Web-Based Application for Accurately Classifying Cancer Type from Microarray Gene Expression Data Using a Support Vector Machine (SVM) Learning Algorithm

Shrikant Pawar[1,2(✉)]

[1] Department of Computer Science, Georgia State University, 34 Peachtree Street, Atlanta, GA 30303, USA
spawar2@gsu.edu
[2] Department of Biology, Georgia State University, 34 Peachtree Street, Atlanta, GA 30303, USA

Abstract. Intelligent optimization algorithms have been widely used to deal complex nonlinear problems. In this paper, we have developed an online tool for accurate cancer classification using a SVM (Support Vector Machine) algorithm, which can accurately predict a lung cancer type with an accuracy of approximately 95%. Based on the user specifications, we chose to write this suite in Python, HTML and based on a MySQL relational database. A Linux server supporting CGI interface hosts the application and database. The hardware requirements of suite on the server side are moderate. Bounds and ranges have also been considered and needs to be used according to the user instructions. The developed web application is easy to use, the data can be quickly entered and retrieved. It has an easy accessibility through any web browser connected to the firewall-protected network. We have provided adequate server and database security measures. Important notable advantages of this system are that it runs entirely in the web browser with no client software need, industry standard server supporting major operating systems (Windows, Linux and OSX), ability to upload external files. The developed application will help researchers to utilize machine learning tools for classifying cancer and its related genes. Availability: The application is hosted on our personal linux server and can be accessed at: http://131.96.32.330/login-system/index.php.

Keywords: Cancer · Microarray · Support Vector Machine (SVM)

1 Introduction

Many machine learning algorithms such as random forest, k-nearest neighbor, neural network, and SVM (Support Vector Machine) have been applied to the classification study of bioinformatics. SVM has distinctive advantages in handling data with high dimensionality and a small sample size. Expression levels of genes can be analyzed using Microarray experiments. In this technique RNA is isolated from different tissues which is labeled and hybridized to the arrays [1]. The expression levels of treatment sample can be compared to control sample to differences in gene expression levels

I. Rojas et al. (Eds.): IWBBIO 2019, LNBI 11466, pp. 149–154, 2019.
https://doi.org/10.1007/978-3-030-17935-9_14

amongst two treatments. Microarrays can be of two type: single channel where only one dye, say red or green dye is used and two channel microarray where tow dyes at same time are used, one for control (CY5) and one for treatment (CY3). Different array spots give different fluorescence intensity values for each gene which can be analyzed with different Bioconductor package in R to perform normalization and statistical analysis.

Normalization is needed prior to making biological comparisons. It is required as the RNA used for hybridization can be of different quantities or there can different labeling conditions for different probes or the expression level scanning may be biased. Normalization essentially adjusts the expression levels of probes to reference probes [1]. Figure 1 is the one channel microarray lung cancer data file after normalization. After normalization a significant change in expression levels with correction of standard deviation and variance within and between samples states the importance of normalization. Four lung cancer samples were read in R and box plots were made on files to measure standard deviation and variance on raw data and RMA normalized data, a significant variance stabilization is seen after normalization as shown in Fig. 1.

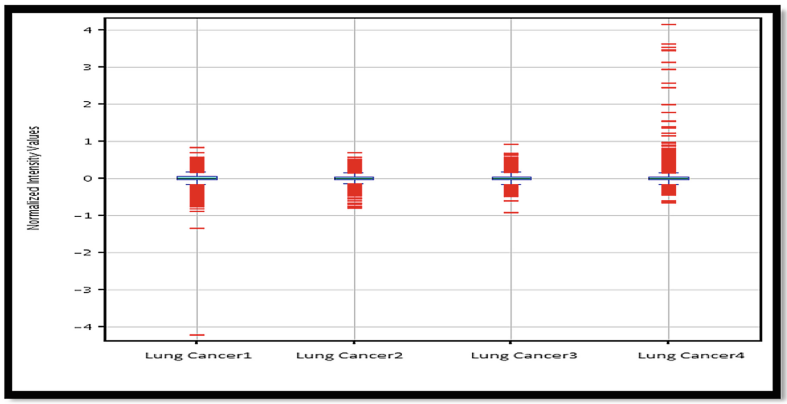

Fig. 1. Four lung cancer samples were read in R and box plots were made on files after RMA normalization to measure standard deviation and variance.

A SVM is a machine learning technique which categorizes new samples based on supervised learning of training data. A hyperplane is defined which forms a minimum distance in training sample which is used for classification [2]. Support Vector Machine makes classifications based on non-probabilistic binary linear classifier [3]. It has been shown that SVM's can significantly increase accuracy compared to traditional query refinement schemes [3]. Although, several groups have shown importance of machine learning tools in classifying big data, there still remains a constant need to develop a user friendly tool do perform its application. This paper summarizes a developed functional user tool, which can input a one channel microarray data and predict accurately the cancer type with machine learning technique (Fig. 2).

