

Advances in Bioinformatics

Advances in Bioinformatics

Editor

Safaai Deris
Mohd. Saberi Mohamad
Afnizanfaizal Abdullah



www.penerbit.utm.my

2008

First Edition 2008
© SAFAAI DERIS, MOHD. SABERI MOHAMAD
& AFNIZANFAIZAL ABDULLAH 2008

Hak cipta terpelihara. Tiada dibenarkan mengeluarkan mana-mana bahagian artikel, ilustrasi, dan isi kandungan buku ini dalam apa jua bentuk dan cara apa jua sama ada dengan cara elektronik, fotokopi, mekanik, atau cara lain sebelum mendapat izin bertulis daripada Timbalan Naib Canselor (Penyelidikan dan Inovasi), Universiti Teknologi Malaysia, 81310 Skudai, Johor Darul Ta'zim, Malaysia. Perundingan tertakluk kepada perkiraan royalti atau honorarium.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical including photocopy, recording, or any information storage and retrieval system, without permission in writing from Universiti Teknologi Malaysia, 81310 Skudai, Johor Darul Ta'zim, Malaysia.

Perpustakaan Negara Malaysia

Cataloguing-in-Publication Data

Advances in bioinformatics / editors Safaai Deris, Mohd. Saberi Mohamad,
Afnizanfaizal Abdullah.
ISBN 978-983-52-0624-5
I. Bioinformatics. I. Safaai Deris. II. Mohd Saberi Mohamad.
III. Afnizanfaizal Abdullah.
572.8

Editor: Safaai Deris dan Rakan-rakan
Pereka Kulit: Mohd Nazir Md. Basri & Mohd Asmawidin Bidin

Ditatur huruf oleh / *Typeset by*
Fakulti Sains Komputer & Sistem Maklumat

Diterbitkan di Malaysia oleh / *Published in Malaysia by*
PENERBIT

UNIVERSITI TEKNOLOGI MALAYSIA
34 – 38, Jln. Kebudayaan 1, Taman Universiti
81300 Skudai,

Johor Darul Ta'zim, MALAYSIA.

(PENERBIT UTM anggota PERSATUAN PENERBIT BUKU MALAYSIA/
MALAYSIAN BOOK PUBLISHERS ASSOCIATION dengan no. keahlian 9101)

Dicetak di Malaysia oleh / *Printed in Malaysia by*

UNIVISION PRESS SDN. BHD
Lot. 47 & 48, Jalan SR 1/9, Seksyen 9,
Jalan Serdang Raya, Taman Serdang Raya,
43300 Seri Kembangan,
Selangor Darul Ehsan, MALAYSIA.

TABLE OF CONTENTS

<i>Preface</i>		viii
Chapter 1	Predicting Protein-Protein Interactions From Sequence Features Using Support Vector Machines Hany Alashwal , Safaai Deris and Razib M. Othman	1
Chapter 2	Predicting Protein-Protein Interactions From Sequence Features Using Support Vector Machines Hany Alashwal , Safaai Deris and Razib M. Othman	20
Chapter 3	A Multi-Objective Approach To Selects Genes For Cancer Classification Mohd Saberi Mohamad, Sigeru Omatu, Safaai Deris and Michifumi Yoshioka	40
Chapter 4	A Combination Of Hybrid Methods To Select Informative Genes From Mixed-Lineage Leukemia Cancers Mohd Saberi Mohamad, Sigeru Omatu, Safaai Deris and Michifumi Yoshioka	51

Chapter 5	Selecting Informative Genes From Genomics Data Using A Cyclic Approach Mohd Saberi Mohamad, Sigeru Omatu, Safaai Deris and Michifumi Yoshioka	65
Chapter 6	Selecting A Smaller Subset Of Genes For Leukemia Cancer Classification Using A Two-Stage Method Mohd Saberi Mohamad, Sigeru Omatu, Safaai Deris and Michifumi Yoshioka	89
Chapter 7	A Three-Stage Method: Selection Of Informative Genes Based On Mixed-Lineage Leukemia Cancer Data Mohd Saberi Mohamad, Sigeru Omatu, Safaai Deris and Michifumi Yoshioka	105
Chapter 8	An Iterative GA-Based Approach: Gene Selection And Classification Of Lung Cancer Data Mohd Saberi Mohamad, Sigeru Omatu, Safaai Deris and Michifumi Yoshioka	121
Chapter 9	An Improved Binary Particle Swarm Optimization Algorithm For Genes Selection And Classification Of Colon Cancer Data Mohd Saberi Mohamad, Sigeru Omatu, Safaai Deris and Michifumi Yoshioka	132

Chapter 10	Text Modeling Approach For Protein 3D-Structure Similarity Measurement	
	Jafar Razmara	
	Safaai Deris	147
Chapter 11	DNA-Chips Data Processing And Analysis	
	Edin Tankovics	
	Ito Wasito	170
Index		193

PREFACE

In biological and medical research areas, the first wave of computational analysis had focused on sequence analysis. However, this role has been dramatically improved since the increasing needs of solving more complex problems in biology especially during this post-genomics era. Advances in high-throughput experiments which had producing massive biological data such as microarray gene expression and protein-protein interaction provide a lot of information about life which mostly are not been fully understand yet. Therefore, computational approach has become completely necessary in experimental designs, results processing and interpretation in order to discover more information in the biology data.

The used of huge size databases of biological information are insufficient for the researchers to discover the mysteries in biology. Thus, computational tools for identifying genes and proteins homologous similarities, classifying cancerous genes and predicting protein structures and functions have become major component in the research process which is essential in understanding the biological behaviors from the molecular to genome level. Furthermore, the improvement of searching and processing algorithms with in advance of artificial intelligence and machine learning have truly show high potential in giving the researchers a lot of further opportunities.

By learning the theory from the data autonomously through inference process, the large amount of data with high possibility of unintended noise and insignificant information can be processed

with more reliable methods. In addition, the used of artificial intelligence and machine learning can also give advantages in intensive computation and progressive speed. Hence, we present this book as a compilation of recent researches and findings to solve diverse biological issues using artificial intelligence and machine learning techniques.

Safaai Deris

Mohd Saberi Mohamad

Afnizanfaizal Abdullah

Faculty of Computer Science and Information Systems

Universiti Teknologi Malaysia

2008

1

PREDICTING PROTEIN-PROTEIN INTERACTIONS FROM SEQUENCE FEATURES USING SUPPORT VECTOR MACHINES

Hany Alashwal
Safaai Deris
Razib M. Othman

1.1 INTRODUCTION

Identifying protein-protein interactions represents a crucial step in understanding proteins functions. This is due to the fact that proteins work in the context of many other proteins and rarely work in isolation. However, the available interactions data that have been identified by high-throughput technologies like the yeast two-hybrid system are known to yield many false positives. As a result, methods for computational prediction of protein-protein interactions based on sequence information are becoming increasingly important.

Over the past few years, several computational approaches to predict protein-protein interaction have been proposed. Some of the earliest techniques were based on the similarity of expression profiles to predict interacting proteins (Marcotte *et al.*, 1999), coordination of occurrence of gene products in genomes, description of similarity of phylogenetic profiles (Pellegrini *et al.*, 1999) or trees (Pazos and Valencia, 2001), and studying the

patterns of domain fusion (Enright *et al.*, 1999). However, it has been noted that these methods predict protein–protein interactions in a general sense, meaning joint involvement in a certain biological process, and not necessarily actual physical interaction (Eisenberg *et al.*, 2000).

These methods which are based on genomic information are not universal because the accuracy and reliability of these methods depend on information of protein homology or interaction marks of the protein partners. For instance, computational methods such as phylogenetic profiles, predict protein-protein interactions by counting for the pattern of the presence or absence of a given gene in a set of genomes (Marcotte *et al.*, 2000; Craig and Liao, 2007). The main limitation of these approaches is that they can be applied only to completely sequenced genomes, which is the precondition to rule out the absence of a given gene. Similarly, they cannot be used with the essential proteins that are common to most organisms (Shen *et al.*, 2007).

Consequently, predicting protein-protein interactions based only on protein sequence features has a significant importance for computational methods. The advantage of such a method is that it is much more universal. This can be done by developing computation methods that predict protein-protein interactions by associating experimental data on interacting proteins with annotated features of protein sequences using machine learning approaches, such as support vector machines (SVM) (Bock and Gough, 2001; Chung *et al.*, 2004) and data mining techniques, such as association rule mining (Oyama *et al.*, 2002).

The most common sequence feature used for this purpose is the protein domains structure. The motivation for this choice is that molecular interactions are typically mediated by a great variety of interacting domains (Pawson and Nash, 2003). It is thus logical to assume that the patterns of domain occurrence in interacting proteins provide useful information for training PPI prediction methods.

A recent approach based on domain-domain interactions information has been presented in (Gomez *et al.*, 2003). They

developed a probabilistic model to predict protein interactions in the context of regulatory networks. Using the Database of Interacting Proteins (DIP) (Xenarios *et al.*, 2002), as the standard of truth and the Protein Families Database (PFAM) domains as sequence features, the authors built a probabilistic network of yeast interactions and reported an ROC score of 0.818.

Another sequence feature that has been used to computationally predict protein-protein interactions is the hydrophobicity properties of the amino acid residues. Chung *et al.*, (2004) used SVM learning system to recognize and predict protein-protein interactions in the yeast *Saccharomyces cerevisiae*. They selected only the hydrophobicity properties as sequence feature to represent the amino acid sequence of interacting proteins.

Therefore, in this research we proposed a better and more realistic method to construct the negative interaction set. Then we compared the use of domain structure and hydrophobicity properties as the protein features for the learning system. The choice of these two features is motivated by the above discussed literature.

1.2 FEATURE SELECTION AND REPRESENTATION

In order to compare two protein sequence features for the prediction of protein-protein interactions, we applied the same process on both features, as shown in Figure 1. This process starts by generating a dataset of interacting and non-interacting proteins pairs. For the interacting pair, it is simply obtained from the Database of Interacting Protein (DIP).

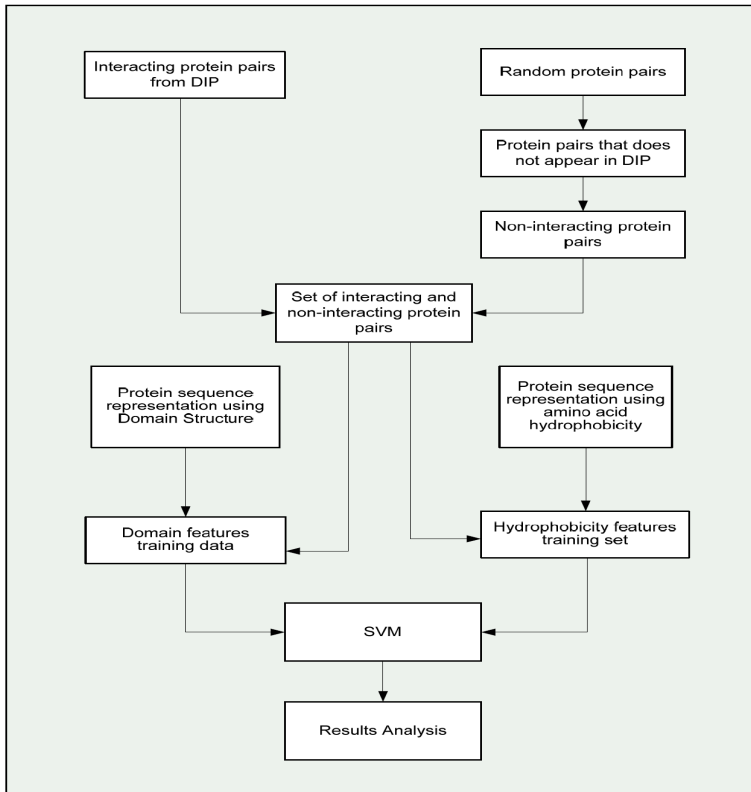


Figure 1 The framework of comparing protein sequence features.

However, there is no dataset of experimentally identified non-interacting proteins. Therefore we use a random method to generate proteins pairs, and then delete all pairs that appear in the DIP. This is acceptable for the purposes of comparing the feature representation since the resulting inaccuracy will be approximately uniform with respect to each feature representation. The Support Vector Machines have been used as the learning system. It has been trained to distinguish between interacting and non-interacting protein pairs using domain and hydrophobicity training sets.

The construction of an appropriate feature space that describes the training data is essential for any supervised machine learning system. In the context of protein-protein interactions, it is believed that the likelihood of two proteins to interact with each other is associated with their structural domain composition (Kim *et al.*, 2003; Pawson and Nash, 2003; Ng *et al.*, 2003). It is also assumed that the hydrophobic effects drive protein-protein interactions (Chung *et al.*, 2004; Uetz and Vollert, 2005). For these reasons, this study investigates the applicability of the domain structure and hydrophobicity properties as protein features to facilitate the prediction of protein-protein interactions using the support vector machines.

The domain data was retrieved from the database of protein families (PFAM) database. PFAM is a reliable collection of multiple sequence alignments of protein families and profile hidden Markov models (Bateman *et al.*, 2004). The current version 10.0 contains 6190 fully annotated PFAM-A families. PFAM-B provides additional PRODOM-generated alignments of sequence clusters in SWISSPROT and TrEMBL that are not modeled in PFAM-A.

When the domain information is used, the dimension size of the feature vector becomes the number of domains appeared in all the yeast proteins. The feature vector for each protein was thus formulated as:

$$x(p) = \{d_1, d_2, \dots, d_i, \dots, d_n\} \quad (\text{Eq. 1})$$

where $d_i = m$ when the protein p has m pieces of domain d_i , and $d_i = 0$ otherwise.

This formula allows the effect of multiple domains to be taken into account. Another representation is by using domain scores that is calculated by PFAM. In this case d_i can be calculated as following:

$$d_i = \sum_{j=1}^k S_{i,j} \quad (\text{Eq. 2})$$

where $S_{i,j}$ is the score of the domain i in the location j , and k is the number of the occurrence of domain i in the protein p . In order to scale the feature value to the interval $[-1,1]$, we use the following formula.

$$d_i = \sum_{j=1}^k (M - (\ln(S_{i,j} + 0.1))) \quad (\text{Eq. 3})$$

where M is the largest number of domain appearance in a protein. For example if domain D appears six times in protein P and no other domains appears more than six times in any proteins, then in this case $M = 6$.

In the same manner, the amino acid hydrophobicity properties can be used to construct the feature vectors for SVM. The amino acids hydrophobicity properties are obtained from (Hopp and Woods, 1981). The hydrophobicity features can be represented in feature vector as:

$$x(p) = \{h_1, h_2, \dots, h_i, \dots, h_r\} \quad (\text{Eq. 4})$$

where r is the number of amino acid in the protein p , $h_i = 1$ when the amino acid is hydrophobic and $h_i = -1$ when the amino acid is hydrophilic. We also consider the case where the hydrophobicity scale can be included in the feature vector by replacing the amino acid with its correspondent hydrophobicity value obtained from (Hopp and Woods, 1981).

Using the above described four feature representations, we constructed four training set (domains, domains with scores, hydrophobicity, and hydrophobicity with scale). Each training example is a pair of interacting proteins (positive example) or a pair of proteins known or presumed not to interact (negative example).

1.3 THE SUPPORT VECTOR MACHINES

The support vector machine (SVM) is a binary classification algorithm. Thus, it is well suited for the task of discriminating between interacting and non-interacting protein pairs. The support vector machine was proposed by Boser *et al.*, (1992). A detailed analysis of SVMs can be found in (Vapnik, 1995 and Shawe-Taylor and Cristianini, 2000). The SVM is based on the idea of constructing the maximal margin hyperplane in the feature space. Suppose we have a set of labeled training data $\{\mathbf{x}_i, y_i\}$, $i = 1, \dots, n$, $y_i \in \{1, -1\}$, $\mathbf{x}_i \in \mathbb{R}^d$, and have the separating hyperplane $(\mathbf{w} \cdot \mathbf{x}) + b = 0$, where feature vector: $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$. In the linear separable case the SVM simply looks for the separating hyperplane that maximizes the margin by minimizing $\|\mathbf{w}\|^2/2$ subject to the following constraint:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \forall i, i = 1, \dots, n \quad (\text{Eq. 5})$$

In the linear non-separable case, the optimal separating hyperplane can be found by introducing slack variables ξ_i , $i = 1, \dots, n$ and user-adjustable parameter C and then minimizing $\|\mathbf{w}\|^2/2 + C \sum_i \xi_i$, subject to the following constraints:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n \quad (\text{Eq. 6})$$

The dual optimization is solved here by introducing the Lagrange multipliers α_i for the non-separable case. Because linear function classes are not sufficient in many cases, we can substitute $\Phi(x_i)$ for each example x_i and use the kernel function K such that $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$. We thus get the following optimization problem:

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (\text{Eq. 7})$$

subject to $0 \leq \alpha_i \leq C, \quad i = 1, \dots, n. \quad \& \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (\text{Eq. 8})$

1.4 MATERIALS AND IMPLEMENTATIONS

The performance of our technique will be tested on dataset obtained from the database of interacting proteins (DIP) (Xenarios *et al.*, 2002). In the following subsections, we will describe the dataset used in this research as well as the data preparation process.

1.4.1 Data Sets

The DIP database was developed to store and organize information on binary protein–protein interactions that was retrieved from individual research articles. At the time of experiments, DIP contains 4749 proteins involved in 15675 interactions for which there is domain information. DIP also provides a high quality core set of 2609 yeast proteins that are involved in 6355 interactions which have been determined by at least one small-scale experiment or at least two independent experiments and predicted as positive by a scoring system (Deane *et al.*, 2002).

The proteins sequence information is needed in this research in order to elucidate the domain structure of the proteins involved in the interaction and to represent the amino acid hydrophobicity in the feature vectors. The proteins sequences files were obtained for the Saccharomyces Genome Database (SGD) (Hong *et al.*, 2005).

1.4.2 Data Preprocessing

Since proteins domains are highly informative for the protein–protein interaction, we used the domain structure of a protein as the main feature of the sequence. We focused on domain data retrieved from the PFAM database which is a reliable collection of multiple sequence alignments of protein families and profile hidden Markov models (Bateman *et al.*, 2004). In order to elucidate the PFAM domain structure in the yeast proteins, we first obtain all sequences of yeast proteins from SGD. Given that sequence file, we then run InterProScan (Mulder *et al.*, 2003) to examine which PFAM domains appear in each protein. We used the stand-alone version of InterProScan.

