Result

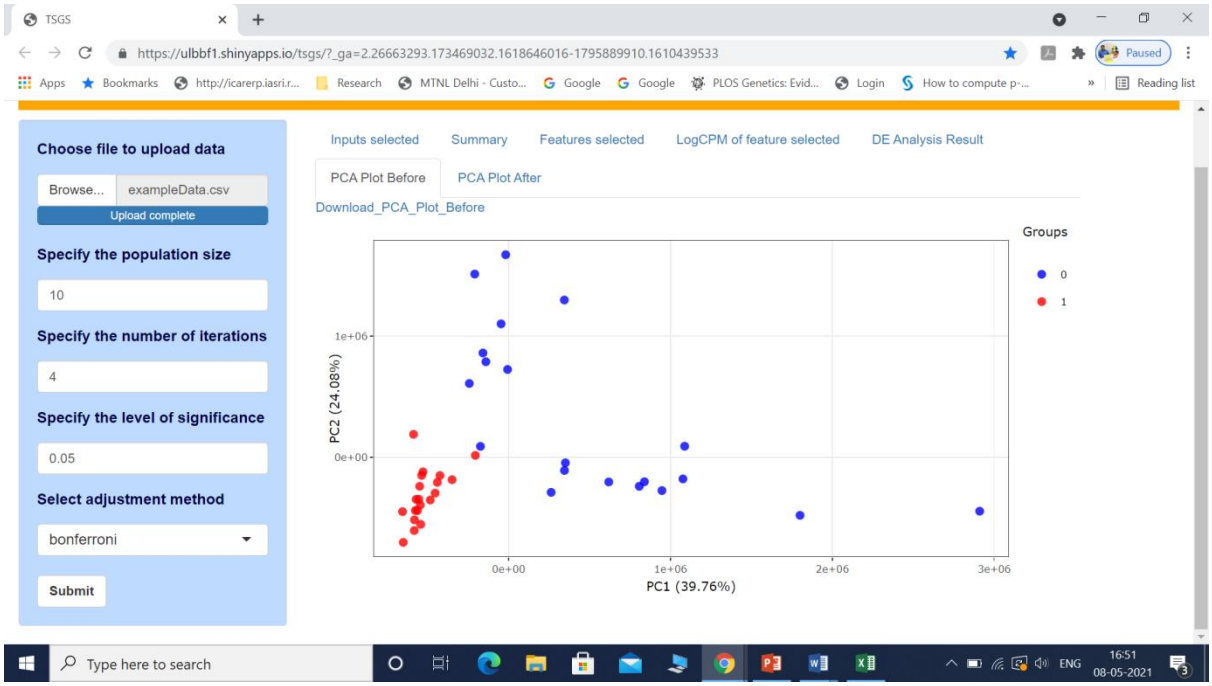


Fig 4.8a PCA Plot before

Fig 4.8b PCA Plot After

4.4 Interface for Biocomputing Portal

In order to provide access to users to analyse their data using the HPC facility at ASHOKA, TSGS has been made available at Biocomputing portal (ashoka.cabgrid.res.in:4443/pbsworks/login). The user can login to the portal with their credentials and access the tool and submit the job on the cluster.