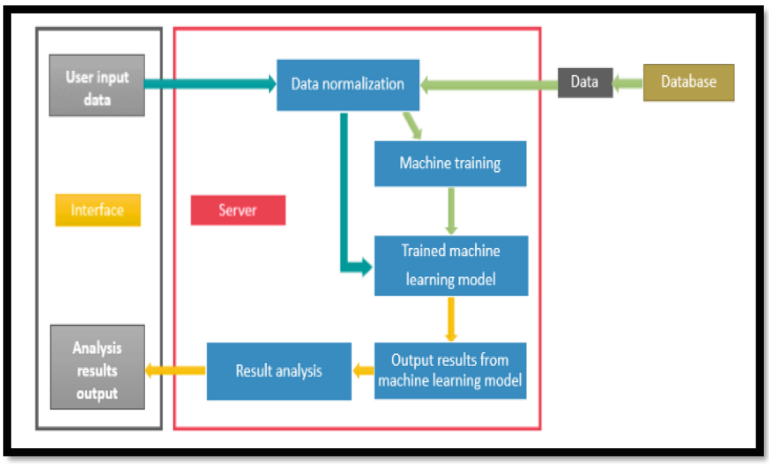

Fig. 2. Depicts the work flow for this project. User inputs a microarray data, it is downloaded in database, normalized, analyzed by SVM, and the results of which thrown on HTML page.

2 Materials and Methods

a. Data Collection

103 one channel microarray lung cancer samples (10,000 genes per channel) (GPL590) were downloaded from Gene Expression Omnibus (GEO) database. Similar number of normal lung tissue microarray data accompanied cancer data. The downloaded raw data needs normalization for variance stabilization.

b. Normalization

Normalization was tried using two techniques mas5.0 and RMA normalization. RMA normalization was performed for the samples [4]. It initially reads the affymetrix data using AFFY package in R language, we then perform Mas5.0 normalization and expression levels are extracted using exprs() function. Since it's a no logarithmic transformed data we transform that to logarithm to base 2 and export that to excel file for further analysis.

c. Machine Learning Analysis

From sklearn.svm module was imported with classification parameters as C = 1.0, cache_size = 200, class_weight = None, coef0 = 0.0, decision_function_shape = None, degree = 3, gamma = 'auto', kernel = 'rbf', max_iter = −1, probability = True, random_state = None, shrinking = True, tol = 0.001, verbose = False to perform SVM on normalized data [5]. Classification is done using support vector machines (SVMs). These are set of supervised learning methods useful with high dimensional spaces where dimensions are greater than number of samples [6]. It is also memory efficient and uses subset of training points in the decision function [6]. Initially a training set of 103 cancer samples was trained and results were stored on server space on local machine. This data will be used to perform predictions on query data.

d. Setting up a Server Space and developing a Graphical User Interface (GUI)

GUI was developed by importing module Tkinter. 200 × 100 dimension frame was made for uploading files from the user. Server was developed with Python using BaseHTTPServer module, other classes like re, os, Classifier, Constants, normalize were declared in main class MyHandler. Three main methods are declared, def do_GET(self) responding to a GET request, def do_POST(self) calls normalization and classifier function and def predict_data(self, test_data) responds to a GET request. Def get_html_response_after_prediction(self, result, fileName) method writes the results on HTML page after predictions have been made. BaseHTTPServer module creates and listens at the HTTP socket, *Re* module accepts regular expression, *Os* module allows to interface with operating system. Class constants defines HOST_NAME = '10.241.213.126', PORT_NUMBER = 9000 and paths for trained data. Based on the user specifications, we chose to write this suite in Python CGI and based on a MySQL database. It enables the set-up of single or a multi-users access controls. A Linux server hosts the application and database. It has been developed on a Mac OS X operating system using MySQL as the relational database management system and Python as the scripting language. The hardware requirements of suite on the server side are moderate. The server we utilized is GSU Orion with CentOS 6.7 64-bit, 6x IBM System x3850 x5, Intel Xeon Processor E7-4850, 4 CPUs (10 cores per CPU), 2.0 GHz processors, 512 GB RAM and 2 TB of scratch storage for jobs. The database has a column text-field table which is updated interactively with the user. Further, the uploading of information is standardized as certain parameters like date has to be inputted in specific format only which helps in retrieval process. Checking of data type (float, integer, text, Boolean) and dates is important before putting into database. Errors with dates or invalid parameters or wrong data type cause a halt of workflow. Bounds and ranges have also been considered and needs to be used according to the user instructions.