Protein ID	PFAM Domain 1	PFAM Domain 2	PFAM Domain 3	PFAM Domain 4	PFAM Domain 5
YBL085W	PF00018	PF00169	PF07647	PF07653	
YBL087C	PF00238				
YBL088C	PF00454	PF02259	PF02260		
YBL089W	PF01490				
YBL091C	PF00557				
YBL091C-A	PF00635				
YBL092W	PF01655				
YBL098W	PF01360				
YBL099W	PF00006	PF00306	PF02874		
YBL101W-A	PF01021				
YBL101W-B	PF01021	PF00665			
YBL103C	PF00010				
YBL105C	PF00168	PF00069	PF02185	PF00433	PF00130
YBL111C	PF00270				
YBRO01C	PF01204	PF07492			

Figure 2 Part of the protein domains structure for the yeast

From the output file of InterProScan, we list up all PFAM domains that appear in yeast proteins and index them. The order of this list is not important as long we keep it through the whole procedure. The number of all domains listed and indexed in this way is considered the dimension size of the feature vector, and the index of each PFAM domain within the list now indicates one of the elements in a feature vector. Figure 2 shows an example of protein domains that appears in yeast genome. The first column represents a protein whereas the following columns represent the domains that appear in the protein.

The next step is to construct a feature vector for each protein. For example, if a protein has domain A and B which happened to be indexed 12 and 56 respectively in the above step, then we assign "1" to the 12th and 56th elements in the feature vector, and "0" to all the other elements. Next we focus on the protein pair to be used for SVM training and testing. The assembling of feature vector for each protein pair can be done by

concatenating the feature vectors of proteins constructed in the previous step. When hydrophobicity is used, each amino acid will be replaced by 1 if it is hydrophobic and 0 if it is hydrophilic. Two separate training sets for domain and hydrophobicity features have been constructed. The hydrophobicity scale information can be incorporated in the feature vector by replacing the amino acid with its correspondent hydrophobicity scale value obtained from (Hopp and Woods, 1981). Figure 3 shows part of the final file where the feature vectors are in SVM format.

1.5 RESULTS AND DISCUSSION

We developed programs using Perl for parsing the DIP databases, control of randomization and sampling of records and sequences, and replacing amino acid sequences of interacting proteins with its corresponding feature. To make a positive interaction set, we represent an interaction pair by concatenating feature vectors of each proteins pair that are listed in the DIP-CORE as interacting proteins. For the domain feature we include only the proteins that have structure domains. The resulting positive set for domain feature contains 1879 protein pairs. But when using hydrophobicity feature, all protein in DIP-CORE were included which yielded 3002 protein pairs.

Constructing a negative interaction set is not an easy task. This is due to the fact that there are no experimental data in which protein pairs have confirmed to be non-interacting pairs. As a result, using a random approach to construct the negative data set is an avoidable at this moment. Furthermore, for the purposes of comparing prediction algorithms or feature representation, the resulting inaccuracy will be approximately uniform with respect to each computational method or feature representation. For these reasons, the negative interaction set was constructed by generating random protein pairs. Then, all protein pairs that exist in DIP were eliminated.

This random approach can generate as many as 20202318 potentially negative candidates. Hence, the number of positive protein pairs is quite small compared to that of potentially negative pairs. The excessive potentially negative examples in the training set may lead to yield many false negatives because many of the positive examples are ambiguously discriminative from the negative examples in the feature space. For this reason, a negative interaction set was constructed containing the same number of protein pairs as for the positive interaction set for domain and hydrophobicity features.

In this study, we used the LIBSVM software (Chang and Lim, 2001) as a classification tool. The standard radial basis function (RBF) as available in LIBSVM was selected as a kernel function. Different values of γ for the kernel $K(x, y) = \exp(-\gamma \|x-y\|^2)$, $\gamma > 0$ were systematically tested to optimize the balance between sensitivity and specificity of the prediction. Ten-fold cross-validation was used to obtain the training accuracy. The entire set of training pairs was split into 10 folds so that each fold contained approximately equal number of positive and negative pairs. Each trial involved selecting one fold as a test set, utilizing the remaining nine folds for training our model, and then applying the trained model to the test set. Then the cross-validation accuracy is calculated in each run as the number of corrected prediction divided by the total number of data $((TP+TN)/(TP+FP+TN+FP))$. Then the average is calculated for the 10 folds.

The receiver operating characteristic (ROC) is also used to evaluate the results of our experiments. It is a graphical plot of the sensitivity (fraction of true positives - TP) vs. 1-specificity (the fraction of false positives - FP) for a binary classifier system as its discrimination threshold is varied. The sensitivity can be defined as: $TP / (TP + FN)$ where TP and FN stand for true positive and false negative, respectively. The specificity can be defined as: $TN /$

Table 1 The overall performance of SVM

Feature	Accuracy	ROC score	Running time
Domain	79.4372 %	0.8480	34 seconds
Domain Scores	76.397 %	0.8190	38 seconds
Hydrophobicity	78.6214 %	0.8159	20,571 seconds (5.7 hours)
Hydrophobicity Scales	79.1375 %	0.7716	34,602 seconds (9.6 hours)

(TN + FP) where TN and FP stand for true negative and false positive, respectively. The area under the ROC curve is called ROC score.

The results of our experiments are summarized in Table 1. All experiments reported in this work, run in Redhat Enterprise Linux AS release 3.2 on 1.8 GHz SMP CPUs with 2 GB of memory.

When only domain structure was considered as the protein feature without information on domain appearance score, the cross-validation accuracy and ROC score were respectively 79.4372% and 0.8480. When domain scores were included the cross-validation accuracy and ROC score were decreased to 76.397% and 0.8190 respectively. These results indicate that it is not important to include the domains score information to the feature representation of the protein pairs. It is informative enough to consider only the existence of domains structure in the protein pairs. It is important here to note that the performance of the prediction algorithm is far better than an absolute random approach which has ROC score of 0.5. This indicates that the difference between interacting and non-interacting protein pairs can be learned from the available data.

In the case of hydrophobicity dataset, the cross-validation and ROC score were respectively 78.6214% and 0.8159. We can see from these results that both domain dataset and hydrophobicity dataset have little difference in terms of cross-validation accuracy. On the other hand, ROC score indicates that domain structure is noticeably better than hydrophobic properties (see Figure 4). Another aspect is the running time for both features. Clearly, when domain structure used, the data set is much smaller than the data set for the hydrophobic properties. Consequently, the running time required for domain structure training data is much less than the running time required for the hydrophobic training data as shown in Table 1.

These results are better and came aligned with the results that have been obtained by (Gomez *et al.*, 2003) who reported ROC score of 0.818. Whereas our predictor achieved ROC score of 0.848 for domains feature dataset. However, Chung *et al.* (2004) reported accuracy of 94% by using hydrophobicity as the protein feature. The reason behind this big difference between our result and their results lies in the approach of constructing the negative interaction dataset. They assign random value to each amino acid in the protein pair sequence. This leads to get new pairs that considered negative interacting pairs and greatly different from the pairs in the positive interaction set. This leads to simplify the learning task and artificially raise classification accuracy for training data. There is no guarantee, however, that the generalized classification accuracy will not degrade if the predictor is presented with new, previously unseen data which are hard to classify. In our work we constructed the negative interactions set by randomly generating non-interacting protein pairs which would be more difficult to distinguish from the positive set than entirely randomizing features values. This makes the learning problem more realistic and ensures that our training accuracy better reflects generalized classification accuracy.

1.6 RESULTS AND DISCUSSION

The prediction approach explained in this chapter generates a binary decision regarding potential protein-protein interactions based on the domain structure or hydrophobicity properties of the interacting proteins. In conclusion the result in this chapter suggests that protein-protein interactions can be predicted from domain structure with reliable accuracy and acceptable running time. Consequently, these results show the possibility of proceeding directly from the automated identification of a cell's gene products to inference of the protein interaction pairs, facilitating protein function and cellular signaling pathway identification.

The most challenging task in this research as discussed in this chapter is to find negative examples of interacting proteins, i.e., to find non-interacting protein pairs. For negative examples of SVM training and testing, we use a randomizing method. But we believe this method is only suitable for comparison of features or algorithms. However, finding proper non-interacting protein pairs is important to ensure that prediction system reflects the real world. In the next chapter, we address the unavailability of non-interaction data by predicting protein-protein interactions as a one-class classification problem.

REFERENCES

- Bateman A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, Khanna, S., Marshall, A., Moxon, S.E., Sonnhammer, L.L., Studholme, D.J., Yeats, C. and Eddy S.R. 2004. The Pfam: Protein Families Database. *Nucleic Acids Research Database Issue*. 32:D138-D141.
- Bock, J.R. and Gough, D.A. 2001. Predicting protein-protein interactions from primary structure. *Bioinformatics*. 17(5):455-60.
- Boser, B.E., Guyon, I.M., and Vapnik, V.N. 1992. A training algorithm for optimal margin classifiers. In Haussler, D.(editor) *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. Pittsburgh, PA, ACM:144-152.
- Chang, C.-C. and Lin, C.-J. 2001. LIBSVM: a library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Chung, Y., Kim, G., Hwang, Y. and Park, H. 2004. Predicting Protein-Protein Interactions from One Feature Using SVM. *In IEA/AIE'04 Conf. Proc.* May 17-20. Ottawa, Canada.
- Craig, R.A. and Liao, L. 2007. Phylogenetic tree information aids supervised learning for predicting protein-protein interaction based on distance matrices. *BMC Bioinformatics*. 8:6.
- Eisenberg, D., Marcotte, E. M., Xenarios, I. and Yeates, T.O. 2000. Protein function in the post-genomic era. *Nature*. 405: 823-826.
- Enright, A.J., Iliopoulos, I., Kyrpides, N.C., and Ouzounis, C.A. 1999. Protein interaction maps for complete genomes based on gene fusion events. *Nature*. 402:86-90.
- Gomez, S. M., Noble, W.S. and Rzhetsky, A. 2003. Learning to predict protein-protein interactions from protein sequences. *Bioinformatics*.19(15):1875-1881.

- Hong, E.L., Balakrishnan, R., Christie, K.R., Costanzo, M.C., Dwight, S.S., Engel, S.R., Fisk, D.G., Hirschman, J.E., Livstone, M.S., Nash, R., Oughtred, R., Park, J., *et al.*, 2005. Saccharomyces Genome Database. <http://www.yeastgenome.org/> (16/2/2006).
- Hopp, T.P. and Woods, K.R. 1981. Predicting of protein antigenic determinants from amino acid sequences. *Proc. Natl Acad. Sci. USA.* 78(6): 3824-3828.
- Kim, W.K., Park, J., and Suh, J.K. 2002. Large scale statistical prediction of protein-protein interaction by potentially interacting domain (PID) pair. *Genome Informatics.* 13:42-50.
- Marcotte, E.M., Pellegrini, M., Ng, H., Rice, D.W., Yeates, T.O., and Eisenberg, D. 1999. Detecting Protein Function and Protein-Protein Interactions from Genome Sequences. *Science.* 285(5428):751-753.
- Marcotte, E.M., Xenarios, I., van Der Blik, A.M. and Eisenberg, D. 2000. Localizing proteins in the cell from their phylogenetic profiles, *Proc. Natl. Acad. Sci. USA* 97:12,115-12,120.
- Mulder, N.J., Apweiler, R., Attwood, T. K., Bairoch, A., Barrell, D., Bateman, A., Binns, D., *et al.*, 2003. The InterPro Database brings increased coverage and new features. *Nucleic Acids Research.* 31:315-318.
- Ng, S., Zhang, Z. and Tan, S. 2003. Integrative approach for computationally inferring protein domain interactions. *Bioinformatics.* 19:923-929.
- Oyama, T., Kitano, K., Satou, K. and Ito, T. 2000. Mining association rules related to protein-protein interactions. *Genome Informatics.* 11:358-359.
- Pawson, T. and Nash, P. 2003. Assembly of cell regulatory systems through protein interaction domains. *Science.* 300:445-452.
- Pawson, T. and Nash, P. 2003. Assembly of cell regulatory systems through protein interaction domains. *Science.* 300:445-452.

- Pazos, F. and Valencia, A. 2001. Similarity of phylogenetic trees as indicator of protein-protein interaction. *Protein Engineering*, 14(9), pp: 609-614.
- Pellegrini, M., Marcotte, E., Thompson, M.J., Eisenberg, D. and Yeates, T.O. 1999. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Proc. Nat. Acad. Sci.* 96:4285-4288.
- Shawe-Taylor, J. and Cristianini, N. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.
- Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., Li, Y., and Jiang, H. 2007. Predicting protein-protein interactions based only on sequences information. *PNAS*. 104:4337-4341.
- Uetz, P. and Vollert, C.S. 2005. Protein-Protein Interactions. *Encyclopedic Reference of Genomics and Proteomics in Molecular Medicine (ERGPM)*, Springer Verlag. 16:1548-1552.
- Vapnik V.N. 1995. *The Nature of Statistical Learning Theory*. Springer.
- Xenarios, I., Salwinski, L., Duan, X.J., Higney, P., Kim, S.M., and Eisenberg, D., 2002. DIP: the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucl. Acids. Res.* 30(1):303- 305.

2

PREDICTING PROTEIN-PROTEIN INTERACTIONS FROM SEQUENCE FEATURES USING SUPPORT VECTOR MACHINES

Hany Alashwal
Safaai Deris
Razib M. Othman

2.1 INTRODUCTION

Identifying protein-protein interactions represents a crucial step in understanding proteins functions. This is due to the fact that proteins work in the context of many other proteins and rarely work in isolation. However, the available interactions data that have been identified by high-throughput technologies like the yeast two-hybrid system are known to yield many false positives. As a result, methods for computational prediction of protein-protein interactions based on sequence information are becoming increasingly important.

Over the past few years, several computational approaches to predict protein-protein interaction have been proposed. Some of the earliest techniques were based on the similarity of expression profiles to predict interacting proteins (Marcotte *et al.*, 1999), coordination of occurrence of gene products in genomes, description of similarity of phylogenetic profiles (Pellegrini *et al.*, 1999) or trees (Pazos and Valencia, 2001), and studying the patterns of domain fusion (Enright *et al.*, 1999). However, it has

been noted that these methods predict protein–protein interactions in a general sense, meaning joint involvement in a certain biological process, and not necessarily actual physical interaction (Eisenberg *et al.*, 2000).

These methods which are based on genomic information are not universal because the accuracy and reliability of these methods depend on information of protein homology or interaction marks of the protein partners. For instance, computational methods such as phylogenetic profiles, predict protein-protein interactions by counting for the pattern of the presence or absence of a given gene in a set of genomes (Marcotte *et al.*, 2000; Craig and Liao, 2007). The main limitation of these approaches is that they can be applied only to completely sequenced genomes, which is the precondition to rule out the absence of a given gene. Similarly, they cannot be used with the essential proteins that are common to most organisms (Shen *et al.*, 2007).

Consequently, predicting protein-protein interactions based only on protein sequence features has a significant importance for computational methods. The advantage of such a method is that it is much more universal. This can be done by developing computation methods that predict protein-protein interactions by associating experimental data on interacting proteins with annotated features of protein sequences using machine learning approaches, such as support vector machines (SVM) (Bock and Gough, 2001; Chung *et al.*, 2004) and data mining techniques, such as association rule mining (Oyama *et al.*, 2002).

The most common sequence feature used for this purpose is the protein domains structure. The motivation for this choice is that molecular interactions are typically mediated by a great variety of interacting domains (Pawson and Nash, 2003). It is thus logical to assume that the patterns of domain occurrence in interacting proteins provide useful information for training PPI prediction methods.

A recent approach based on domain-domain interactions information has been presented in (Gomez *et al.*, 2003). They developed a probabilistic model to predict protein interactions in

the context of regulatory networks. Using the Database of Interacting Proteins (DIP) (Xenarios *et al.*, 2002), as the standard of truth and the Protein Families Database (PFAM) domains as sequence features, the authors built a probabilistic network of yeast interactions and reported an ROC score of 0.818.

Another sequence feature that has been used to computationally predict protein-protein interactions is the hydrophobicity properties of the amino acid residues. Chung *et al.*, (2004) used SVM learning system to recognize and predict protein-protein interactions in the yeast *Saccharomyces cerevisiae*. They selected only the hydrophobicity properties as sequence feature to represent the amino acid sequence of interacting proteins.

Therefore, in this research we proposed a better and more realistic method to construct the negative interaction set. Then we compared the use of domain structure and hydrophobicity properties as the protein features for the learning system. The choice of these two features is motivated by the above discussed literature.

2.2 FEATURE SELECTION AND REPRESENTATION

In order to compare two protein sequence features for the prediction of protein-protein interactions, we applied the same process on both features, as shown in Figure 1. This process starts by generating a dataset of interacting and non-interacting proteins pairs. For the interacting pair, it is simply obtained from the Database of Interacting Protein (DIP).

However, there is no dataset of experimentally identified non-interacting proteins. Therefore we use a random method to generate proteins pairs, and then delete all pairs that appear in the DIP. This is acceptable for the purposes of comparing the feature

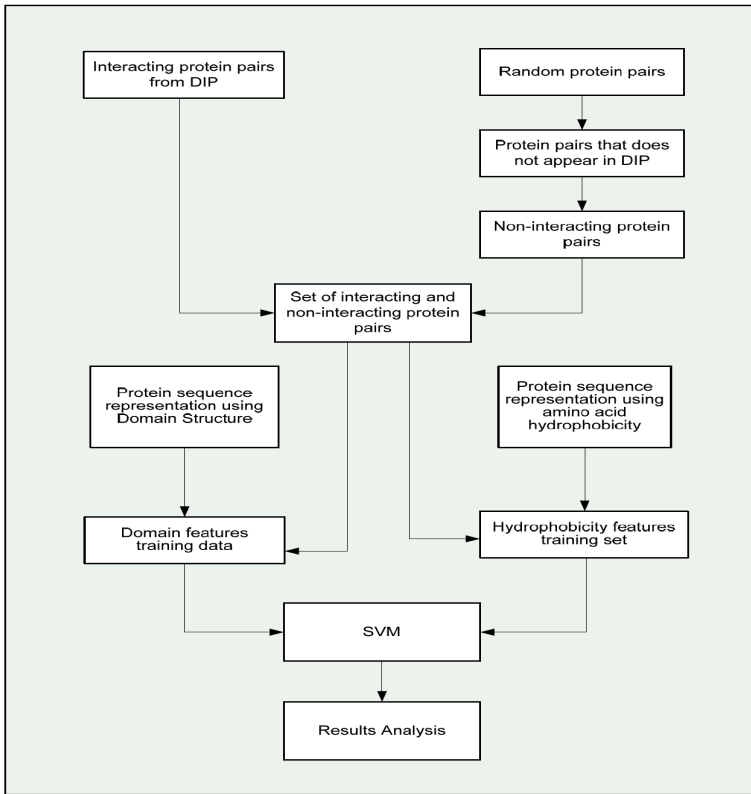


Figure 1 The framework of comparing protein sequence features.

representation since the resulting inaccuracy will be approximately uniform with respect to each feature representation. The Support Vector Machines have been used as the learning system. It has been trained to distinguish between interacting and non-interacting protein pairs using domain and hydrophobicity training sets.