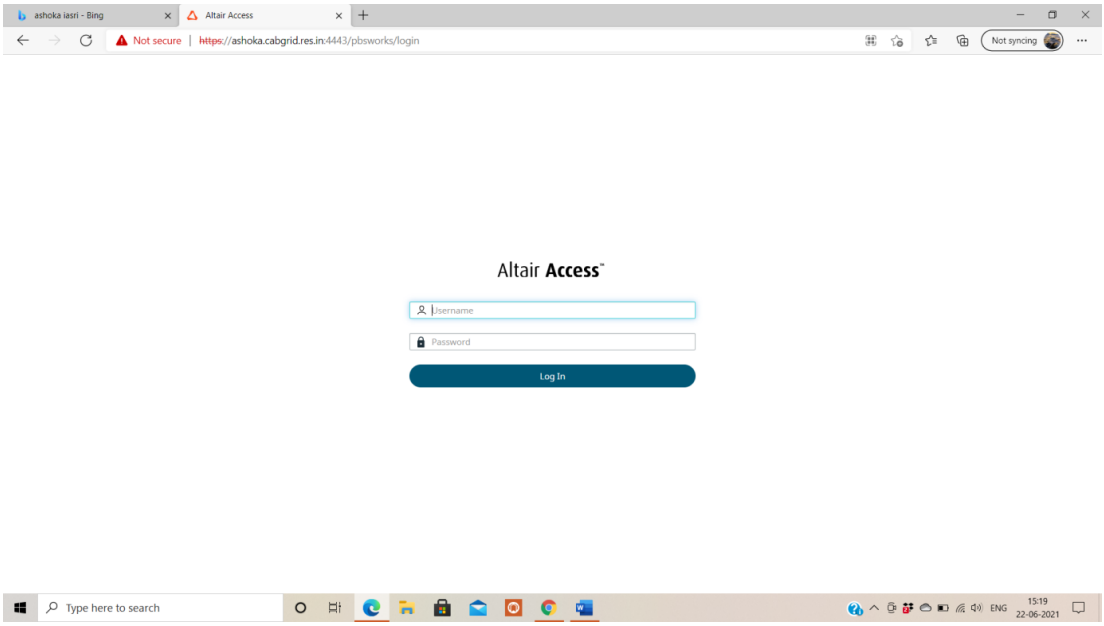


Fig 4.9a Login Screen for Biocomputing Portal

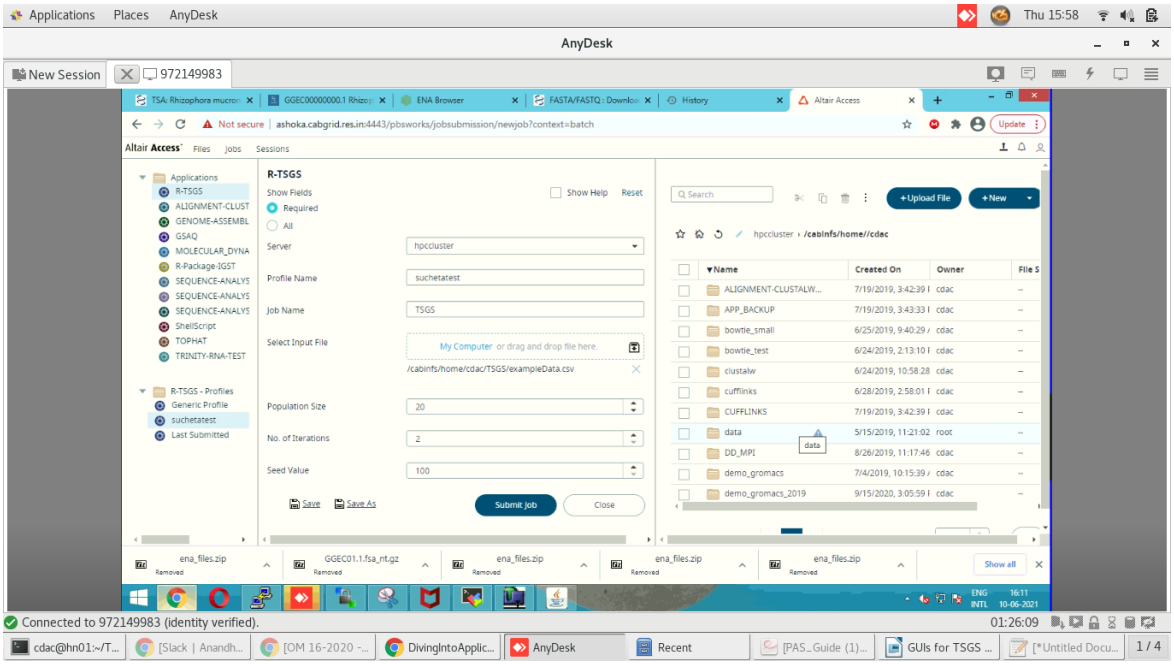


Fig 4.9b Accessing TSGS from Biocomputing Portal

Chapter 5: Conclusion

NGS and microarrays are a popular source of data for gathering gene expressions. Analysing these can be difficult due to the size of the data. The analysis of huge amount of gene expression data requires the execution of algorithm and tools to infer the hidden information or knowledge from these resources. Selection of informative genes from available high dimensional GE data is a challenging task. The complicated relations among the different genes make analysis more difficult and removing excess features can improve the quality of the results. Feature selection plays a bigger role in removing irrelevant features. In this study combination GA-SVM technique was used to develop a methodology which is a heuristic approach for informative gene selection from such GE data by considering gene relevance and redundancy simultaneously. In this methodology fitness value is used to evaluate individual chromosome present in the population. Fitness of the population can be increased in hope of producing child chromosomes with better genetic material in the subsequent generations. The average fitness could be improved by eliminating the unfit chromosomes in a population and replacing them with fitter chromosomes. Those chromosomes with the highest fitness values are given more opportunities to reproduce and the offspring share features taken from their parents. This ensures that the selected genes are carried to the next generation. The classification accuracy of the obtained gene set from the developed methodology was found to be better when compared with the gene sets obtained from methods such as Boot-MRMR, MRMR, t-score and F-score of R- package "GSAQ". The findings of this study will guide the genome researchers and experimental biologists to select informative gene set scientifically and objectively.