e. Security

We have implemented strong security measures for authentication of SQL server. Kerberos protocol uses a number of encrypted messages to authenticate SQL server and the passwords are not passed across the network. Authentication is more reliable and managing it can be reduced by leveraging active directory groups for role- based access to SQL server. The sysadmin (sa) account is vulnerable when it exits unchanged so we have disabled the sa account on the SQL server instance. We chose to give options of complex passwords for sa and all other SQL-server-specific logins on SQL server and checked in the 'Enforce password expiration' and 'Enforce password policy' options for sa. We haven't allowed to explicit grant control server permission because logins with this permission get full administrative privileges. For permissions to users, an built-in fixed server roles and database roles or creating own custom server roles and database roles can be achieved. Guest user exists in every user and system database, which is a potential security risk in a lock down environment because it allows database access to logins who don't have associated users in the database. We have restricted this access and also accesses to user and system stored procedures. Furthermore, we have used common specific TCP ports (excluding 1433 and 1434) instead of dynamic ports. SQL server browser service is only running on SQL servers, and secure SQL server error logs

and registry keys using NTFS permissions are utilized as they can provide great deal of information about the SQL server instance and installation.

3 Results and Conclusions

An HTML page is made to provide options to the user for uploading data. Once the user uploads a .CEL file it calls functions in server, does machine learning and uploads results in same HTML page. A sample lung cancer .CEL file (GSM159355) was uploaded by user as a query sample and our prediction analysis gave a prediction of 94.88% to be a lung cancer type, a robust multichip analysis was performed and the results are shown in Table 1. This application is a primary skeleton to dig complex gene patterns via SVM [7–9]. Such open source skeletons with large server space, interface and algorithm implementations are currently unavailable and this is an attempt to provide it to the user in a friendly and cost effective interface, which will further be expanded to apply machine learning algorithms to understand complex gene patterns [10].

Table 1. A robust multichip analysis was performed on lung cancer chips and the prediction analysis (accuracy) results recorded as follows.

Cancer type	GSM series ID	Accuracy (%)
Lung	GSM159355	94.88
Lung	GSM1322678	93.08
Lung	GSM1322679	92.11
Lung	GSM1322680	91.05
Lung	GSM1322681	92.00
Lung	GSM1322682	93.10
Lung	GSM1322683	94.99
Lung	GSM1322684	90.11
Lung	GSM1322685	92.95

4 Future Scope

The expansion of training set is needed to improve prediction accuracies. Also increasing cancer types (breast, brain, liver etc.) can make this program usable for different groups [11]. Enhancement of the current pipeline is needed in terms of incorporation of other machine learning algorithms [12]. The current pipeline seems applicable in future to predict unknown genes involved in different cancer types using machine learning tools.

Acknowledgements. Support from the Georgia State University Information Technology Department (GSU IT) for server space is gratefully acknowledged.

Author's Contributions. SP conceived, designed the study and critically revised the manuscript. SP developed, tested the software and also did the setup of SQL database and server space.

Funding. No external funding for developing this suite has been utilized.

Competing Interests. The authors declare that they have no competing interests.

References

1. Quackenbush, J.: Microarray data normalization and transformation. Nat. Genet. 496–501 (2002). https://doi.org/10.1038/ng1032
2. OpenCV. Introduction to Support Vector Machine (2014). http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html
3. Ben-Hur, A., Horn, D., Siegelmann, H., Vapnik, V.: Support vector clustering. J. Mach. Learn. Res. **2**, 125–137 (2001)
4. Gautier, L., Cope, L., Bolstad, B.M., Irizarry, R.A.: Affy - an R package for the analysis of Affymetrix GeneChip data at the probe level. Bioinformatics (2003)
5. Smola, A.J., Schölkopf, B.: A Tutorial on support vector regression. Stat. Comput. **14**(3), 199–222 (2004)
6. Scikit-learn developers. Support Vector Machines (2014). http://scikit-learn.org/stable/modules/svm.html
7. Pawar, S., Ashraf, M., Mujawar, S., Mishra, R., Lahiri, C.: In silico identification of the indispensable quorum sensing proteins of multidrug resistant Proteus mirabilis. Front. Cell. Infect. Micro-Biol. **8**, 269 (2018)
8. Ashraf, M., et al.: A side-effect free method for identifying cancer drug targets. Sci. Rep. (2018)
9. Lahiri, C., Pawar, S., Sabarinathan, R., Ashraf, M.I., Chand, Y., Chakravortty, D.: Interactome analyses of Salmonella pathogenicity islands reveal SicA indispensable for virulence. J. Theor. Biol. **363**, 188–197 (2014)
10. Lahiri, C., Shrikant, P., Sabarinathan, R., Ashraf, M., Chakravortty, D.: Identifying indispensable proteins of the type III secretion systems of Salmonella enterica serovar Typhimurium strain LT2. BMC Bioinform. **13**(Suppl. 12), SA10 (2012)
11. Pawar, S., Ashraf, M., Mehata, K., Lahiri, C.: Computational identification of indispensable virulence proteins of Salmonella typhi CT18. Curr. Top. Salmonella Salmonellosis (2017)
12. Pawar, S., Davis, C., Rinehart, C.: Statistical analysis of microarray gene expression data from a mouse model of toxoplasmosis. BMC Bioinform. **12**(Suppl. 7), SA19 (2011)