The construction of an appropriate feature space that describes the training data is essential for any supervised machine learning system. In the context of protein-protein interactions, it is believed that the likelihood of two proteins to interact with each

other is associated with their structural domain composition (Kim *et al.*, 2003; Pawson and Nash, 2003; Ng *et al.*, 2003). It is also assumed that the hydrophobic effects drive protein-protein interactions (Chung *et al.*, 2004; Uetz and Vollert, 2005). For these reasons, this study investigates the applicability of the domain structure and hydrophobicity properties as protein features to facilitate the prediction of protein-protein interactions using the support vector machines.

The domain data was retrieved from the database of protein families (PFAM) database. PFAM is a reliable collection of multiple sequence alignments of protein families and profile hidden Markov models (Bateman *et al.*, 2004). The current version 10.0 contains 6190 fully annotated PFAM-A families. PFAM-B provides additional PRODOM-generated alignments of sequence clusters in SWISSPROT and TrEMBL that are not modeled in PFAM-A.

When the domain information is used, the dimension size of the feature vector becomes the number of domains appeared in all the yeast proteins. The feature vector for each protein was thus formulated as:

$$x(p) = \{d_1, d_2, \dots, d_i, \dots, d_n\} \quad (\text{Eq. 1}) \quad (1)$$

where $d_i = m$ when the protein p has m pieces of domain d_i , and $d_i = 0$ otherwise.

This formula allows the effect of multiple domains to be taken into account. Another representation is by using domain scores that is calculated by PFAM. In this case d_i can be calculated as following:

$$d_i = \sum_{j=1}^k S_{i,j} \quad (\text{Eq. 2}) \quad (2)$$

where $S_{i,j}$ is the score of the domain i in the location j , and k is the number of the occurrence of domain i in the protein p . In order to scale the feature value to the interval $[-1,1]$, we use the following formula.

$$d_i = \sum_{j=1}^k (M - (\ln(S_{i,j} + 0.1))) \quad (\text{Eq. 3}) \quad (3)$$

where M is the largest number of domain appearance in a protein. For example if domain D appears six times in protein P and no other domains appears more than six times in any proteins, then in this case $M=6$.

In the same manner, the amino acid hydrophobicity properties can be used to construct the feature vectors for SVM. The amino acids hydrophobicity properties are obtained from (Hopp and Woods, 1981). The hydrophobicity features can be represented in feature vector as:

$$x(p) = \{h_1, h_2, \dots, h_i, \dots, h_r\} \quad (\text{Eq. 4}) \quad (4)$$

where r is the number of amino acid in the protein p , $h_i = 1$ when the amino acid is hydrophobic and $h_i = -1$ when the amino acid is hydrophilic. We also consider the case where the hydrophobicity scale can be included in the feature vector by replacing the amino

acid with its correspondent hydrophobicity value obtained from (Hopp and Woods, 1981).

Using the above described four feature representations, we constructed four training set (domains, domains with scores, hydrophobicity, and hydrophobicity with scale). Each training example is a pair of interacting proteins (positive example) or a pair of proteins known or presumed not to interact (negative example).

2.3 SUPPORT VECTOR MACHINES

The support vector machine (SVM) is a binary classification algorithm. Thus, it is well suited for the task of discriminating between interacting and non-interacting protein pairs. The support vector machine was proposed by Boser *et al.*, (1992). A detailed analysis of SVMs can be found in (Vapnik, 1995 and Shawe-Taylor and Cristianini, 2000). The SVM is based on the idea of constructing the maximal margin hyperplane in the feature space. Suppose we have a set of labeled training data $\{\mathbf{x}_i, y_i\}$, $i = 1, \dots, n$, $y_i \in \{1, -1\}$, $\mathbf{x}_i \in \mathbb{R}^d$, and have the separating hyperplane $(\mathbf{w} \cdot \mathbf{x}) + b = 0$, where feature vector: $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$. In the linear separable case the SVM simply looks for the separating hyperplane that maximizes the margin by minimizing $\|\mathbf{w}\|^2/2$ subject to the following constraint:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \forall i, i = 1, \dots, n \quad (\text{Eq. 5})$$

In the linear non-separable case, the optimal separating hyperplane can be found by introducing slack variables ξ_i , $i = 1, \dots, n$ and user-adjustable parameter C and then minimizing $\|\mathbf{w}\|^2/2 + C \sum_i \xi_i$, subject to the following constraints:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n \quad (\text{Eq. 6})$$

The dual optimization is solved here by introducing the Lagrange multipliers α_i for the non-separable case. Because linear function classes are not sufficient in many cases, we can substitute $\Phi(x_i)$ for each example x_i and use the kernel function K such that $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$. We thus get the following optimization problem:

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (\text{Eq. 7})$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, n. \quad \& \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (\text{Eq. 8})$$

2.4 MATERIALS AND IMPLEMENTATIONS

The performance of our technique will be tested on dataset obtained from the database of interacting proteins (DIP) (Xenarios *et al.*, 2002). In the following subsections, we will describe the dataset used in this research as well as the data preparation process.

2.4.1 Data Sets

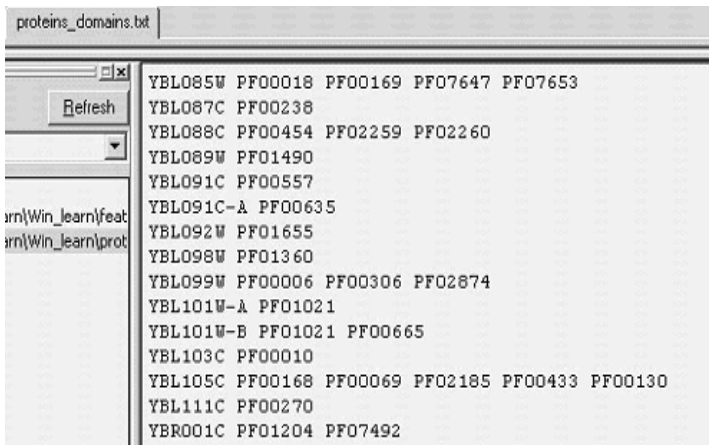
The DIP database was developed to store and organize information on binary protein–protein interactions that was retrieved from individual research articles. At the time of experiments, DIP contains 4749 proteins involved in 15675 interactions for which there is domain information. DIP also provides a high quality core set of 2609 yeast proteins that are involved in 6355 interactions which have been determined by at least one small-scale experiment or at least two independent experiments and predicted as positive by a scoring system (Deane *et al.*, 2002).

The proteins sequence information is needed in this research in order to elucidate the domain structure of the proteins involved in the interaction and to represent the amino acid hydrophobicity in the feature vectors. The proteins sequences files were obtained for the Saccharomyces Genome Database (SGD) (Hong *et al.*, 2005).

2.4.2 Data Preprocessing

Since proteins domains are highly informative for the protein–protein interaction, we used the domain structure of a protein as the main feature of the sequence. We focused on domain data retrieved from the PFAM database which is a reliable collection of multiple sequence alignments of protein families and profile hidden Markov models (Bateman *et al.*, 2004). In order to elucidate the PFAM domain structure in the yeast proteins, we first obtain all sequences of yeast proteins from SGD. Given that sequence file, we then run InterProScan (Mulder *et al.*, 2003) to examine which PFAM domains appear in each protein. We used the stand-alone version of InterProScan.

From the output file of InterProScan, we list up all PFAM domains that appear in yeast proteins and index them. The order of this list is not important as long we keep it through the whole procedure. The number of all domains listed and indexed in this way is considered the dimension size of the feature vector, and the index of each PFAM domain within the list now indicates one of the elements in a feature vector. Figure 2 shows an example of protein domains that appears in yeast genome. The first column represents a protein whereas the following columns represent the domains that appear in the protein.



```
proteins_domains.txt
YBL085W PF00018 PF00169 PF07647 PF07653
YBL087C PF00238
YBL088C PF00454 PF02259 PF02260
YBL089W PF01490
YBL091C PF00557
YBL091C-A PF00635
YBL092W PF01655
YBL098W PF01360
YBL099W PF00006 PF00306 PF02874
YBL101W-A PF01021
YBL101W-B PF01021 PF00665
YBL103C PF00010
YBL105C PF00168 PF00069 PF02185 PF00433 PF00130
YBL111C PF00270
YBR001C PF01204 PF07492
```

Figure 2 Part of the protein domains structure for the yeast genome.

The next step is to construct a feature vector for each protein. For example, if a protein has domain A and B which happened to be indexed 12 and 56 respectively in the above step, then we assign "1" to the 12th and 56th elements in the feature vector, and "0" to all the other elements. Next we focus on the protein pair to be used for SVM training and testing. The assembling of feature vector for each protein pair can be done by concatenating the feature vectors of proteins constructed in the previous step. When hydrophobicity is used, each amino acid will be replaced by 1 if it is hydrophobic and 0 if it is hydrophilic. Two separate training sets for domain and hydrophobicity features have been constructed. The hydrophobicity scale information can be incorporated in the feature vector by replacing the amino acid with its correspondent hydrophobicity scale value obtained from (Hopp and Woods, 1981). Figure 3 shows part of the final file where the feature vectors are in SVM format.

```
domains_core.txt
+1 229:1 229:1 525:1
+1 229:1 229:1 525:1
+1 229:1 229:1 525:1
+1 160:1 161:1 162:1 160:1 161:1 162:1 479:1 480:1
+1 230:1 27:1
+1 231:1 464:1 54:1
+1 231:1 464:1 54:1
+1 242:1 243:1 445:14 446:1 447:1 448:1 449:1 450:1 451:1 452:1
+1 242:1 243:1 446:1 447:1 450:1 451:1 452:1
+1 242:1 243:1 446:1 447:1 450:1 451:1 452:1
+1 149:3 1448:1 858:1
+1 149:3 1448:1 858:1
+1 27:1 54:1
+1 27:1 464:1 54:1
+1 27:1 27:1
+1 27:1 689:8
+1 27:1 464:1 54:1
+1 27:1 464:1 54:1
```

Figure 3 Part of the training data file

2.5 RESULTS AND DISCUSSION

We developed programs using Perl for parsing the DIP databases, control of randomization and sampling of records and sequences, and replacing amino acid sequences of interacting proteins with its corresponding feature. To make a positive interaction set, we represent an interaction pair by concatenating feature vectors of each proteins pair that are listed in the DIP-CORE as interacting proteins. For the domain feature we include only the proteins that have structure domains. The resulting positive set for domain feature contains 1879 protein pairs. But when using hydrophobicity feature, all protein in DIP-CORE were included which yielded 3002 protein pairs.

Constructing a negative interaction set is not an easy task. This is due to the fact that there are no experimental data in which protein pairs have confirmed to be non-interacting pairs. As a result, using a random approach to construct the negative data set is an avoidable at this moment. Furthermore, for the purposes of comparing prediction algorithms or feature representation, the resulting inaccuracy will be approximately uniform with respect to each computational method or feature representation. For these reasons, the negative interaction set was constructed by generating random protein pairs. Then, all protein pairs that exist in DIP were eliminated.

This random approach can generate as many as 20202318 potentially negative candidates. Hence, the number of positive protein pairs is quite small compared to that of potentially negative pairs. The excessive potentially negative examples in the training set may lead to yield many false negatives because many of the positive examples are ambiguously discriminative from the negative examples in the feature space. For this reason, a negative interaction set was constructed containing the same number of protein pairs as for the positive interaction set for domain and hydrophobicity features.

In this study, we used the LIBSVM software (Chang and

Lim, 2001) as a classification tool. The standard radial basis function (RBF) as available in LIBSVM was selected as a kernel function. Different values of γ for the kernel $K(x, y) = \exp(-\gamma \|x - y\|^2)$, $\gamma > 0$ were systematically tested to optimize the balance between sensitivity and specificity of the prediction. Ten-fold cross-validation was used to obtain the training accuracy. The entire set of training pairs was split into 10 folds so that each fold contained approximately equal number of positive and negative pairs. Each trial involved selecting one fold as a test set, utilizing the remaining nine folds for training our model, and then applying the trained model to the test set. Then the cross-validation accuracy is calculated in each run as the number of corrected prediction divided by the total number of data $((TP+TN)/(TP+FP+TN+FP))$. Then the average is calculated for the 10 folds.

The receiver operating characteristic (ROC) is also used to evaluate the results of our experiments. It is a graphical plot of the sensitivity (fraction of true positives - TP) vs. 1-specificity (the fraction of false positives - FP) for a binary classifier system as its discrimination threshold is varied. The sensitivity can be defined as: $TP / (TP + FN)$ where TP and FN stand for true positive and false negative, respectively. The specificity can be defined as: $TN / (TN + FP)$ where TN and FP stand for true negative and false positive, respectively. The area under the ROC curve is called ROC score.

The results of our experiments are summarized in Table 1. All experiments reported in this work, run in Redhat Enterprise Linux AS release 3.2 on 1.8 GHz SMP CPUs with 2 GB of memory.

When only domain structure was considered as the protein feature without information on domain appearance score, the cross-validation accuracy and ROC score were respectively 79.4372% and 0.8480. When domain scores were included the cross-validation accuracy and ROC score were decreased to 76.397% and 0.8190 respectively. These results indicate that it is not important to include the domains score information to the

Table 1 The overall performance of SVM

Feature	Accuracy	ROC score	Running time
Domain	79.4372 %	0.8480	34 seconds
Domain Scores	76.397 %	0.8190	38 seconds
Hydrophobicity	78.6214 %	0.8159	20,571 seconds (5.7 hours)
Hydrophobicity Scales	79.1375 %	0.7716	34,602 seconds (9.6 hours)

feature representation of the protein pairs. It is informative enough to consider only the existence of domains structure in the protein pairs. It is important here to note that the performance of the prediction algorithm is far better than an absolute random approach which has ROC score of 0.5. This indicates that the difference between interacting and non-interacting protein pairs can be learned from the available data.

In the case of hydrophobicity dataset, the cross-validation and ROC score were respectively 78.6214% and 0.8159. We can see from these results that both domain dataset and hydrophobicity dataset have little difference in terms of cross-validation accuracy. On the other hand, ROC score indicates that domain structure is noticeably better than hydrophobic properties (see Figure 4). Another aspect is the running time for both features. Clearly, when domain structure used, the data set is much smaller than the data set for the hydrophobic properties. Consequently, the running time required for domain structure training data is much less than the running time required for the hydrophobic training data as shown in Table 1.

These results are better and came aligned with the results that have been obtained by (Gomez *et al.*, 2003) who reported ROC score of 0.818. Whereas our predictor achieved ROC score of 0.848 for domains feature dataset. However, Chung *et al.* (2004) reported accuracy of 94% by using hydrophobicity as the protein

feature. The reason behind this big difference between our result and their results lies in the approach of constructing the negative interaction dataset. They assign random value to each amino acid in the protein pair sequence. This leads to get new pairs that considered negative interacting pairs and greatly different from the pairs in the positive interaction set. This leads to simplify the learning task and artificially raise classification accuracy for training data. There is no guarantee, however, that the generalized classification accuracy will not degrade if the predictor is presented with new, previously unseen data which are hard to classify. In our work we constructed the negative interactions set by randomly generating non-interacting protein pairs which would be more difficult to distinguish from the positive set than entirely randomizing features values. This makes the learning problem more realistic and ensures that our training accuracy better reflects generalized classification accuracy.

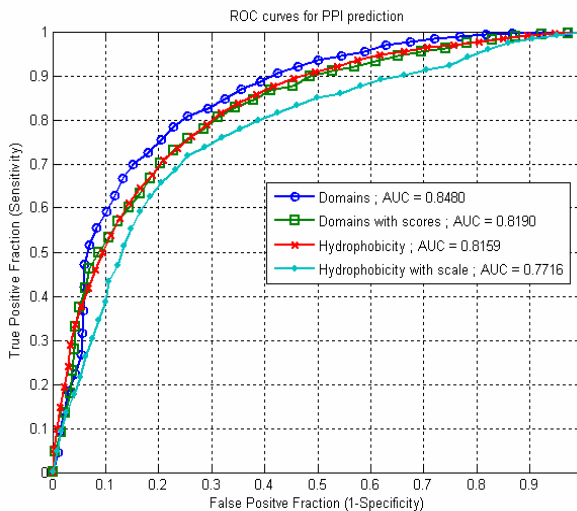


Figure 4 ROC curves and scores for predicting protein-protein interactions.

2.6 SUMMARY

The prediction approach explained in this chapter generates a binary decision regarding potential protein-protein interactions based on the domain structure or hydrophobicity properties of the interacting proteins. In conclusion the result in this chapter suggests that protein-protein interactions can be predicted from domain structure with reliable accuracy and acceptable running time. Consequently, these results show the possibility of proceeding directly from the automated identification of a cell's gene products to inference of the protein interaction pairs, facilitating protein function and cellular signaling pathway identification.

The most challenging task in this research as discussed in this chapter is to find negative examples of interacting proteins, i.e., to find non-interacting protein pairs. For negative examples of SVM training and testing, we use a randomizing method. But we believe this method is only suitable for comparison of features or algorithms. However, finding proper non-interacting protein pairs is important to ensure that prediction system reflects the real world. In the next chapter, we address the unavailability of non-interaction data by predicting protein-protein interactions as a one-class classification problem.