सारांश

पिछले कुछ दशकों में, अनुसंधान प्रयोगशालाओं सहित अधिकांश उद्योग में बड़ी मात्रा में डेटा उत्पन्न किया जा रहा है। प्रयोगात्मक डेटा की इस बढ़ती हुई मात्रा ने वैज्ञानिकों को ऐसी विधि और उपकरण विकसित करने में सक्षम बनाया है जो इस विशाल मात्रा में डेटा के विश्लेषण सहित पहुंच में आसानी प्रदान करते हैं। उच्च थ्रूपुट प्रौद्योगिकियों द्वारा उत्पन्न जैविक डेटा की मात्रा और प्रकार ने बड़े पैमाने पर डेटा हैंडलिंग और इसके विश्लेषण की कई चुनौतियों का सामना किया है। एनजीएस और माइक्रोएरे जीन अभिव्यक्ति एकत्र करने के लिए डेटा का एक लोकप्रिय स्रोत हैं। डेटा के आकार के कारण इनका विश्लेषण करना मुश्किल हो सकता है। इसके अलावा, विभिन्न जीनों के बीच जटिल संबंध विश्लेषण को और अधिक कठिन बनाते हैं और अतिरिक्त सुविधाओं को हटाने से परिणामों की गुणवत्ता में सुधार हो सकता है। अप्रासंगिक सुविधाओं को हटाने में फीचर चयन एक बड़ी भूमिका निभाता है। कई अलग-अलग फीचर चयन और फीचर निष्कर्षण विधियां मौजूद हैं और उनका व्यापक रूप से उपयोग किया जा रहा है। इन सभी विधियों का उद्देश्य अनावश्यक और अप्रासंगिक विशेषताओं को हटाना है ताकि नए उदाहरणों का वर्गीकरण अधिक सटीक हो सके।

उच्च आयामी जीन अभिव्यक्ति डेटा से सूचनात्मक जीन का चयन ट्रांसक्रिपटामिक में एक महत्वपूर्ण अनुसंधान क्षेत्र के रूप में उभरा है। संपूर्ण जीनोम ट्रांसक्रिप्टोम विश्लेषण में RNA-Seq दृष्टिकोण के साथ प्रमुख मुद्दों में से एक यह है कि, विभिन्न विभिन्न जीनों की अभिव्यक्ति की गतिशीलता पर कब्जा कर लिया जाता है। इसका परिणाम डेटा में बहुत उच्च आयामीता है, जिसका अर्थ है कि जीन की संख्या नमूनों की संख्या से बहुत अधिक है। इसलिए, उपयुक्त कम्प्यूटेशनल दृष्टिकोणों की सहायता से हजारों जीनों में से स्थिति वर्ग से संबंधित सबसे प्रासंगिक जीन का चयन करना महत्वपूर्ण है।