REFERENCES

- Bateman A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, Khanna, S., Marshall, A., Moxon, S.E., Sonnhammer, L.L., Studholme, D.J., Yeats, C. and Eddy S.R. 2004. The Pfam: Protein Families Database. *Nucleic Acids Research Database Issue*. **32**:D138-D141.
- Bock, J.R. and Gough, D.A. 2001. Predicting protein-protein interactions from primary structure. *Bioinformatics*. **17(5)**:455-60.
- Boser, B.E., Guyon, I.M., and Vapnik, V.N. 1992. A training algorithm for optimal margin classifiers. In *Hausler, D.(editor) Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. Pittsburgh, PA, ACM:144-152.
- Chang, C.-C. and Lin, C.-J. 2001. LIBSVM: a library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Chung, Y., Kim, G., Hwang, Y. and Park, H. 2004. Predicting Protein-Protein Interactions from One Feature Using SVM. In *IEA/AIE'04 Conf. Proc.* May 17-20. Ottawa, Canada.
- Craig, R.A. and Liao, L. 2007. Phylogenetic tree information aids supervised learning for predicting protein-protein interaction based on distance matrices. *BMC Bioinformatics*. **8**:6.
- Eisenberg, D., Marcotte, E. M., Xenarios, I. and Yeates, T.O. 2000. Protein function in the post-genomic era. *Nature*. **405**: 823-826.
- Enright, A.J., Iliopoulos, I., Kyrpides, N.C., and Ouzounis, C.A. 1999. Protein interaction maps for complete genomes based on gene fusion events. *Nature*. **402**:86-90.

- Gomez, S. M., Noble, W.S. and Rzhetsky, A. 2003. Learning to predict protein-protein interactions from protein sequences. *Bioinformatics*. **19(15)**:1875-1881.
- Hong, E.L., Balakrishnan, R., Christie, K.R., Costanzo, M.C., Dwight, S.S., Engel, S.R., Fisk, D.G., Hirschman, J.E., Livstone, M.S., Nash, R., Oughtred, R., Park, J., *et al.*, 2005. Saccharomyces Genome Database. <http://www.yeastgenome.org/> (16/2/2006).
- Hopp, T.P. and Woods, K.R. 1981. Predicting of protein antigenic determinants from amino acid sequences. *Proc. Natl Acad. Sci. USA*. **78(6)**: 3824-3828.
- Kim, W.K., Park, J., and Suh, J.K. 2002. Large scale statistical prediction of protein-protein interaction by potentially interacting domain (PID) pair. *Genome Informatics*. **13**:42-50.
- Marcotte, E.M., Pellegrini, M., Ng, H., Rice, D.W., Yeates, T.O., and Eisenberg, D. 1999. Detecting Protein Function and Protein-Protein Interactions from Genome Sequences. *Science*. **285(5428)**:751-753.
- Marcotte, E.M., Xenarios, I., van Der Blik, A.M. and Eisenberg, D. 2000. Localizing proteins in the cell from their phylogenetic profiles, *Proc. Natl. Acad. Sci. USA* **97**:12,115-12,120.
- Mulder, N.J., Apweiler, R., Attwood, T. K., Bairoch, A., Barrell, D., Bateman, A., Binns, D., *et al.*, 2003. The InterPro Database brings increased coverage and new features. *Nucleic Acids Research*. **31**:315-318.
- Ng, S., Zhang, Z. and Tan, S. 2003. Integrative approach for computationally inferring protein domain interactions. *Bioinformatics*. **19**:923-929.
- Oyama, T., Kitano, K., Satou, K. and Ito, T. 2000. Mining association rules related to protein-protein interactions. *Genome Informatics*. **11**:358-359.

- Pawson, T. and Nash, P. 2003. Assembly of cell regulatory systems through protein interaction domains. *Science*. **300**:445–452.
- Pawson, T. and Nash, P. 2003. Assembly of cell regulatory systems through protein interaction domains. *Science*. **300**:445–452.
- Pazos, F. and Valencia, A. 2001. Similarity of phylogenetic trees as indicator of protein-protein interaction. *Protein Engineering*, **14(9)**, pp: 609-614.
- Pellegrini, M., Marcotte, E., Thompson, M.J., Eisenberg, D. and Yeates, T.O. 1999. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Proc. Nat. Acad. Sci.* **96**:4285-4288.
- Shawe-Taylor, J. and Cristianini, N. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.
- Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., Li, Y., and Jiang, H. 2007. Predicting protein–protein interactions based only on sequences information. *PNAS*. **104**:4337-4341.
- Uetz, P. and Vollert, C.S. 2005. Protein-Protein Interactions. *Encyclopedic Reference of Genomics and Proteomics in Molecular Medicine (ERGPM)*, Springer Verlag. **16**:1548-1552.
- Vapnik V.N. 1995. *The Nature of Statistical Learning Theory*. Springer.
- Xenarios, I., Salwinski, L., Duan, X.J., Higney, P., Kim, S.M., and Eisenberg, D., 2002. DIP: the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucl. Acids. Res.* **30(1)**:303- 305.

3

A MULTI-OBJECTIVE APPROACH TO SELECTS GENES FOR CANCER CLASSIFICATION

Mohd Saberi Mohamad

Sigeru Omatu

Safaai Deris

Michifumi Yoshioka

3.1 INTRODUCTION

Gene expression is a process by which mRNA and eventually protein are synthesised from the DNA template of each gene. Recent advances in microarray technology allow scientists to measure the expression levels of thousands of genes simultaneously and determine whether those genes are active, hyperactive, or silent in normal or cancerous tissues. This technology finally produces gene expression data. Current studies on the molecular level classification of tissue have produced remarkable results and indicated that gene expression data could significantly aid in the development of an efficient cancer classification (Mohamad *et al.*, 2005). However, classification based on the data confronts with more challenges. One of the major challenges is the overwhelming number of genes relative to the number of samples in a data set. Many of the genes are also not relevant to the classification process. Hence, the selection of genes is the key of molecular classification, and should be taken with more attention.

The task of cancer classification using gene expression data is to classify tissue samples into related classes of phenotypes, e.g., cancer versus normal (Mohamad *et al.*, 2007). A gene selection process is used to reduce the number of genes used in classification while maintaining an acceptable classification accuracy. Gene selection methods can be classified into two categories. If gene selection is carried out independently from the classification procedure, the methods belong to the filter approach. Otherwise, it is said to follow a wrapper (hybrid) approach. Most previous works have used the filter approach to select genes since it is computationally more efficient than the hybrid approach. However, the hybrid approach usually provides greater accuracy than the filter approach (Mohamad *et al.*, 2005). The application of hybrid approaches using genetic algorithm (GA) with a classifier has grown in recent years. From the previous works, the GA performed well but only on data that have a number of features that is less than 1,000.

Multi-objective optimisation (MOO) is an optimisation problem that involves multiple objectives or goals. Generally, the objectives may estimate very different aspects of solutions. Being aware that gene selection is a MOO problem in the sense of classification accuracy maximisation, and gene subset size minimisation.

Therefore, this research proposes a multi-objective approach in a hybrid of GA and support vector machine classifier (GASVM) for genes selection and classification of gene expression data. It is known as MOGASVM.

3.2 A MULTI-OBJECTIVE APPROACH IN GA

MOGASVM is developed to improve the performance of GASVM that uses single-objective (Mohamad *et al.*, 2005). All information of GASVM such as flowchart, algorithm, chromosome representation, fitness function, and parameter values are available in Mohamad *et al.* (Mohamad *et al.*, 2005).

In the sense of classification accuracy maximisation and gene subset size minimisation, a gene selection can be viewed as a MOO problem. Formally, each gene subset (a solution) is represented by x (n -dimensional decision vector). It is associated with a vector objective function $f(x)$:

$$f(x) = (f_1(x), f_2(x), \dots, f_m(x)) \quad (\text{Eq. 1})$$

with $x = (x_1, x_2, \dots, x_n) \in X$, where X is the decision space, i.e., the set of all expressible solutions.

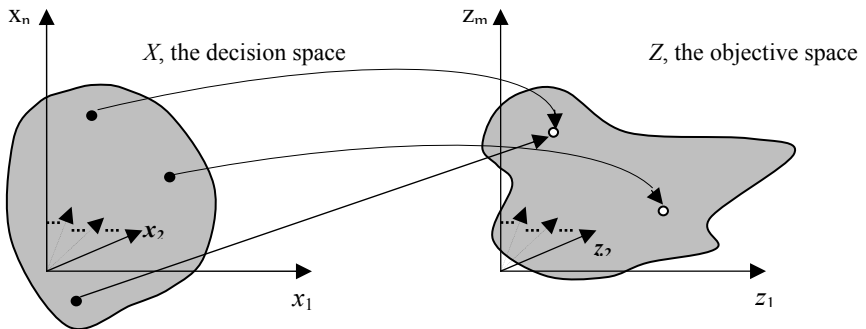


Figure 1 The n -dimensional decision space maps to the m -dimensional objective space

The vector objective function $f(x)$ maps X into \mathfrak{R}^m , where \mathfrak{R} is the objective space and $m \geq 2$ is a number of objectives. f_i is the i th objective. The vector $z = f(x)$ is an objective vector. The image of X in the objective space is the set of all attainable points, z (see Fig. 1). If all objective functions are for maximisation, a subset x is said to dominate than another x^* if and only if:

$x > x^*$ if

$$\forall i \in 1..m, f_i(x) \geq f_i(x^*) \wedge \exists j \in 1..m, f_j(x) > f_j(x^*)$$

A solution (gene subset) is said to be Pareto optimal if it is not dominated by any other solutions in the decision space. A Pareto optimal solution cannot be improved with respect to any objective without worsening at least one other objectives. The set of all feasible non-dominated solutions in X is referred to as the Pareto optimal set, and for a given Pareto optimal set, the corresponding objective function values in the objective space are called as the Pareto front (Handl *et al.*, 2007).

Pareto front in this research is defined as the set of non-dominated gene subsets. MOGASVM is one of promising approaches to find or approximate the Pareto front. The role of this approach is guided with the search towards the Pareto front and preserving the non-dominated solutions as diverse as possible. Therefore, original GASVM is customised to accommodate multi-objective problems by using a specialised fitness function. The ultimate goal of MOGASVM is to identify a non-dominated gene subset Pareto front. This subset (individual) is evaluated by its accuracy on the training data and the number of genes selected in it. These criteria are denoted as f_1 and f_2 separately, and used in the fitness function. Therefore, the fitness of an individual is calculated such equation (4):

$$f_1 = w_1 \times A(x) \quad (\text{Eq. 2})$$

$$f_2 = w_2 \times ((M - R(x)) / M) \quad (\text{Eq. 3})$$

$$\text{fitness}(x) = f_1 + f_2 \quad (\text{Eq. 4})$$

where $A(x) \in [0,1]$ is the leave-one-out-cross-validation (LOOCV) accuracy on the training data using the only expression values of the selected genes in a subset x , $R(x)$ is the number of selected genes in x . M is the total number of genes. w_1 and w_2 are two priority weights corresponding to the importance of accuracy and the number of selected genes, respectively, where $w_1 \in [0.1, 0.9]$ and $w_2 = 1 - w_1$. f_2 is calculated such above in order to support the maximisation function of minimisation of gene subset size. In this paper, the accuracy is more important than the number of selected genes (gene subset size).

Ambroise and McLachlan (2002) have indicated that testing results could be overoptimistic, caused by the “selection bias”, if the test samples were not excluded from the classifier building process in a hybrid approach (Ambroise and McLachlan, 2002). Therefore, the proposed MOGASVM is totally excluded the test samples from the classifier building process in order to avoid the influence of bias.

3.3 EXPERIMENTAL RESULTS

3.3.1 Data Sets

One gene expression data set is used to evaluate the proposed approach, namely the Mixed-Lineage Leukemia (MLL) cancer. The MLL cancer data set is a multi-classes data set. It has three leukemia classes: acute lymphoblastic leukemia (ALL), acute myeloid leukemia (AML), and MLL. The training set contains 57 samples, while the testing set contains 19 samples. There are 12,582 genes in each sample. This data set can be downloaded at <http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi>

3.3.2 Experimental Setup

Three criteria following its important are used to evaluate MOGASVM performances: the test accuracy, the LOOCV accuracy, and the number of selected genes.

The experimental results presented in this section pursue two objectives. The first objective is to show that gene selection using MOGASVM is needed for reducing the number of genes and achieving better classification of gene expression data. The second objective is to show that MOGASVM is better than the original version of GASVM (Mohamad *et al.*, 2005) that use a single-objective approach. To achieve these objectives, several experiments are conducted 10 times using different values of w_1 and w_2 ($w_1 \in [0.1, 0.9]$ and $w_2 = 1 - w_1$). The subset that produces

the highest LOOCV accuracy with the lowest number of selected genes is chosen as the best subset. SVM, GASVM (single-objective), and GASVM-II (Mohamad *et al.*, 2005) are also experimented in this research as a comparison with MOGASVM.

3.3.3 Result Analysis and Discussion

Table 1 displays results of the experiments for the MLL cancer data set using different values of w_1 and w_2 . A value of the form $x \pm y$ represents an average value x with a standard deviation y . Overall, classification accuracy and the number of selected genes sets were fluctuated because of the diversity of the solutions based on adjusted weights (w_1 and w_2). Moreover, multiple objectives search simultaneously in a run and consequently populations tend to converge to the solutions which are superior in one objective, but poor at others. The highest averages of LOOCV and test accuracies were 94.74% and 90.00%, respectively, using $w_1 = 0.7$ and $w_2 = 0.3$.

4,465.2 average genes in a subset were finally selected to obtain the highest accuracies (LOOCV and test) of the data set. This subset was being chosen as the best subset. It is called best-known Pareto front because it is close to the true Pareto front. MOGASVM could obtain the best subsets since it distributed successfully diverse gene subsets over a solution space.

Table 1 Classification accuracies for different gene subsets using MOGASVM (10 runs on average)

Weight		Average for the MLL Data Set		
w_1	w_2	Accuracy (%)		Number of Selected Genes
		LOOCV	Test	
0.1	0.9	94.74 ± 0	88.67 ± 5.49	4,472.1 ± 29.40
0.2	0.8	94.74 ± 0	89.33 ± 4.66	4,470.6 ± 16.54
0.3	0.7	94.74 ± 0	88.67 ± 7.06	4,466.9 ± 21.25
0.4	0.6	94.74 ± 0	89.33 ± 4.66	4,471.4 ± 19.50
0.5	0.5	94.74 ± 0	89.33 ± 5.62	4,465.3 ± 24.60
0.6	0.4	94.74 ± 0	88.67 ± 3.22	4,479.2 ± 21.73
0.7	0.3	94.74 ± 0	90.00 ± 3.51	4,465.2 ± 18.34
0.8	0.2	94.74 ± 0	88.00 ± 6.13	4,479.3 ± 22.24
0.9	0.1	94.74 ± 0	88.00 ± 6.13	4,468.4 ± 16.03

Note: Best result shown in shaded cells.

Table 2 The result of the best subset in 10 runs (using $w_1 = 0.8$ and $w_2 = 0.3$)

Data set	LOOCV (%)	Test (%)	Experiment Number	Number of Selected Genes
MLL	94.74	93.33	7	4,437

Table 2 shows that the best performances (LOOCV and test accuracies) were 94.74% and 93.33%, respectively using 4,437 genes. The best performances have been found in the seventh experiment.

In table 3, the LOOCV accuracy, the test accuracy, and the number of selected genes are written in the parenthesis; the first and second parts are the average result and showcased the best result, respectively. This table shows that the performance of MOGASVM was better than GASVM and SVM in terms of LOOCV accuracy, test accuracy, and the number of selected genes on average and the best results. In general, MOGASVM has reduced about three-quarters of the total number of genes, whereas about a half of GASVM. This is due to the ability of MOGASVM to simultaneously search different regions of a solution space and therefore it is possible to find a diverse set of solution in a high-dimensional space. Moreover, it may also exploit structures of good solutions with respect to different objectives to create new non-dominated solutions in unexplored parts of the Pareto optimal set. This suggests that gene selection using the multi-objective approach is needed for disease classification of gene expression data.

Table 3 The benchmark of MOGASVM with GASVM (single-objective) and SVM

Method	MLL Data Set (Average; The Best)		
	Number of Selected Genes	Accuracy (%)	
		LOOCV	Test
MOGASVM	(4,465.2 ± 18.34; 4,437)	(94.74 ± 0; 94.74)	(90.00 ± 3.51; 93.33)
GASVM (single-objective)	(6,298.8 ± 51.51; 6,224)	(94.74 ± 0; 94.74)	(87.33 ± 2.11; 86.67)
SVM	(12,582 ± 0; 12,582)	(92.98 ± 0; 92.98)	(86.67 ± 0; 86.67)

Note: Best result shown in shaded cells.

3.4 CONCLUSION

MOGASVM has been proposed, developed, and analysed to solve the gene selection problems. By performing experiments, this research found that classification accuracy and the number of selected genes were more fluctuating and not equal when using different values of w_1 and w_2 . This result concludes that there are many irrelevant genes in gene expression data and some of them act negatively on the acquired accuracy by the relevant genes.

Generally, MOGASVM achieved significant the LOOCV accuracy, the test accuracy, and the number of selected genes, and were better than GASVM (single-objective) and SVM since the multi-objective approach in it can find a diverse solution in Pareto optimal set. However, MOGASVM did not achieve the higher accuracy, and the number of selected genes was still higher. MOGASVM can also be extended to other applications such as pattern recognitions, computer visions, and cognitive sciences.

REFERENCES

- Ambroise, C. and McLachlan, G.J. 2002. Selection Bias in Gene Extraction on the Basis of Microarray Gene-expression Data. *Proceedings of the 2002 National Academy of Science of the USA*, Washington, **99**(10), pp. 6562-6566.
- Handl, J., Kell, D.B. and Knowles, J. 2007. Multi-objective Optimisation in Bioinformatics and Computational Biology. *IEEE/ACM Transaction on Computational Biology & Bioinformatics*, **4**(2): 279-292.
- Mohamad, M.S., Deris, S. and Illias, R.M. 2005. A Hybrid of Genetic Algorithm and Support Vector Machine for Features Selection and Classification of Gene Expression Microarray. *International Journal of Computational Intelligence and Applications*, **5**: 91-107.
- Mohamad, M.S., Omatu, S., Deris, S. and Hashim, S.Z.M. 2007. A Model for Gene Selection and Classification of Gene Expression Data. *International Journal of Artificial Life & Robotics*, **11**(2): 219-222.