पिछले एक दशक में कई फीचर सेलेक्शन एल्गोरिदम (एफएसए) पेश किए गए हैं, लेकिन उनमें से ज्यादातर बड़ी संख्या में बेमानी सुविधाओं के साथ उच्च-आयामी डेटासेट पर अच्छा प्रदर्शन नहीं करते हैं। इस प्रकार वर्तमान परियोजना में, जीन अभिव्यक्ति डेटा से विशेषता विशिष्ट जीन के प्रासंगिक सेट प्राप्त करने के लिए कार्यप्रणाली विकसित करने की योजना बनाई गई थी। इस परियोजना के तहत दो पारंपरिक मशीन लर्निंग एल्गोरिदम, सपोर्ट वेक्टर मशीन (एसवीएम) और एक जेनेटिक एल्गोरिथम (जीए) के संयोजन को लागू करके विशेषता विशिष्ट जीन चयन उपकरण (टीएसजीएस) विकसित किया गया है। वे एक आवरण दृष्टिकोण के आधार पर प्रभावी

ढंग से एकीकृत होते हैं। GA का उपयोग वर्गीकरण और मूल्यांकन के लिए SVM को भेजे गए जीन के सबसेट को नियंत्रित और अनुकूलित करने के लिए किया जाता है। एसवीएम को क्लासिफायर प्रदर्शन के रूप में और फीचर चयन के लिए जेनेटिक एल्गोरिदम का उपयोग करके सूचनात्मक जीन सेट का एक सेट प्राप्त किया जा सकता है। विकसित पद्धति से प्राप्त जीन सेट की वर्गीकरण सटीकता की तुलना बूट-एमआरएमआर, एमआरएमआर, टी-स्कोर और आर-पैकेज "जीएसएक्यू" के एफ-स्कोर जैसी विधियों से प्राप्त जीन सेट से की गई थी।

TSGS की आसान उपलब्धता के लिए उपयोगकर्ता को शाइनी ऐप का उपयोग करने वाला एक वेब टूल बनाया गया है, आगे टूल को बायोकंप्यूटिंग पोर्टल के माध्यम से भी एक्सेस प्रदान किया जाता है ताकि उपयोगकर्ता अशोका का उपयोग करके अपने उच्च आयामी जीन अभिव्यक्ति डेटा का विश्लेषण कर सकें।

Summary

In last few decades, huge amount of data is being generated in most of the industry including research labs. This ever-increasing amount of experimental data has enabled the scientists to develop the method and tools that provide an ease of access including analysis of this huge amount of data. The amount and type of biological data generated by high throughput technologies have posed many challenges of large-scale data handling and its analysis. NGS and microarrays are a popular source of data for gathering gene expressions. Analysing these can be difficult due to the size of the data. In addition, the complicated relations among the different genes make analysis more difficult and removing excess features can improve the quality of the results. Feature selection plays a bigger role in removing irrelevant features. Many different feature selection and feature extraction methods exist and they are being widely used. All these methods aim to remove redundant and irrelevant features so that classification of new instances will be more accurate.

Selection of informative genes from high dimensional gene expression data has emerged as an important research area in transcriptomic. One of the major issues with the RNA-Seq approach in whole genome transcriptome analysis is that, the expression dynamics of various different genes are captured. This result in very high dimensionality in the data, which means the number of genes is much larger than the number of samples. Therefore, it is important to select most relevant genes related to condition class from thousands of genes with the help of appropriate computational approaches.

Many feature selection algorithms (FSA) are introduced in past decade but most of them do not perform well on high-dimensional datasets with a large number of redundant features. Thus in the present project, it was planned to develop the methodology for obtaining relevant set of trait specific genes from gene expression data. Under this project trait specific gene selection tool (TSGS) has been developed by applying combination of two conventional machine learning algorithms, support vector machine (SVM) and a genetic algorithm (GA). They are integrated effectively based on a wrapper approach. GA is used to control and optimize the subset of genes sent to the SVM for classification and evaluation. Using SVM as the classifier performance and the Genetic algorithm for feature selection a set of informative gene set can be obtained. The classification accuracy of the obtained gene set from the developed methodology was compared with the gene sets obtained from methods such as Boot-MRMR, MRMR, t-score and F-score of R- package “GSAQ”.