4

A COMBINATION OF HYBRID METHODS TO SELECT INFORMATIVE GENES FROM MIXED-LINEAGE LEUKEMIA CANCERS

Mohd Saberi Mohamad

Sigeru Omatu

Safaai Deris

Michifumi Yoshioka

4.1 INTRODUCTION

The traditional cancer diagnosis relies on a complex and inexact combination of clinical and histopathological data. This classic approach may fail when dealing with atypical tumours or morphologically indistinguishable tumour subtypes. Advances in the area of microarray-based expression analysis have led to the promise of cancer diagnosis using new molecular-based approaches (Wang *et al.*, 2007). A microarray machine is used to measure the expression levels of thousands of genes simultaneously in a cell mixture, and finally it produces microarray data. The task of cancer classification using microarray data is to classify tissue samples into related classes of phenotypes, e.g., cancer versus normal (Mohamad *et al.*, 2007).

Given N tissue samples and expression of M genes, microarray data are stored in a matrix as shown in Figure 1. Cancer classification using these data poses a major challenge because of the following characteristics:

- $M \gg N$. M is in the range of 2,000-20,000, while N is in the range of 30-200.
- Most genes are not relevant for classifying different tissue types.
- These data have a noisy nature.

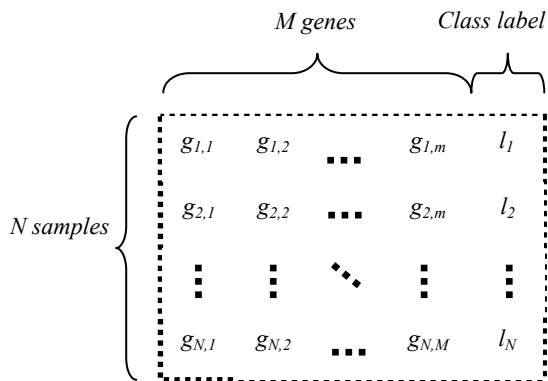


Figure 1 The matrix of microarray data $G_{N \times (M+1)}$. $g_{i,j}$ is a numeric value representing the gene expression level of the j th gene in the i th sample. l_i in the last column is the class label for the i th sample

To overcome the challenge, a gene selection approach is usually used to select a small subset of informative genes that maximises the classifier's ability to classify samples accurately (Mohamad *et al.*, 2007). This approach has several advantages:

- It can maintain or improve classification accuracy.
- It can reduce the dimensionality of data.
- It can remove noisy genes.

Gene selection methods can be classified into two categories. If gene selection is carried out independently from the classification procedure, the method belongs to the filter approach. Otherwise, it is said to follow a hybrid approach. Most previous works have used the filter approach to select genes since it is computationally more efficient than the hybrid approach. However, the hybrid approach usually provides greater accuracy than the filter approach (Mohamad *et al.*, 2005). In this paper, an approach that involves two hybrid methods is proposed to select a small subset of informative genes for cancer classification.

4.2 THE PROPOSED APPROACH

Mohamad *et al.* (Mohamad *et al.*, 2005) have reported that a hybrid of genetic algorithms and support vector machines (GASVM), and an improved GASVM (NewGASVM) have several advantages and disadvantages (Mohamad *et al.*, 2005). In this paper, NewGASVM is called GASVM-II. All information of GASVM and GASVM-II are available in Mohamad *et al.* (Mohamad *et al.*, 2005). The advantage of GASVM is that it can automatically select and optimise a number of genes to produce a gene subset. However, it performs poorly in high-dimensional data. In contrast, GASVM-II performs well in the high-dimensional data. It can also reduce the complexity of search spaces and maybe able to evaluate all possible subsets of genes. Nevertheless, the drawback of GASVM-II is that it selects a number of genes manually to yield a gene subset.

As a result, this paper proposes an approach using two hybrid methods for selecting informative genes. This approach is called as GASVM-II+GASVM. It is developed to improve the performances of GASVM and GASVM-II. Figure 2 shows that the algorithm of GASVM-II+GASVM involving two stages. In the first stage, GASVM-II is applied to manually select genes from the overall microarray data to produce a subset of genes. It is used to reduce the dimensionality of the data, and therefore the complexity of the searches or solution spaces can also be decreased.

Step 1: Select a number of genes and produce initial populations with each chromosome is represented by an integer string.

Step 2: Evaluate each individual (chromosome) in each population using a fitness function.

Step 2.1: Sort integer values in a chromosome.

Step 2.2: Select genes based on position of the integer values in a chromosome (e.g: if integer value=10, then select 10th gene).

Step 2.3: Store the selected genes into a subset.

Step 2.4: $fitness(x) = w_1 \times A(x) + (w_2(M - R(x)) / M)$

Step 3: GA operates on the populations to evolve the best solution (a subset of selected genes) until the final generation.

Step 3.1: Apply a selection strategy and GA operators (crossover and mutation).

Step 3.2: Repeat **Step 2**.

Step 4: Return a subset of genes (the highest fitness).

Step 5: Get the total number of genes from the subset of genes that is produced by Step 4, and produce new initial populations with each chromosome is represented by a bit (0 and 1) string.

Step 6: Evaluate each chromosome in each population using a fitness function.

Step 6.1: Select genes based on bit values in a chromosome (bit 1=select; bit 0=unselect).

Step 6.2: Store the selected genes into a subset.

Step 6.3: $fitness(x) = w_1 \times A(x) + (w_2(M - R(x)) / M)$

Step 7: GA operates on the populations to evolve the best solution (the best subset of genes) until the final generation.

Step 7.1: Apply a selection strategy and GA operators (crossover and mutation).

Step 7.2: Repeat **Step 6**.

Step 8: Return the optimal subset of genes.

Step 9: Classify the optimal subset using an SVM classifier.

Figure 2 The algorithm of GASVM-II+GASVM

In the second stage, GASVM is used to select and optimise a small subset of informative genes from the subset that is produced by the first stage. If the size of the subset is small and the combination of genes is not very complex, GASVM can easily find and optimise the subset. GASVM is applied because it can automatically select a number of genes and finally produce an optimised gene subset. This second stage can also remove noisy genes because the first step has reduced the size and complexity of the search spaces.

Therefore, this proposed approach has totally excluded the test samples from the classifier building process in order to avoid the influence of selection bias (Ambroise and MacLachlan, 2002). The fitness of an individual is calculated as follows:

$$fitness(x) = w_1 \times A(x) + (w_2(M - R(x)) / M) \quad (\text{Eq. 1})$$

in which $A(x) \in [0,1]$ is the leave-one-out-cross-validation (LOOCV) accuracy on the training data using the only expression values of the selected genes in a subset x , $R(x)$ is the number of selected genes

in x . M is the total number of genes. w_1 and w_2 are two priority weights corresponding to the importance of accuracy and the number of selected genes, where $w_1 \in [0.1, 0.9]$ and $w_2 = 1 - w_1$. In this paper, the accuracy is more important than the number of selected genes. Hence, w_1 and w_2 are set to 0.7 and 0.3 respectively for the acute myeloid leukemia (MLL) cancer data set. These values are based on experimental results in Mohamad *et al.*'s paper (Mohamad *et al.*, 2009).

4.3 EXPERIMENT RESULTS

4.3.1 Data Sets

The MLL cancer microarray data set is used to evaluate the proposed algorithm. The MLL cancer data set is a multi-classes data set. It has three leukemia classes: acute lymphoblastic leukemia (ALL), acute myeloid leukemia (AML), and MLL. The training set contains 57 samples (20 ALL, 17 MLL, and 20 AML). While the testing set contains 4 ALL, 3 MLL, and 8 AML samples. There are 12,582 genes in each sample. This data set was published by Armstrong *et al.* (Armstrong *et al.*, 2002). It can be downloaded at http://www.broad.mit.edu/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=63.

4.3.2 Experimental Setup

Three criteria following its importance are considered to evaluate the performances of the proposed approach: the test accuracy, the LOOCV accuracy, and the number of selected genes.

The experimental results presented in this section pursue two objectives. The first objective is to show that gene selection using GASVM-II+GASVM is needed for better classification of microarray data. The second objective is to show that it is better than GASVMs (single-objective and multi-objective) and GASVM-II. To achieve these objectives, several experiments are conducted on the proposed approach 10 times on each data set. In the first stage, it is experimented by using different number of pre-selected genes (10, 20, 30,..., 600). Furthermore, in the second stage, GASVM chooses a number of the final selected genes automatically. Lastly, it produces an optimised gene subset that contains the final selected genes. The subset that produces the highest LOOCV accuracy with the possible least number of selected genes is chosen as the best subset. SVM classifier, GASVMs, and GASVM-II are also experimented for comparison with GASVM-II+GASVM.

4.3.3 Result Analysis and Discussion

In this paper, a value of the form $x \pm y$ represents average value x with standard deviation y . Furthermore, #Pre-Selected Genes, #Final Selected Genes, and Run# represent the number of pre-selected genes, the number of the final selected genes, and a run number, respectively.

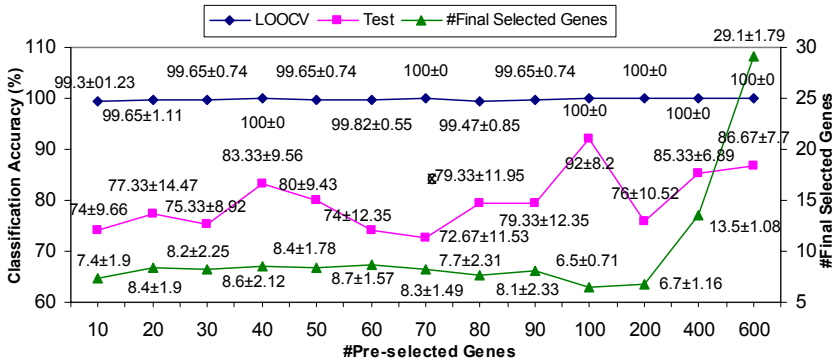


Figure 3 A relation between classification accuracies and the numbers of selected genes (#Pre-selected genes and #Final selected genes) on the MLL data set (10 runs on average)

Figure 3 shows that the highest averages of LOOCV and test accuracies are 100% and 92%, respectively. Only 6.5 average genes were finally selected to obtain the highest average of the accuracies of the data set. All the different numbers of pre-selected genes and the final selected genes have obtained more than 99% LOOCV accuracy, and this has proven that the proposed approach search and select the optimal solution (the best gene subset) in the solution space successfully. However, the test accuracy was much lower than the LOOCV accuracy due to over-fitting problem. This problem happened because of the number of training samples is smaller than the number of test samples, and many expression values of the test samples may be different from those of the training samples.

Table 1 shows that the best performances (LOOCV and test accuracies) of the proposed approach in the best subsets were both 100% using the only six genes. The best performances have been found in the first, second, sixth, and ninth run.

Table 1 The result of the best gene subsets in 10 runs

Data set	#Pre-Selected Genes	LOOCV (%)	Test (%)	#Final Selected Genes	Run#
MLL	100	100	100	6	1,2,6,9

The selected genes in the best gene subsets as founded by GASVM-II+GASVM in Table 1 are shown in Table 2. The probe-set name, gene description, and gene accession number of the selected informative genes are also given. From this finding, the existence of some kinds of relations among the six selected genes of the MLL data set is noted (gene description). Based on graph in Figure 3, different number of selected genes in a subset has produced dissimilar test accuracy. Thus, GASVM-II+GASVM preserves the interactions among the genes that result in the best classification accuracy by using only two genes of the data set. These genes among thousand of genes may be excellent candidate for medical investigations.

Table 2 A list of informative genes in the best gene subsets

Data Set	Run#	Probe-set Name	Gene Accession Number	Gene Description
MLL	1, 2, 6, 9	36873_at	D16532	Human gene for very low density lipoprotein receptor, 5'flanking
	1, 2, 6, 9	40520_g_a t	Y00638	Human mRNA for leukocyte common antigen (T200)
	1, 2, 6, 9	38462_at	U64028	Human NADH:ubiquinone oxidoreductase subunit B13 mRNA, complete cds
	1, 2, 9	31340_at	Y12779	H.sapiens mRNA for enamelysin
	1, 2, 9	1116_at	M28170	Human cell surface protein CD19 (CD19) gene, complete cds
	1, 2	40489_at	D31840	Human DRPLA mRNA for ORF, complete cds
	6	32392_s_at	M57951	Human bilirubin UDP-glucuronosyltransferase isozyme 2 mRNA, complete cds
	6	34950_at	AB018303	Homo sapiens mRNA for KIAA0760 protein, partial cds
	6	1356_at	U18321	Human ionizing radiation resistance conferring protein mRNA, complete cds
	9	32262_at	AL049669	Human gene from PAC 612B18, chromosome 1

The benchmark of the proposed approach comparing with GASVM-II, GASVMs (single-objective and multi-objective), and SVM is summarised in Table 3. The LOOCV accuracy, test accuracy, and number of selected genes are written in the parenthesis; the first and second parts are average result and showcased the best result, respectively. In the table, GASVM-II+GASVM has outperformed GASVM-II, GASVMs, and SVM in terms of the LOOCV accuracy, test accuracy, and number of selected genes on average results and the best results. Generally, GASVM-II was better than GASVMs and SVM. A smaller size gene subset that is produced by the GASVM-II+GASVM results in higher classification accuracy. Hence, it may provide more insights into the molecular classification and diagnosis of cancers. This suggests that gene selection using the proposed approach is needed for cancer classification of microarray data.

Table 3 The benchmark of GASVM-II+GASVM with GASVMs, GASVM-II, and SVM

<i>Method</i>	MLL Data Set (Average; The Best)		
	#Final Selected Genes	Accuracy (%)	
		LOOCV	Test
GASVM-II+GASVM	(6.5 ± 0.71; 6)	(100 ± 0; 100)	(92 ± 8.20; 100)
GASVM-II	(30 ± 0; 30)	(100 ± 0; 100)	(84.67 ± 6.33; 93.33)
GASVM (multi- objective)	(4,465.2 ± 18.34; 4,437)	(94.74 ± 0; 94.74)	(90 ± 3.51; 93.33)
GASVM (single- objective)	(6,298.8 ± 51.51; 6,224)	(94.74 ± 0; 94.74)	(87.33 ± 2.11; 86.67)
SVM classifier	(12,582 ± 0; 12,582)	(92.98 ± 0; 92.98)	(86.67 ± 0; 86.67)

Note: The best result shown in shaded cells.

Table 4 The benchmark of GASVM-II+GASVM with previous works

<i>Author [Reference]</i>	MLL Data Set			
	#Final Genes	Selected	Accuracy (%)	
			LOOCV	Test
Our work	6		100	100
Li <i>et al.</i> 2003		-	-	100
Wang <i>et al.</i> 2005		39	100	-
Wang, 2006		-	98.61	-
Yang <i>et al.</i> 2006		56	97.2	-
Armstrong <i>et al.</i> 2002		100	95	-

Note: The best result shown in shaded cells. ‘-’ means that result is not available.

Table 4 displays benchmark of the best results of this work and previous related works. The best result of the proposed approach was obtained from the best subset in Table 1. Based on LOOCV and test accuracies, it was noted that the best results from this work were equal to the best result from the famous previous work (Li *et al.*, 2003). However, this work only did not display the LOOCV accuracy and the number of selected genes to achieve the accuracy.

4.4 CONCLUSIONS

In this paper, an approach (GASVM-II+GASVM) that involved two hybrid methods has been proposed, developed, and analysed for gene selection and classification. This research found many

combinations of gene subsets that were not equal number of genes have produced the different classification accuracy. This finding suggests that there are many irrelevant and noisy genes in microarray data. In addition, the performances of the GASVM-II+GASVM were superior to the GASVM-II, GASVMs, and SVM. Focusing attention on a smaller subset of genes is useful not only because it produces good classification accuracy, but also since informative genes in this subset may provide insights into the mechanisms responsible for the cancer itself.

It can also be applied in other applications such as robotics, computer intrusion detections, and computer graphics. Even though the proposed approach has classified tumours with higher accuracy, it is still can not avoid the over-fitting problem. A recursive genetic algorithm is currently studied to better select a small subset of genes for cancer classification.

REFERENCES

- Ambroise, C. and McLachlan, G.J. 2002. Selection Bias in Gene Extraction on the Basis of Microarray Gene-expression Data. *Proceedings of the 2002 National Academy of Science of the USA*, Washington, **99**(10), pp. 6562-6566.
- Armstrong, S.A., Staunton, J.E., Silverman, L.B., Pieters, R., Boer, M.L., Minden, M.D., Sallan, S.E., Lander, E.S., Golub, T.R. and Korsmeyer, S.J. 2002. MLL Translocations Specify a Distinct Gene Expression Profile that Distinguishes a Unique Leukemia. *Nature Genetics*, **30**: 41-47.
- Li, J., Liu, H., Ng, S.K. and Wong, L. 2003. Discovery of Significant Rules for Classifying Cancer Diagnosis Data, *Bioinformatics*, **19**: 93-102.

- Mohamad, M.S., Deris, S. and Illias, R.M. 2005. A Hybrid of Genetic Algorithm and Support Vector Machine for Features Selection and Classification of Gene Expression Microarray. *International Journal of Computational Intelligence and Applications*, **5**: 91-107.
- Mohamad, M.S., Omatu, S., Deris, S. and Hashim, S.Z.M. 2007. A Model for Gene Selection and Classification of Gene Expression Data. *International Journal of Artificial Life & Robotics*, **11**(2): 219-222.
- Mohamad, M.S., Omatu, S., Deris, S., Misman, M.F. and Yoshioka, M. 2009. A Multi-objective Strategy in Genetic Algorithms for Gene Selection of Gene Expression Data, *International Journal of Artificial Life & Robotics*, **13**(2).
- Wang, C.W. 2006. New Ensemble Machine Learning Method for Classification and Prediction on Gene Expression Data. *Proceedings of the 28th IEEE Engineering in Medicine and Biology Society*, New York, pp. 3478-3481.
- Wang, L., Chu, F. and Xie, W. 2007. Accurate Cancer Classification Using Expressions of Very Few Genes. *IEEE/ACM Transaction on Computational Biology & Bioinformatics*, **4**(1): 40-53.
- Wang, Y., Makedon, F.S., Ford, J.C. and Pearlman, J. 2005. HykGene: A Hybrid Approach for Selecting Marker Genes For Phenotype Classification using Microarray Gene Expression Data”, *Bioinformatics*, **21**(8): 1530-1537.
- Yang, K., Cai, Z., Li, J. and Lin, G. 2006. A Stable Gene Selection in Microarray Data Analysis”, *BMC Bioinformatics*, **7**: 228-246.