For the easy availability of the TSGS the user a web tool using shiny app has been created further the tool is also provided access through Biocomputing portal for the user to analyse their high dimensional gene expression data using ASHOKA.

References:

1. Schneider MV, Orchard S. Omics technologies, data and bioinformatics principles. Bioinformatics for Omics Data: Methods and Protocols. 2011; p. 3-30.

2. Alyass A, Turcotte M, Meyre D. From big data analysis to personalized medicine for all: challenges and opportunities. BMC medical genomics. 2015; 8(1):33. https://doi.org/10.1186/s12920-015-0108-y PMID: 26112054.

3. Slonim DK, Yanai I. Getting started in gene expression microarray analysis. PLoS Comput Biol. 2009; 5 (10):e1000543. https://doi.org/10.1371/journal.pcbi.1000543 PMID: 19876380.

4. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences. 2005; 102(43):15545-15550. https://doi.org/10.1073/pnas. 0506580102.

5. Slonim DK. From patterns to pathways: gene expression data analysis comes of age. Nature genetics. 2002; 32:502-508. https://doi.org/10.1038/ng1033 PMID: 12454645.

6. Osareh A, Shadgar B. Classification and diagnostic prediction of cancers using gene microarray data analysis. Journal of Applied Sciences. 2009; 9(3):459-468. https://doi.org/10.3923/jas.2009.459.468.

7. Armstrong SA, Staunton JE, Silverman LB, Pieters R, den Boer ML, Minden MD, et al. MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. Nature genetics. 2002; 30(1):41- 47. https://doi.org/10.1038/ng765 PMID: 11731795.

8. Pomeroy SL, Tamayo P, Gaasenbeek M, Sturla LM, Angelo M, McLaughlin ME, et al. Prediction of central nervous system embryonal tumour outcome based on gene expression. Nature. 2002; 415 (6870):436-442. https://doi.org/10.1038/415436a PMID: 11807556

9. Glinsky GV, Glinskii AB, Stephenson AJ, Hoffman RM, Gerald WL. Gene expression profiling predicts clinical outcome of prostate cancer. The Journal of clinical investigation. 2004; 113(6):913-923. https://doi.org/10.1172/JCI20032 PMID: 15067324

10. Kurun, O., Akar, C.O., Favorov, O., Aydin, N., Urgen, F.. Using covariates for improving the minimum redundancy maximum relevance feature selection method. Turkish Journal of Electrical Engineering and Computer Sciences 2010;18(6):975–987.

11. Kamandar, M., Ghassemian, H.. Maximum relevance, minimum redundancy band selection for hyperspectral images. In: 19th Iranian Conference on Electrical Engineering (ICEE),. 2011,

12. Dy, J.G., Brodley, C.E., Kak, A., Broderick, L.S., Aisen, A.M.. Unsupervised feature selection applied to content-based retrieval of lung images. IEEE Transaction on Pattern Analysis and Machine Intellegence 2003;25(3):373–378.

13. Zhang, Z., R.Hancock, E.. A graph-based approach to feature selection. In: International Workshop on Graph-Based Representations in Pattern Recognition. 2011,