5

SELECTING INFORMATIVE GENES FROM GENOMICS DATA USING A CYCLIC APPROACH

Mohd Saberi Mohamad

Sigeru Omatu

Safaai Deris

Michifumi Yoshioka

5.1 INTRODUCTION

Advances in the area of microarray-based gene expression analysis have led to a promising future of cancer diagnosis using new molecular-based approaches. Microarray technology is used to measure the expression levels of thousands of genes simultaneously, and finally produce gene expression data. A comparison between the gene expression levels of cancerous and normal tissues can also be done. This comparison is useful to select those genes that might anticipate the clinical behaviour of cancers. Thus, there is a need to select informative genes that contribute to a cancerous state. However, the gene selection process poses a major challenge because of the following characteristics of gene expression data: the huge number of genes compared to the small number of samples (higher-dimensional data), irrelevant genes, and noisy data.

To overcome the challenge, a gene selection method is used to select a subset of genes that maximises the classifier's ability to classify samples more accurately. The gene selection

method has several advantages such as improving classification accuracy, reducing the dimensionality of data, and removing irrelevant and noisy genes.

There are two types of gene selection methods (Mohamad *et al.*, 2005): if a gene selection method is carried out independently from a classifier, it belongs to the filter approach; otherwise, it is said to follow a hybrid (wrapper) approach. In the early era of microarray analysis, most previous works have used the filter approach to select genes because it is computationally more efficient than the hybrid approach (Chang *et al.*, 2007; Li *et al.*, 2007). However, the filter approach results in inclusion of irrelevant and noisy genes in a gene subset for the cancer classification. The hybrid approach usually provides greater accuracy than the filter approach since the genes are selected by considering and optimising relations among genes (Mohamad *et al.*, 2007). Until now, several hybrid methods, especially a combination between a genetic algorithm (GA) and a support vector machine (SVM) classifier (GASVM), have been implemented to select informative genes (Huang and Chang, 2007; Li *et al.*, 2005; Li *et al.*, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009; Peng *et al.*, 2003). The drawbacks of the hybrid methods (GASVM-based methods) in the previous works are: 1) intractable to efficiently produce a smaller (near-optimal) subset of informative genes when the total number of genes is too large (higher-dimensional data); 2) the high risk of over-fitting problems.

In order to overcome the limitations of the previous works and solve the problems derived from gene expression data, we propose a cyclic GASVM-based method (C-GASVM). The diagnostic goal is to develop a medical procedure based on the least number of possible genes that needed to detect diseases. Thus, the ultimate goal of this paper is to automatically select a smaller (near-optimal) subset of informative genes that is most relevant for the cancer classification. The proposed method is optimal in the sense that it minimises the number of selected genes and maximises the classification accuracy. To achieve the goal, we

adopt C-GASVM. The proposed method is evaluated on two real gene expression data sets.

The outline of this paper is as follows: Section 2 describes the problems of previous related works, whereas Section 3 discusses the detail of the proposed C-GASVM. In Section 4, gene expression data sets used, experimental setup, and experimental results are described. The conclusion of this paper is provided in Section 5.

5.2 PREVIOUS WORKS

Several hybrid methods, i.e., GASVM-based methods have been proposed for genes selection of gene expression data (Huang and Chang, 2007; Li *et al.*, 2005; Li *et al.*, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009; Peng *et al.*, 2003). Generally, our previous GASVM-based methods performed well in higher-dimensional data, e.g., gene expression data since we proposed a modified chromosome representation and a multi-objective approach (Mohamad *et al.*, 2005; Mohamad *et al.*, 2009). However, the methods yielded inconsistent results when they were run independently.

Li *et al.* (Li *et al.*, 2005) proposed a GASVM-based method for the same purpose. Next, the work of Huang and Chang can simultaneously optimise genes and SVM parameter settings by using a GASVM-based method (Huang and Chang, 2007). An improved GASVM-based method has been recently introduced in Li *et al.* (Li *et al.*, 2008) to produce a small subset of genes. Peng *et al.* (Peng *et al.*, 2003) introduced a recursive feature elimination post-processing step after the step of a GASVM-based method in order to reduce the number of selected genes again (Peng *et al.*, 2003).

Nevertheless, the GASVM-based methods of the previous works

are still intractable to produce a near-optimal subset of genes from higher-dimensional data due to their binary chromosome representation drawback (Huang and Chang, 2007; Li *et al.*, 2005; Li *et al.*, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009; Peng *et al.*, 2003). The total number of gene subsets produced by GASVM-based methods is calculated by $M_c = 2^M - 1$ where M_c is the total number of gene subsets, and M is the total number of genes. Based on this equation, the GASVM-based methods are almost impossible to evaluate all possible subsets of selected genes if M is too many (higher-dimensional data). Although the works of Peng *et al.* and Li *et al.* have implemented a pre-processing step to decrease the dimensionality of data, but it can only reduce a small number of genes, and many genes are still available in the data (Li *et al.*, 2008; Peng *et al.*, 2003). The GASVM-based methods also face with the high risk of over-fitting problems. The over-fitting problem that occurred on hybrid methods (e.g., GASVM-based methods) was also reported in a review paper written by Saeys *et al.* (Saeys *et al.* 2007).

5.3 THE PROPOSED CYCLIC HYBRID METHOD (C-GASVM)

In this paper, we propose C-GASVM for gene selection from gene expression data. C-GASVM is a hybrid method based on GASVM. C-GASVM in our work differs from the GASVM-based methods in the previous works (Huang and Chang, 2007; Li *et al.*, 2005; Li *et al.*, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009; Peng *et al.*, 2003) in one major part. The major difference is that our proposed method involves a cyclic (an iterative) approach, whereas the previous works did not use any a cyclic approach for gene selection. The general procedure of C-GASVM is shown in

Figure 1.

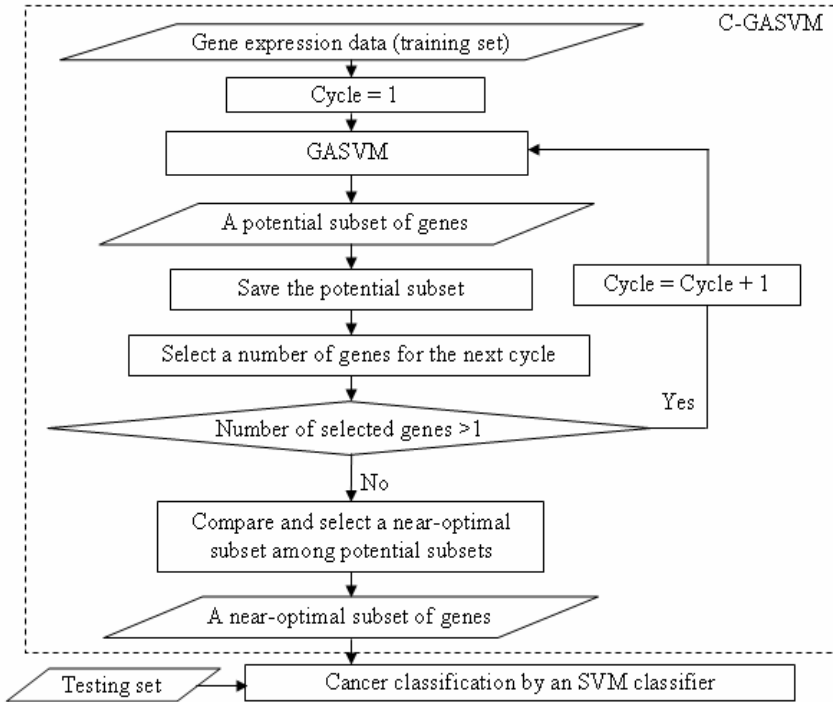


Figure 1 The general procedure of C-GASVM

5.3.1 Pseudo-code for C-GASVM

The detail of C-GASVM is shown in Fig. 2. Basically, C-GASVM repeats the process of GASVM to produce potential subsets and reduce the dimensionality of data iteratively. The description of each step in Fig. 2 is explained as follows:

- Step 1: Starting a cyclic (an iterative) process. It is repeated until the number of selected genes in the potential subset of the cycle c is equal to 1. Every cycle (cycle c) is started here.
- Step 2: Starting GASVM to search and produce a potential subset of selected genes.
- Step 3: End of Step 2.
- Step 4: Producing and saving the potential subset of genes. This potential subset is used for the next cycle (cycle $c+1$) as an input set. The selection of genes in the next cycle (cycle $c+1$) only uses genes in the potential subset that are resulted by the cycle c . Therefore, the dimensionality and complexity of solution spaces can be decreased on a cycle by cycle basis.
- Step 5: Select a number of genes for the next cycle (cycle $c+1$). In each cycle of C-GASVM, a number of selected genes are automatically selected and the dimensionality is automatically reduced. C-GASVM only chooses the

large number of selected genes in each cycle in order to avoid an over-fitting problem. This is reported and proved on the experimental results in Section 4.3.3. The problem could be avoided since the large selection involves many relations among genes, and C-GASVM reduces the number of genes (dimensionality) iteratively.

Step 6: End of Step 5.

Step 7: End of Step 1.

Step 8: A near-optimal subset is selected among the potential subsets based on the highest fitness value (the highest LOOCV accuracy with the smallest number of selected genes).

Step 9: The cyclic (iterative) process (Steps 1-8) results a near-optimal subset of genes. This near-optimal subset is possible to be found due to the dimensionality of data has been iteratively reduced. The subset is then used to construct an SVM classifier, and the constructed SVM is tested by using the test set.

5.3.2 Chromosome Representation for C-GASVM

We use integer chromosome representation in C-GASVM in order to overcome the limitation of the binary chromosome representation in previous related works (Huang and Chang, 2007; Li *et al.*, 2005; Li *et al.*, 2008; Mohamad *et al.*, 2009; Peng *et al.*, 2003). We modify the mechanism of gene selection of C-GASVM based on the representation to efficiently select gene subsets from higher-dimensional data. The modification idea is based on Eq. (1) to reduce the number of gene subsets by fixing the number of selected genes. The fixing process is automatically done by a cyclic process in C-GASVM for each cycle.

$$y = {}_M C_x = \frac{M!}{x!(M-x)!} \quad (\text{Eq. 1})$$

where ${}_M C_x$ is the total number of subsets of selected genes x from the total number of genes M .

Figure 3 shows a graph based on Eq. (1). A maximum number of subsets are reached when the number of selected genes is chosen at $M/2$. Hence, the selection number at $M/2$ or about $M/2$ should be avoided. If the selection uses the number, C-GASVM is impossible to evaluate all subsets due to the huge number of subsets. Conversely, all subsets of genes are possible evaluated if a small or large number of the selected genes are chosen. In this work, C-GASVM only chooses the large number of selected genes in each cycle in order to avoid an over-fitting problem. If the selection chooses the small number, C-GASVM faces with the problem. This is reported and proved by the experimental results in Section 4.3.3.

Therefore, in C-GASVM, the chromosome representation is modified as shown in Fig. 4 which has integer representation. It includes values of integers g^j that indicate which genes are

needed to be selected among the total genes in a data set. For example, if $g^j = 10$, then C-GASVM selects the 10th gene from the data set, and groups it into a subset of genes. The number of selected genes is represented by n_s . The number of g^j in a chromosome is equal to n_s . The binary chromosome representation of GASVM-based methods in the related previous works is encoded with all genes and its size depends entirely on the total number of genes, M (Huang and Chang, 2007; Li *et al.*, 2005; Li *et al.*, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009; Peng *et al.*, 2003). In contrast, the integer chromosome representation in C-GASVM is only encoded with a number of selected genes that is automatically fixed by the cyclic process. Hence, the total number of genes, M does not really affect the size (length) of the chromosome so as to keep its size relatively small. Its size can vary according to M and n_s . The size of chromosomes and the number of selected genes are also same for a similar cycle, but they are different for dissimilar cycles. Finally, a chromosome (a gene subset) is represented as $x = (g^1, g^2, \dots, g^{n_s-1}, g^{n_s})$. For example, the a th chromosome is represented by $x_a = (g_a^1, g_a^2, \dots, g_a^{n_s-1}, g_a^{n_s})$.

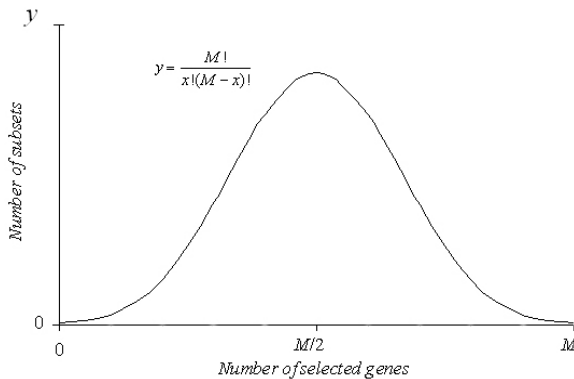


Figure 2 The relation between the number of subsets y and the number of selected genes x from the total number of genes M

g^1	g^2	...	g^{n_s-1}	g^{n_s}
-------	-------	-----	-------------	-----------

Note:

n_s = a number of selected genes from an input set (S_{c-1}), $1 \leq n_s \leq M$.

M = the total number of genes in an input set (S_{c-1}).

g^j = an integer value in a chromosome, $1 \leq g^j \leq M$.

j = the j th gene in a chromosome, $1 \leq j \leq n_s$.

Figure 3 Integer chromosome representation in C-GASVM

```

VARIABLE:
c : cth cycle.  $n_s$  : the number of selected genes.  $x_a$  : ath chromosome.
 $x_a$ .fitness : the fitness of ath chromosome.  $x_a$ .#gene : the number of genes in ath chromosome.
 $S_c$  : a potential subset of genes of cycle c.  $S_c$ .fitness : the fitness value of  $S_c$ .
 $S_c$ .#gene : the number of genes in  $S_c$ . gen : generation. N : the total number of samples
M : the total number of genes. div_gene : the divider for the number of selected genes.

INPUT:
 $G_{N \times (M+1)}$  : microarray data (training set). pop_num : the number of population.
gen_num : the number of generation. cross_rate : the rate of crossover operator.
mut_rate : the rate of mutation operator.

OUTPUT:
 $S_{opt}$  : a near-optimal subset of genes.  $S_{opt}$ .fitness : the fitness value of  $S_{opt}$ .
 $S_{opt}$ .#gene : the number of genes in  $S_{opt}$ .

Begin
gen := 0; c := 1;  $n_s$  := M / div_gene;
 $S_c$  :=  $G_N(M+1)$ ;  $S_c$ .fitness := 0;  $S_c$ .#gene := M;
 $S_{opt}$  := 0;  $S_{opt}$ .fitness := 0;  $S_{opt}$ .#gene := 0;
while ( $S_c$ .#gene > 1) do // Step 1: Starting a cyclic (iterative) process
  for (a = 1; a ≤ pop_num; a++)
     $x_a$  := initialise(int,  $n_s$ ,  $S_c$ );
  end for
  while (gen < gen_num) do // Step 2: Starting GASVM to produce a potential subset of selected genes
    for (a = 1; a ≤ pop_num; a++)
       $SVM(x_a)$ ;
       $x_a$ .fitness :=  $w \times A(x) + (v \times (N - A(x))) / N$ ;
    end for
    selection_method(poulette_wheel, gen);
    crossover(two_point, cross_rate);
    mutation(gaussian, mut_rate);
    gen := gen + 1;
  end while // Step 3: Ending GASVM (Step 2)
  return ( $S_c$ ); // Step 4: Producing and saving the potential subset for the cycle c

  if ( $S_c$ .#gene > 100) then // Step 5: Selecting a number of genes for the next cycle (cycle c+1)
     $n_s$  :=  $S_c$ .#gene / div_gene;
    if ( $n_s$  < 100) then
       $n_s$  := 100;
    end if
  end if
  else if (10 <  $S_c$ .#gene ≤ 100) then
     $n_s$  :=  $S_c$ .#gene - 10;
  end else if
  else if (1 <  $S_c$ .#gene ≤ 10) then
     $n_s$  :=  $S_c$ .#gene - 1;
  end else if
  c := c + 1; gen := 0; // Step 6: Ending the selection process (Step 5)
end while // Step 7: Ending the cyclic process (Step 1)

for (i = 0; i < c; i++) // Step 8: Compare and select an optimal subset among potential subsets
  if ( $S_i$ .fitness >  $S_{opt}$ .fitness) then
     $S_{opt}$  :=  $S_i$ ;  $S_{opt}$ .fitness =  $S_i$ .fitness;  $S_{opt}$ .#gene =  $S_i$ .#gene;
  end if
end for
return ( $S_{opt}$ ); // Step 9: Producing a near-optimal subset of selected genes
End

```

Figure 4 The pseudo-code of C-GASVM

5.3.3 A Fitness Function for C-GASVM

A fitness value of an individual (a gene subset) is calculated as follows:

$$fitness(x) = w_1 \times A(x) + (w_2(M - R(x)) / M) \quad (\text{Eq. 2})$$

in which $A(x) \in [0,1]$ is leave-one-out-cross-validation (LOOCV) accuracy on the training set using the only expression values of the selected genes in a gene subset, x . This accuracy is provided by an SVM classifier. $R(x)$ is the number of selected genes in x . M is the total number of genes for each sample in the training set. w_1 and w_2 are two priority weights corresponding to the importance of accuracy and the number of selected genes, respectively, where $w_1 \in [0.1, 0.9]$ and $w_2 = 1 - w_1$.

5.4 EXPERIMENT

5.4.1 Data Sets

Two real gene expression data sets that contain binary classes and multi-classes are used to evaluate C-GASVM: lung cancer and small round blue cell tumours (SRBCT) cancer data sets. The lung

cancer data set has two classes: malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA). There are 181 samples (31 MPM and 150 ADCA). The training set contains 32 of them (16 MPM and 16 ADCA). The rest of the 149 samples (15 MPM and 134 ADCA) are used for the test set. Each sample is described by 12,533 genes. It can be obtained at <http://chestsurg.org/publications/2002-microarray.aspx>.

The SRBCT cancer data set is a multi-classes data set. It has four classes: ewing family of tumours (EWS), rhabdomyosarcoma (RMS), neuroblastoma (NB), and burkitt lymphomas (BL). The training set contains 63 samples (23 EWS, 20 RMS, 12 NB, and 8 BL), whereas the test set contains 20 samples (6 EWS, 5 RMS, 6 NB, and 3 BL). There are 2,308 genes in each sample.. It can be downloaded at <http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi>.

5.4.2 Experimental Setup

Since the number of training samples in gene expression data is small, the accuracy on the training set is calculated through an LOOCV procedure. For the test accuracy, a classifier is built using all the training samples, and the classes of test samples from the test set are predicted one by one using the built classifier. The test accuracy is estimated by the number of the correctly classified test samples, divided by the number of samples in the test set.

Table 1 contains parameter values for C-GASVM. These values are chosen based on the results of preliminary runs. Three criteria following their importance are considered to evaluate the performances of C-GASVM and other experimental methods: the test accuracy, the LOOCV accuracy, and the number of selected genes. Higher accuracies and a smaller number of selected genes are needed to obtain an excellent performance.

Table 1 Parameter settings for C-GASVM

Parameter	Data Set	SRBCT	Lung
The size of population		50	50
The number of generation		100	100
Replacement rate (Roulette wheel selection)		0.8	0.8
Crossover rate (Two-point)		0.7	0.7
Mutation rate (Gaussian)		0.01	0.01
w_1		0.8	0.7
w_2		0.2	0.3
<i>div_gene</i>		1.33	1.33
<i>Cost</i> for generalisation of an SVM classifier		100	0.7

Experimental results presented in this paper pursue four objectives. The first objective is to show that a gene selection using C-GASVM is needed to produce a smaller (near-optimal) subset of informative genes for better classification accuracy. The second objective is to display a list of informative genes in the best subsets produced by C-GASVM for biological usage. The third objective is to show that C-GASVM is better than other experimental methods such as GASVM (single-objective), multi-objective GASVM (MOGASVM), GASVM version 2 (GASVM-II), and an SVM classifier. The last objective is to compare C-GASVM with other previous works that only used GASVM-based methods. To achieve the four objectives, several experiments are conducted 10 times on each data set using C-GASVM and other experimental methods. Next, an average result of the 10 independent runs is obtained. A near-optimal subset that produces the highest classification accuracies with the possible least number of genes is selected as the best subset.

5.4.3 Experimental Results

5.4.3.1 *LOOCV And Test Accuracies Of Selected Genes With C-GASVM*

Table 2 shows the classification accuracy for each run using C-GASVM on both data sets. Interestingly, all runs have achieved 100% LOOCV accuracy. This has proven that C-GASVM has efficiently selected and produced the near-optimal solution in a solution space. This is due to the fact of its ability to automatically reduce the dimensionality and complexity of the solution space on a cycle by cycle basis. C-GASVM also removes irrelevant and noisy genes in order to yield the higher accuracy. The smaller gene subsets that are produced by the proposed C-GASVM result in the higher classification accuracy.

Practically, the best subset of a data set is firstly chosen and the genes in it are then listed for biological usage. These informative genes among the thousand of genes may be the excellent candidates for clinical and medical investigations. Biologists can save much time since they can directly refer to the genes that have higher possibility to be useful for cancer diagnosis and drug target in the future. The best subset is chosen based on the highest classification accuracy with the smallest number of selected genes. The highest accuracy gives confidence to us for the most accurate classification of cancer types. Moreover, the smallest number of selected genes for cancer classification can reduce the cost in a clinical setting.

Table 2 Classification accuracies for each run using C-GASVM

Run#	Lung Data Set			SRBCT Data Set		
	LOOCV (%)	Test (%)	#Genes	LOOCV (%)	Test (%)	#Genes
1	100	94.63	2	100	90	7
2	100	93.96	5	100	85	6
3	100	94.63	2	100	80	7
4	100	90.60	2	100	85	7
5	100	93.96	5	100	80	7
6	100	98.66	4	100	85	7
7	100	94.63	2	100	85	7
8	100	94.63	2	100	85	7
9	100	90.60	2	100	85	7
10	100	90.60	2	100	85	8
Average ± S.D.	100 ± 0	93.69 ± 2.52	2.80 ± 1.32	100 ± 0	84.5 ± 2.84	8.3 ± 4.14

Note: The result of the best subsets of each data set shown in shaded cells. S.D. denotes the standard deviation, whereas Run# and #Genes represent a run number and a number of selected genes, respectively.

5.4.3.2 A List Of Informative Genes For Biological Usage

Informative genes in the best gene subsets as produced by the proposed C-GASVM and reported in Table 2 are listed in Table 3. Some of these genes are already identified to be highly possible clinical markers for cancer diagnosis by biological researches. the remaining genes may be the excellent candidates for further clinical investigation.

Table 3 The list of informative genes in the best gene subsets

Data Set	Probe-set Name / Image ID	Gene Accession Number / Gene Card Identifier	Gene Description
SRBCT	884867	GC14P102870	eukaryotic translation initiation factor 5 (ETI5)
	868304	GC10M090684	actin, alpha 2, smooth muscle, aorta (ACTA2)
	1323448	GC14P105024	cysteine-rich protein 1 (intestinal) (CRIP1)
	450152	GC09P000263	dedicator of cytokinesis 8 (DOCK8)
	298963	GC14P104957	metastasis associated 1 (MTA1)
	725188	GC02P063727	malate dehydrogenase 1, NAD (soluble) (MDH1)
	823696	GC10P091142	interferon-induced protein 56 (IFIT1)
	295985	GC07M092072	cyclin-dependent kinase 6 (CDK6)
	139705	GC13M072181	similar to RIKEN cDNA 2410129H14
	244652	GC09P130485	SET translocation (myeloid leukemia-associated) (SET)
Lung	32551_at	U03877	EGF-containing fibulin-like extracellular matrix protein 1
	33634_at	AF038007	ATPase, Class I, type 8B, member 1
	35708_at	W27414	Homo sapiens, clone IMAGE:3502329, mRNA, partial cds
	36938_at	U70063	N-acylsphingosine amidohydrolase (acid ceramidase)

5.4.3.3 C-GASVM Versus Other Experimental Methods

The benchmark of C-GASVM in comparison with other experimental methods that have been experimented in this work is summarized in Table 4.

Table 4 The benchmark of C-GASVM with other experimental method

Method	SRBCT Data Set (Average \pm S.D.; The Best)			Lung Data Set (Average \pm S.D.; The Best)		
	#Genes	Accuracy (%)		#Genes	Accuracy (%)	
		LOOCV	Test		LOOCV	Test
C-GASVM	(8.3 \pm 4.14; 20)	(100 \pm 0; 100)	(84.5 \pm 2.84; 90)	(2.80 \pm 1.32; 4)	(100 \pm 0; 100)	(93.69 \pm 2.52; 98.66)
GASVM-II (Mohamad <i>et al.</i> , 2005)	(10 \pm 0; 10)	(99.84 \pm 0.50; 100)	(68 \pm 9.49; 85)	(10 \pm 0; 10)	(100 \pm 0; 100)	(59.33 \pm 29.32; 97.32)
MOGASVM (Mohamad <i>et al.</i> , 2008)	(444.7 \pm 19.09; 429)	(100 \pm 0; 100)	(81.5 \pm 7.47; 85)	(4,418.5 \pm 50.19; 4,433)	(75.31 \pm 0.99; 78.13)	(85.84 \pm 3.97; 93.29)
GASVM (Mohamad <i>et al.</i> , 2005)	(1146 \pm 10.33; 1134)	(98.41 \pm 0; 98.41)	(78.5 \pm 3.38; 85)	(6,267.8 \pm 56.34; 6,342)	75 \pm 0; 75)	(84.77 \pm 2.53; 87.92)
SVM (Mohamad <i>et al.</i> , 2008)	(2,308 \pm 0; 2,308)	(6,298.8 \pm 51.51; 6,224)	(80 \pm 0; 80)	(12,533 \pm 0; 12,533)	(65.63 \pm 0; 65.63)	(85.91 \pm 0; 85.91)

Note: The best result of each data set shown in shaded cells. S.D. denotes the standard deviation, whereas #Genes represents a number of selected genes.

GASVM (single-objective) is developed to implement a single-objective approach in its fitness function, while MOGASVM for multi-objective approach. Binary chromosome representation has been used in these hybrid methods. GASVM-II and C-GASVM are almost same in terms of chromosome representation, algorithm, etc. The difference is that GASVM-II not implements the iterative process in its mechanism. It is developed to prove that an over-fitting problem is happen when the selection using a small number of selected genes, and compare its experimental results with C-GASVM.

GASVM (single-objective) and MOGASVM cannot produce a near-optimal subset of informative genes because they perform poorly in higher-dimensional data due to their chromosome representation drawback. The LOOCV accuracy of GASVM-II is much higher than its test accuracy. These findings prove that GASVM-II causes the over-fitting problem even if it uses a smaller numbers of selected genes. This problem happens since the smaller selections not involve many relations among genes. This method would also be difficult for the usage because it selects a number of genes manually.

On the contrary, C-GASVM selects a large number of genes automatically in each cycle of the iterative process to finally produce a smaller (near-optimal) subset of informative genes. The gap between LOOCV accuracy and test accuracy was also lower. Therefore, C-GASVM is more efficient than other experimental methods since it has produced the higher classification accuracies, smaller number of selected genes, smaller standard deviations, and smaller gap between LOOCV accuracy and test accuracy. However, due to the iterative process, C-GASVM is computationally more extensive than other methods.

5.4.3.4 C-GASVM Versus Related Previous Works

For an objective comparison, we only compare our work with related previous works that used GASVM-based methods in their works (Huang and Chang, 2007; Mohamad *et al.*, 2005). The previous works produced the average results of classification accuracy since they used hybrid approaches. We make the comparison using the averages of LOOCV accuracy and the number of selected genes. This is due to the most previous works only evaluated the performance of their approaches using the LOOCV procedure or k -fold-cross-validation and the number of

selected genes on averages. At the moment, they used higher-dimensional data such as the SRBCT data set for experimental usage. Additionally, our work has used very higher-dimensional data (more than 12,000 genes) such as the lung data set to test the effectiveness of C-GASVM. The experimental result of the very higher-dimensional data is only shown in Tables 2, 3, and 4.

Table 5 The comparison between C-GASVM and other previous GASVM-based methods

Method	SRBCT Data Set (Average \pm S.D.; The Best)		
	#Genes	Accuracy (%)	
		LOOCV	Test
C-GASVM	(8.3 \pm 4.14; 20)	(100 \pm 0; 100)	(84.5 \pm 2.84; 90)
Huang and Chang, 2007	(6.2 \pm NA; NA)	NA	(98.75 using 10-CV \pm NA; NA)
Mohamad <i>et al.</i> , 2005	(10 \pm 0; 10)	(99.84 \pm 0.50; 100)	(68 \pm 9.49; 85)

Note: The best result of each data set shown in shaded cells. 'NA' means that the result is not reported in the related previous works. S.D. denotes the standard deviation, whereas 10-CV means 10-fold-cross-validation. #Genes represents a number of selected genes.

Table 5 displays the benchmark of this work and previous related works. The averages of LOOCV accuracy and test accuracy of our work were 100% and 84.5%, respectively. However, the average of the number of selected genes (8.3 genes) was slightly higher than the previous work (Huang and Chang, 2007). The work of Huang and Chang (Huang and Chang, 2007) only achieved 98.75% LOOCV accuracy on average using 6.2 average genes. The LOOCV accuracy and test accuracy genes set that produced in Mohamad *et al.*, (Mohamad *et al.*, 2005) were also less than our work. Overall, this work has outperformed the related previous works on the data sets in terms of LOOCV accuracy and the number of selected genes. The previous work is intractable to efficiently produce a near-optimal subset of genes in high-dimensional data due to their binary chromosome representation drawback (Huang and Chang, 2007).

5.5 CONCLUSIONS

In this paper, a cyclic GASVM-based method (C-GASVM) has been proposed and tested for gene selection on two real gene expression data that contain binary classes and multi-classes of tumour samples. Based on the experimental results, the performance of C-GASVM was superior to the other experimental methods and related previous works. This is due to the fact that C-GASVM can automatically reduce the dimensionality of the data on a cycle by cycle basis. When the dimensionality was reduced, the combination of genes and the complexity of solution spaces can also be automatically decreased iteratively. This iterative process is done to produce potential gene subsets from higher-dimensional data (gene expression data), and finally generate a near-optimal subset of informative genes. Hence, the gene selection using C-GASVM is needed to produce a smaller subset of informative genes for better cancer classification. Moreover, focusing the attention on the informative genes in the best subset may provide insights into the mechanisms responsible for the

cancer itself. However, due to the iterative process, C-GASVM is computationally more extensive than the other methods. Even though C-GASVM has classified tumours with higher accuracy, it is still not able to completely avoid the over-fitting problem. Therefore, a combination between a filter approach and a hybrid approach will be proposed to solve the computational time and over-fitting problems.

REFERENCES

- Chang, C.C., Lu, T., Chang, Y.F. and Lee, C.T. 2007. Reversible Data Hiding Schemes for Deoxyribonucleic Acid (DNA) Medium. *International Journal of Innovative Computing, Information and Control*, 3(5): 1145-1160.
- Huang, H.L. and Chang, F.L. 2007. ESVM: Evolutionary Support Vector Machine for Automatic Feature Selection and Classification of Microarray Data. *BioSystems*, 90(2): 516-528.
- Li, J., Chu, S., Ho, J. and Pan, J. 2007. Adaptive Data-dependent Matrix Norm Based Gaussian Kernel for Facial Feature Extraction. *International Journal of Innovative Computing, Information and Control*, 3(5): 1263-1272.
- Li, L., Jiang, W., Li, X., Moser, K.L., Guo, Z., Du, L., Wang, Q., Topol, E.J., Wang, Q. and Rao, S. 2005. A Robust Hybrid Between Genetic Algorithm and Support Vector Machine for Extracting an Optimal Feature Gene Subset. *Genomics*, 85(1): 16-23.

- Li, S., Wu, X. and Hu, X. 2008. Gene Selection Using Genetic Algorithm and Support Vectors Machines. *Soft Computing*, 12(7): 693-698.
- Mohamad, M.S., Deris, S. and Illias, R.M. 2005. A Hybrid of Genetic Algorithm and Support Vector Machine for Features Selection and Classification of Gene Expression Microarray. *International Journal of Computational Intelligence and Applications*, 5: 91-107.
- Mohamad, M.S., Omatu, S., Deris, S. and Hashim, S.Z.M. 2007. A Model for Gene Selection and Classification of Gene Expression Data. *International Journal of Artificial Life & Robotics*, 11(2): 219-222.
- Mohamad, M.S., Omatu, S., Deris, S., Misman, M.F. and Yoshioka, M. 2009. A Multi-objective Strategy in Genetic Algorithms for Gene Selection of Gene Expression Data, *International Journal of Artificial Life & Robotics*, 13(2).
- Peng, S., Xu, Q., Ling, X.B., Peng, X., Du, W. and Chen, L. 2003. Molecular Classification of Cancer Types from Microarray Data Using the Combination of Genetic Algorithms and Support Vector Machines. *FEBS Letters*, 555(2): 358-362.
- Saeys, Y., Inza, I. and Larranaga, P. 2007. A Review of Feature Selection Techniques in Bioinformatics. *Bioinformatics*, 23(19): 2507-2517.

6

SELECTING A SMALLER SUBSET OF GENES FOR LEUKEMIA CANCER CLASSIFICATION USING A TWO- STAGE METHOD

Mohd Saberi Mohamad

Sigeru Omatu

Safaai Deris

Michifumi Yoshioka

6.1 INTRODUCTION

Microarray technology is used to measure the expression levels of thousands of genes simultaneously, and finally produce microarray data. A comparison between the gene expression levels of cancerous and normal tissues can also be done. This comparison is useful to select those genes that might anticipate the clinical behaviour of cancers. Thus, there is a need to select informative genes that contribute to a cancerous state. However, the gene selection poses a major challenge because of the following characteristics of microarray data:

- High-dimensional data, for example, a huge number of genes and a small number of samples are in the ranges of 7,000-15,000 and 30-200, respectively.
- Most genes are not relevant for classifying different tissue types.
- These data have noisy genes.

To overcome the problems, a gene selection method is used to select a subset of genes that maximises the classifier's ability to classify samples more accurately. The gene selection method has several advantages such as improving classification accuracy, reducing the dimensionality of data, and removing irrelevant and noisy genes.

In the context of cancer classification, gene selection methods can be classified into two categories. If a gene selection method is carried out independently from a classifier, it belongs to the filter approach. Otherwise, it is said to follow a hybrid (wrapper) approach. In the early era of microarray analysis, most previous works have used the filter approach to select genes since it is computationally more efficient than the hybrid method (Saeys *et al.*, 2007). However, the hybrid approach usually provides greater accuracy than the filter approach. Until now, several hybrid methods (Huang and Chang, 2007; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*, 2009b; Mohamad *et al.*, 2009c; Peng *et al.*, 2003), especially a combination between a genetic algorithm (GA) (Elmahi *et al.*, 2006) and a support vector machine (SVM) (She *et al.*, 2008) classifier (GASVM), have been implemented to select informative genes. Generally, our previous hybrid methods, i.e., GASVM-based methods performed well in high-dimensional data since we proposed a modified chromosome representation, a cyclic approach, and a multi-objective strategy (Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*, 2009b; Mohamad *et al.*, 2009c). However, the methods yielded inconsistent results when they were run independently.