14. Cai, D., Zhang, C., He, X.. Unsupervised feature selection for multi-cluster data. In: 16th ACM SIGKDD International Conference on Knowledge Discovery and Data mining. 2010,
15. Ruiza, R., Riquelmea, J.C., Aguilar-Ruizb, J.S.. Incremental wrapper-based gene selection from microarray data for cancer classification. Pattern Recognition 2006;39(12):2383–2392.
16. Mitra, P., Murthy, C., Pal, S.K.. Unsupervised feature selection using feature similarity. IEEE Transaction on Pattern Analysis and Machine Intellegence 2002;24(3):301–312.
17. Sondberg-Madsen, N., Thomsen, C., Pena, J.M.. Unsupervised feature subset selection. In: In Proc. of the Workshop on Probabilistic Graphical Models for Classification. 2003,
18. Ding, C.H.Q.. Unsupervised feature selection via two-way ordering in gene expression analysis. Bioinformatics 2003;19(10):1259–1266.
19. Kohavi, R., John., G.. Wrapper for feature subset selection. Artificial Intelligence 1997;97:273–324.
20. Jiang, S., Wang, L.. An unsupervised feature selection framework based on clustering. In: New Frontiers in Applied Data Mining. 2008.
21. Morita, M., Oliveira, L.S., Sabourin, R.. Unsupervised feature selection for ensemble of classifiers. In: Frontiers in Handwriting Recognition. 2004.
22. Handl, J., Knowles, J.. Feature subset selection in unsupervised learning via multi objective optimization. International Journal of Computational Intelligence Research 2006;2(3):217–238.
23. Dash, M., Liu, H.. Unsupervised feature selection. In: In Proc. Pacific Asia Conf. Knowledge Discovery and Data Mining. 2000,
24. Lee, I.H., Lushington, G.H., Visvanathan, M.. A filter-based feature selection approach for identifying potential biomarkers for lung cancer. Journal of Clinical Bioinformatics 2011;1(11).
25. Li, J., Tang, X., Zhao, W., Huang, J.. A new framework for identifying differentially expressed genes. Pattern Recognition 2007;40:3249–3262.
26. Van Der Maaten L, Postma E & Van Den Herik J (2009) Tilburg centre for Creative Computing Dimensionality Reduction: A Comparative Review Dimensionality Reduction: A Comparative Review. Available at:http://www.uvt.nl/ticc.
27. R. E. Bellman, Dynamic Programming, Princeton University Press, Princeton, NJ, USA, 1957.
28. C.Arun Kumara, Sooraj M. P., S. Ramakrishnan. A Comparative Performance Evaluation of Supervised Feature Selection Algorithms on Microarray Datasets. Procedia Computer Science 115 (2017) 209–217.
29. Bolon-Canedo Verónica, et al. A review of feature selection methods on synthetic data. Knowledge and information systems. Springer 2013; 34 (3): 483-519.
30. Guyon Isabelle, et al. Gene selection for cancer classification using support vector machines. Machine learning. Springer 2002; 46: 389-422.

31. Maldonado Sebastián, et al. Simultaneous feature selection and classification using kernel-penalized support vector machines. Information Sciences. Elsevier 2011; 181 (1): 115-128.
32. Breiman L. Random forests. Mach Learn 2001; 45:5–32.
33 Dı´az-Uriarte R, De Andres SA. Gene selection and classification of microarray data using random forest. BMC Bioinformatics 2006;7:3.
34. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc Series B Stat Methodol 1996;58:267–88.
35. Zou H, Hastie T. Regularization and variable selection via the elastic net. J R Stat Soc Series B Stat Methodol 2005;67:301–20.
36. Pineda S, Real FX, Kogevinas M, et al. Integration analysis ofthree omics data using penalized regression methods: an application to bladder cancer. PLoS Genet 2015;11:e1005689.
37. Wu TT, Chen YF, Hastie T, et al. Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics 2009;25:714–21.
38. Neto EC, Bare JC, Margolin AA. Simulation studies as designed experiments: the comparison of penalized regression models in the "large p, small n" setting. PLoS One 2014;9:e107957.
39. Waldron L, Pintilie M, Tsao M-S, et al. Optimized application of penalized regression methods to diverse genomic data. Bioinformatics 2011;27:3399–406.
40. Xiao, Y., Hsiao T. H., Suresh, U., Chen H.,I., H.,Wu, X., et al., (2014), 'A novel significance score for gene selection and ranking', Bioinformatics, Vol.30, pp.801–807.
41. Zhang HH, Ahn J, Lin X, Park C. Gene selection using support vector machines with non-convex penalty. bioinformatics. 2006; 22(1):88-95. https://doi.org/10.1093/bioinformatics/bti736 PMID: 16249260.
42. Shevade SK, Keerthi SS. A simple and efficient algorithm for gene selection using sparse logistic regression. Bioinformatics. 2003; 19(17):2246-2253. https://doi.org/10.1093/bioinformatics/btg308 PMID: 14630653.
43. Huang HH, Liu XY, Liang Y. Feature Selection and Cancer Classification via Sparse Logistic Regression with the Hybrid L 1/2- 2 Regularization. PloS one. 2016; 11(5):e0149675. https://doi.org/10.1371/ journal.pone.0149675 PMID: 27136190.
44. Ai-Jun Y, Xin-Yuan S. Bayesian variable selection for disease classification using gene expression data. Bioinformatics. 2010; 26(2):215-222. https://doi.org/10.1093/bioinformatics/btp638.
45. Han F, Yang C, Wu YQ, Zhu JS, Ling QH, Song YQ, et al. A Gene Selection Method for Microarray Data Based on Binary PSO Encoding Gene-to-Class Sensitivity Information. IEEE/ACM Transactionson Computational Biology and Bioinformatics (TCBB). 2017; 14(1):85-96. https://doi.org/10.1109/TCBB.2015.2465906.
46. Hinton G, Deng L, Yu D, Dahl GE, Mohamed Ar, Jaitly N, et al. Deep neural networks for acoustic modelling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine. 2012; 29(6):82-97. https://doi.org/10.1109/MSP.2012.2205597

47. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems; 2012. p. 1097-1105.
48. Wainwright MJ, Jordan MI, et al. Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning. 2008; 1(1-2):1-305.
49. Koller D, Friedman N. Probabilistic graphical models: principles and techniques. MIT press; 2009.
50. Zhu J, Chen N, Xing EP. Bayesian inference with posterior regularization and applications to infinite latent SVMs. Journal of Machine Learning Research. 2014; 15(1):1799-1847.
51. Ghahramani Z, Griffiths TL. Infinite latent feature models and the Indian buffet process. In: Advances in neural information processing systems; 2006. p. 475-482.
52. Carlson TA, Schrater P, He S. Patterns of activity in the categorical representations of objects. Journal of cognitive neuroscience. 2003; 15(5):704-717. https://doi.org/10.1162/jocn.2003.15.5.704 PMID: 12965044.
53. Mitchell TM, Hutchinson R, Niculescu RS, Pereira F, Wang X, Just M, et al. Learning to decode cognitive states from brain images. Machine learning. 2004; 57(1):145-175. https://doi.org/10.1023/B: MACH.0000035475.85309.1b.
54. Liu B, Wu H, Zhang D, Wang X, Chou KC. Pse-Analysis: a python package for DNA/RNA and protein/ peptide sequence analysis based on pseudo components and kernel methods. Oncotarget. 2017; 8(8):13338. https://doi.org/10.18632/oncotarget.14524 PMID: 28076851.
55. Chang Cc, Lin H. A library for support vector machines. 2007.
56. Tipping ME. Sparse Bayesian learning and the relevance vector machine. Journal of machine learning research. 2001; 1(Jun):211-244.
57. Pan W. Bayesian learning for nonlinear system identification. Imperial College London; 2015.
58. Sparse Bayesian classification and feature selection for biological data PLOSONE https://doi.org/10.1371/journal.pone.0189541 December 27, 2017 17 / 18.
59. Szymczak S, Biernacka JM, Cordell HJ, et al. Machine learning in genome-wide association studies. Genet Epidemiol 2009;33:S51–7.
60. Alexe G, Monaco J, Doyle S, et al. Towards improved cancer diagnosis and prognosis using analysis of gene expression data and computer aided imaging. Exp BiolMed 2009;234:860–79.
61. Frauke Degenhardt, Stephan Seifert and Silke Szymczak. Evaluation of variable selection methods for random forests and omics data sets. Briefings in Bioinformatics, 2017, 1–12 4.
62. Wilhelm T. Phenotype prediction based on genome-wide DNA methylation data. BMC Bioinformatics 2014;15:193.

63. Swan AL, Mobasheri A, Allaway D, et al. Application of machine learning to proteomics data: classification and biomarker identification in post genomics biology. Omics 2013;17: 595–610.
64. Smolinska A, Hauschild A-C, Fijten R, et al. Current breathomics—a review on data pre-processing techniques and machine learning in metabolomics breath analysis. J Breath Res 2014;8:027105.
65. Maji Pradipta, Paul Sushmita, Rough set based maximum relevance-maximum significance criterion and Gene selection from microarray data, International Journal of Approximate Reasoning, Volume 52, Issue 3, 2011,Pages 408-426.
66. J. Li, H. Su, H. Chen, B.W. Futscher, Optimal search-based gene subset selection for gene array cancer classification, IEEE Transactions on Information Technology in Biomedicine 11 (4) (2007) 398–405.
67. Ding C, Peng H. Minimum redundancy feature selection from microarray gene expression data. J Bioinforma Comput Biol. 2005;03(02):185–205.
68. H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of Max-Dependency,Max-Relevance, and Min-Redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1226–1238.
69. Anjum, Arfa & Jaggi, Seema & Varghese, Eldho & Lall, Shwetank & Bhowmik, Arpan & Rai, Anil. (2016). Identification of Differentially Expressed Genes in RNA-seq Data of Arabidopsis thaliana: A Compound Distribution Approach. Journal of Computational Biology. 23. 10.1089/cmb.2015.0205.
70. Al-Shahrour, F., Diaz-Uriarte, R., Dopazo, J., 2005. Discovering molecular functions significantly related to phenotypes by combining gene expression data and biological information. Bioinformatics 21,2988–2993.
71. E. B. Huerta, B. Duval, and J.-K. Hao, “Gene selection for microarray data by a LDA based genetic algorithm,”. M. Chetty, A. Ngom, and S. Ahmad, Eds., vol. 5265 of Lecture Notes in Computer Science, pp. 250–261, Springer, Berlin, Germany, 2008.
72. Hira, Z. M., & Gillies, D. F. (2015). A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data. Advances in Bioinformatics, 2015, 1–13.doi:10.1155/2015/198363.
73. Edgar R, Domrachev M, Lash AE (January 2002). "Gene Expression Omnibus: NCBI gene expression and hybridization array data repository". Nucleic Acids Research. 30 (1): 207–10. doi:10.1093/nar/30.1.207
74. Chen Y, Lun ATL, and Smyth, GK (2016). From reads to genes to pathways: differential expression analysis of RNA-Seq experiments using Rsubread and the edgeR quasi-likelihood pipeline. F1000Research 5, 1438. http://f1000research.com/articles/5-1438
75. Robinson MD, McCarthy DJ, Smyth GK (2010). “edgeR: a Bioconductor package for differential expression analysis of digital gene expression data.” *Bioinformatics*, 26(1), 139-140. doi: 10.1093/bioinformatics/btp616.
76. Robinson MD, Oshlack A (2010). A scaling normalization method for differential expression analysis of RNA-seq data. Genome Biology 11, R25.

77. Robinson MD and Smyth GK (2008). Small-sample estimation of negative binomial dispersion, with applications to SAGE data. Biostatistics, 9, 321-332. http://biostatistics.oxfordjournals. org/content/9/2/321
78. R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
79. Winston Chang, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert and Barbara Borges (2021). shiny: Web Application Framework for R. R package version 1.6.0. https://CRAN.R-project.org/package=shiny
80. W. Huber, V.J. Carey, R. Gentleman, ..., M. Morgan (2015). Orchestrating high-throughput genomic analysis with Bioconductor. Nature Methods, **12**, 115-121.
81. Das, S., Meher, P.K., Rai, A., Bhar, L.M., Mandal, B.N., (2017). Statistical approaches for gene selection, hub gene identification and module interaction in gene co-expression network analysis: An application to Aluminum stress in Soybean (Glycine max L.). PLoS One 12(1), e0169605.
82. Martin Morgan and Lori Shepherd (2021). ExperimentHub: Client to access ExperimentHub resources. R package version 1.16.1.