The previous work of Huang and Chang (Huang and Chang, 2007) that proposed GASVM-based methods can simultaneously optimise genes and SVM parameter settings. The work of Peng *et al.* (Peng *et al.* 2003) introduced a recursive feature elimination post-processing step after the step of a GASVM-based method in order to reduce the number of selected genes again. Nevertheless, the hybrid methods (GASVM-based methods) of the previous works are intractable to efficiently produce a smaller subset of genes in high-dimensional data due to

their binary chromosome representation drawback (Huang and Chang, 2007; Peng *et al.*, 2003). The total number of gene subsets produced by the GASVM-based methods in the previous works are calculated by $M_c = 2^M - 1$ where M_c is the total number of subsets, whereas M is the total number of genes. Based on this equation, the GASVM-based methods are almost impossible to evaluate all possible subsets of selected genes if M is too many (high-dimensional data). Although the work of Peng *et al.*, (Peng *et al.*, 2003) implemented a pre-processing step to decrease the dimensionality of data, but it can only reduce a small number of genes, and many genes are still available in the data. The GASVM-based methods (Huang and Chang, 2007; Peng *et al.*, 2003) also face with the high risk of over-fitting problems. An over-fitting problem is happened because the number of genes greatly exceeds the number of samples. The over-fitting problem that occurred on hybrid methods (e.g., GASVM-based methods) is also reported in a review paper in Saeys *et al.*, 2007.

In order to solve the problems derived from microarray data and overcome the limitation of the GASVM-based methods in the previous works (Huang and Chang, 2007; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*, 2009b; Mohamad *et al.*, 2009c; Peng *et al.*, 2003), we propose a two-stage gene selection method (Filter+MOGASVM). This proposed method is able to perform well in high-dimensional data and reduce a risk of over-fitting problems since it has two stages as follows: stage 1 to decrease the dimensionality of data; stage 2 to produce a smaller (near-optimal) genes subset. The diagnostic goal is to develop a medical procedure based on the least number of possible genes that needed to detect diseases. Thus, the ultimate goal of this paper is to select a smaller subset of informative genes (minimise the number of selected genes) for yielding higher cancer classification accuracy (maximise the classification accuracy). To achieve the goal, we adopt Filter+MOGASVM. The proposed method is evaluated on one real microarray data sets, namely the leukemia cancer data set.

The outline of this paper is as follows: Section 2 discusses the detail of the proposed Filter+MOGASVM. In Section 3, microarray data sets used, experimental setup, and experimental results are described. The conclusion of this paper is provided in Section 4.

6.2 THE PROPOSED TWO-STAGE GENE SELECTION METHOD (FILTER+MOGASVM)

In this paper, we propose Filter+MOGASVM to overcome the drawbacks of GASVM-based methods in the related previous works (Huang and Chang, 2007; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*, 2009b; Mohamad *et al.*, 2009c; Peng *et al.*, 2003). Filter+MOGASVM in our work differs from the methods in the previous works in one major part. The major difference is that our proposed method involves two stages (using a filter method and a hybrid method), whereas the previous works usually used only one stage (using a hybrid method) for gene selection. The difference is necessary in order to produce a smaller (near-optimal) gene subset from high-dimensional data and reduce the high risk of over-fitting problems. For more understanding, the general flowcharts of our work and the previous works are shown in Fig. 1 (a) and Fig. 1 (b), respectively. The detailed stages of Filter+MOGASVM are described as follows.

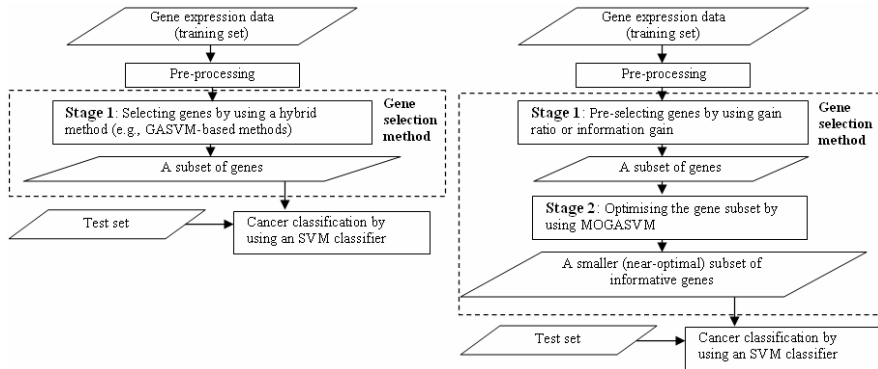


Figure 1 General flowcharts of (a) previous works (GASVM-based methods); (b) our work (Filter+MOGASVM)

6.2.1 Stage 1: Pre-Selecting Genes Using A Filter Method

In the first stage, we apply a filter method such as gain ratio (GR) or information gain (IG) on the training set to pre-select genes and finally produce a subset of genes. After the pre-select process, the dimensionality of data is also decreased. The filter method calculates and ranks a score for each gene. Genes with the highest scores are selected and put into the gene subset. This subset is used as an input to the second stage.

Since GASVM-based methods in previous works performs poorly in high-dimensional data, and meanwhile, we use a GASVM-based method (MOGASVM) in the second stage of Filter+MOGASVM, a filter method (GR or IG) in this first stage is used to reduce the high-dimensional in order to overcome the drawback of GASVM-based methods. If the subset that produced by the filter method is small-dimension, the combination of genes is not complex, and then MOGASVM in the next stage can possible to produce a smaller (near-optimal) subset of informative genes.

6.2.2 Stage 2: Optimizing A Gene Subset Using MOGASVM

In this stage, we develop and use MOGASVM to automatically optimise the gene subset that is produced by the first stage, and finally yield a smaller (near-optimal) subset of informative genes. This smaller subset is identified by an evaluation function in MOGASVM that uses two criteria: maximisation of the leave-one-out-cross-validation (LOOCV) accuracy and minimisation of the number of selected genes. MOGASVM selects and optimises genes by considering relations among them in order to remove irrelevant and noisy genes. The smaller subset is possible to be found due to the dimensionality and complexity of data has been firstly reduced by the first stage. The high risk of over-fitting problems can be also decreased because of the reduction. The detail of MOGASVM can be found in Mohamad *et al.* (Mohamad *et al.* 2009a)

Finally, the smaller subset of the training set is used to construct an SVM classifier for cancer classification, and the constructed SVM is then tested by using the test set (independent set). This paper has produced two methods of Filter+MOGASVM obtained from combinations of two different filter methods (GR and IG) and MOGASVM. These methods are GR+MOGASVM and IG+MOGASVM.

6.3 EXPERIMENTS

6.3.1 Data Sets

One benchmark microarray data set, namely the leukemia cancer data set is used to evaluate Filter+MOGASVM. It contains the expression levels of 7,129 genes and can be obtained at <http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi>. In this data set, bone marrow and blood samples were taken from 72 patients (47 acute lymphoblastic leukemia (ALL) samples, 25 acute myeloid leukemia (AML) samples). The training set contains 38 samples (27 ALL, 11 AML), whereas the test set consists 34 samples (20 ALL, 14 AML).

6.3.2 Experimental Setup

Since the number of training samples in microarray data is small, the cross-validation (CV) accuracy on the training set is calculated through an LOOCV procedure (Mohamad *et al.*, 2005). For the test accuracy, a classifier is built using all the training samples, and the classes of test samples from the test set are predicted one by one using the built classifier. The test accuracy is estimated by the number of the correctly classified, divided by the number of samples in the test set.

Table 1 Parameter Settings for Filter+MOGASVM

Parameter	Leukemia Data Set
Size of population	100
Number of generation	300
Crossover rate	0.7
Mutation rate	0.01
Weight 1, w_1	0.8
Weight 2, w_2	0.2
<i>Cost</i> for SVM	100

Table 1 contains parameter values for Filter+MOGASVM. These values are chosen based on the results of preliminary runs. Three criteria following their importance are considered to evaluate and compare the performance of Filter+MOGASVM with existing methods (Huang and Chang, 2007; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*, 2009b; Mohamad *et al.*, 2009c; Peng *et al.*, 2003): test accuracy, CV accuracy, and the number of selected genes. Higher accuracies and a smaller number of selected genes are needed to obtain an excellent performance. The top 200 genes are pre-selected by using GR and IG in the first stage of the proposed method, and are then used for the second stage. Several experiments are conducted 10 times on each data set using Filter+MOGASVM and other experimental methods such as GASVM (single-objective), MOGASVM, GASVM version 2 (GASVM-II), and SVM. Filter+GASVM methods (IG+GASVM and GR+GASVM) are also experimented for the comparison. Next, an average result of the 10 independent runs is obtained.

6.3.3 Experimental Results

6.3.3.1 LOOCV And Test Accuracies Of Selected Genes With Filter+MOGASVM

Table 2 shows the results for each run on the leukemia data set. The results of the best subsets are shown in shaded cells, whereas the results in boldface display the best result of averages. S.D. denotes the standard deviation. Run# and #Genes represent a run number and a number of selected genes, respectively. Almost all runs have achieved 100% LOOCV accuracy. This has proved that Filter+MOGASVM has efficiently selected and produced a near-optimal gene subset from a solution space.

Table 2 Classification accuracies using Filter+MOGASVM on the leukemia data set

Run#	GR+MOGASVM (Filter+MOGASVM)			IG+MOGASVM (Filter+MOGASVM)		
	LOOCV (%)	Test (%)	#Genes	LOOCV (%)	Test (%)	#Genes
1	100	91.18	3	100	91.18	3
2	100	88.24	3	100	91.18	3
3	100	94.12	2	100	94.12	3
4	100	91.18	2	100	91.18	2
5	100	91.18	3	100	91.18	3
6	100	94.12	3	100	88.24	2
7	100	91.18	2	100	94.12	2
8	100	91.18	3	100	88.24	3
9	100	94.12	3	100	85.30	3
10	100	91.18	3	100	91.18	3
Average	100	91.77	2.70	100	90.59	2.70
± S.D.	± 0	± 1.86	± 0.48	± 0	± 2.70	± 0.48

Note: The results of the best subsets shown in shaded cells. Result in boldface displayed the best result of averages. S.D. denotes the standard deviation.

6.3.3.2 *Filter+MOGASVM Versus Other Experimental Methods*

The benchmark of Filter+MOGASVM in comparison with other experimental methods that have been experimented in this work is summarized in Table 3. Overall, the LOOCV and test accuracies of Filter+MOGASVM for all the data sets were higher than Filter+GASVM, MOGASVM, GASVM-II, GASVM, and SVM. Moreover, the number of selected genes by using Filter+MOGASVM was also lower.

Based on the standard deviations of LOOCV accuracy, test accuracy, and the number of selected genes, Filter+MOGASVM was also more consistent than the other experimental methods except the SVM classifier. This SVM classifier achieved 0 for the standard deviations in all experiments since it did not implement any gene selection approach. The gap between LOOCV accuracy and test accuracy that resulted by Filter+MOGASVM was also lower. This small gap shows that the risk of the over-fitting problem can be reduced. On the other hand, the results of LOOCV accuracy of the others were much higher than their test accuracy because they were unable to avoid or reduce the risk of over-fitting problems. Over-fitting is a major problem of hybrid methods in gene selection and classification of microarray data when the classification accuracy on training samples, e.g., LOOCV accuracy is much higher than the test accuracy.

GASVM and MOGASVM cannot produce a near-optimal subset of informative genes because they perform poorly in high-dimensional data due to their chromosome representation drawback. GASVM-II method is impractical to be used in real applications because a variety number of selected genes should be tested in order to obtain the near-optimal one. On the contrary, the proposed Filter+MOGASVM that pre-selects a number of genes in the first stage can automatically optimise the selected genes by the second stage in order to remove irrelevant genes and produce a smaller (near-optimal) subset of informative genes.

Table 3 Classification accuracies using Filter+MOGASVM on the Leukemia data set

Method	Leukemia Data Set (Average \pm S.D.; The Best)		
	#Genes	Accuracy (%)	
		LOOCV	Test
GR+MOGASVM (Filter+MOGASVM)	2.70 \pm 0.48; 3	100 \pm 0; 100	91.77 \pm 1.86; 94.12
IG+MOGASVM (Filter+MOGASVM)	2.70 \pm 0.48; 2	100 \pm 0; 100	90.59 \pm 2.70; 94.12
GR+GASVM (Filter+GASVM)	97.40 \pm 4.43; 91	100 \pm 0; 100	86.18 \pm 1.99; 88.24
IG+GASVM (Filter+GASVM)	99.30 \pm 6.29; 96	100 \pm 0; 100	88.53 \pm 2.93; 91.18
A recursive GASVM (Mohamad <i>et al.</i> , 2009c)	2.9 \pm 1.73; 2	100 \pm 0; 100	88.82 \pm 3.04; 94.12
GASVM-II+GASVM (Mohamad <i>et al.</i> , 2009b)	3.4 \pm 1.35; 2	100 \pm 0; 100	85.88 \pm 8.86; 97.06
GASVM-II (Mohamad <i>et al.</i> , 2005)	10 \pm 0; 10	100 \pm 0; 100	81.18 \pm 0.21; 94.12
MOGASVM (Mohamad <i>et al.</i> , 2009a)	2,212.6 \pm 26.63; 2,189	95.53 \pm 1.27; 97.37	84.41 \pm 2.42; 88.24
GASVM (Mohamad <i>et al.</i> , 2005)	3,574.9 \pm 40.05; 3,531	94.74 \pm 0; 94.74	83.53 \pm 2.48; 88.24
SVM classifier (Mohamad <i>et al.</i> , 2009a)	7,129 \pm 0; 7,129	94.74 \pm 0; 94.74	85.29 \pm 0; 85.29

Note: The best result shown in shaded cells. S.D. denotes the standard deviation, whereas #Genes represents a number of selected genes.

6.3.3.3 *Filter+MOGASVM Versus Other Experimental Methods*

Table 4 displays the benchmark of this work and previous related works on the leukemia data set. The averages of LOOCV accuracy and the number of selected genes of our work were 100% and 2.7 genes, respectively. The latest previous work, Huang and Chang (Huang and Chang, 2007) also came up with the similar LOOCV result to ours, but the number of selected genes is slightly higher in order to obtain the same result. The work of Peng *et al.* (Peng *et al.*, 2003) analysed this data set and finally yielded 100% average LOOCV accuracy with six average selected genes. Overall, this work has outperformed the related previous works in terms of classification accuracy and the number of selected genes. Filter+MOGASVM in our work has produced a near-optimal (smaller) gene subset from high-dimensional data and reduced the high risk of over-fitting problems. This is due to the fact that a filter method in the first stage of Filter+MOGASVM reduces the dimensionality of the solution space in order to produce a gene subset. Next, MOGASVM in the second stage of Filter+MOGASVM optimises the subset automatically to yield a smaller subset of informative genes with higher classification accuracy. This smaller subset is obtained since Filter+MOGASVM considers and optimises a relation among genes.

Unfortunately, the previous works (Huang and Chang, 2007; Peng *et al.* 2003) did not provide any test accuracy result on the test set (independent data set) and did not show any standard deviation result for comparative comparison with our work. GASVM-based methods in the previous works may almost possible face with a high risk of over-fitting problems and the difficulty to obtain a near-optimal solution in high-dimensional data since they used binary chromosome representation for gene selection mechanisms. This is also supported by a review paper in Saeys *et al.* (Saeys *et al.* 2007) which reported that hybrid methods (e.g., GASVM-based methods) confront with the risk of over-fitting problems because of the high-dimensional data.

Table 4 The comparison between our proposed method (Filter+MOGASVM) and other previous GASVM-based methods

Data	Experiment Evaluation		Our work (Filter+MOGASVM)	Huang and Chang, 2007	Peng et al., 2003
Leukemia (Average ± S.D; The Best)	CV Accuracy (%)		100 ± 0; 100 (using LOOCV)	100 ± NA; NA (using 10-CV)	100 ± NA; NA (using LOOCV)
	The Test Accuracy (%)		91.77 ± 1.86; 94.12	NA	NA
	#Genes		2.70 ± 0.48; 3	3.4 ± NA; NA	6 ± NA; NA

Note: The best result shown in shaded cells. ‘NA’ means that results are not reported in the related previous works. S.D. denotes the standard deviation, whereas 10-CV means 10-fold-cross-validation. #Genes represents a number of selected genes.

6.4 CONCLUSION

In this paper, Filter+MOGASVM has been proposed and tested for gene selection on the leukemia microarray data set. Based on the experimental results, the performance of Filter+MOGASVM was superior to the other experimental methods and related previous works. This is due to the fact that the filter method in the first stage of the proposed method can pre-select genes and reduce dimensionality of data in order to produce a subset of genes. When the dimensionality was reduced, the combination of genes and complexity of solution spaces were automatically decreased. The second stage of Filter+MOGASVM can automatically optimise the subset that is yielded by the first stage. This optimisation process is done to remove irrelevant and noisy genes, and finally produce a smaller (near-optimal) subset of informative genes. Hence, the gene selection using Filter+MOGASVM is needed to produce a

smaller subset of informative genes for better cancer classification of microarray data. However, due to the application of a filter method in the first stage of Filter+MOGASVM, pre-selecting genes is difficult since it is manually done. Even though Filter+MOGASVM has classified tumours with higher accuracy, it is still not able to completely avoid the over-fitting problem. Therefore, a combination between constraint based reasoning methods and particle swarm optimisation techniques is recently developed to solve the over-fitting problem.

REFERENCES

- Elmahi, I., Grunder, O. and Elmoudni, A. 2006. A Modelling-Optimization Approach for Discrete Event Systems Using the (MAX,+) Algebra and Genetic Algorithms. *International Journal of Innovative Computing, Information and Control*, 2(4): 771-788.
- Huang, H.L. and Chang, F.L. 2007. ESVM: Evolutionary Support Vector Machine for Automatic Feature Selection and Classification of Microarray Data. *BioSystems*, 90(2): 516-528.
- Mohamad, M.S., Deris, S. and Illias, R.M. 2005. A Hybrid of Genetic Algorithm and Support Vector Machine for Features Selection and Classification of Gene Expression Microarray. *International Journal of Computational Intelligence and Applications*, 5: 91-107.
- Mohamad, M.S., Omatu, S., Deris, S., Misman, M.F. and Yoshioka, M. 2009a. A Multi-objective Strategy in Genetic Algorithms for Gene Selection of Gene Expression Data, *International Journal of Artificial Life & Robotics*, 13(2).

- Mohamad, M.S., Omatu, S., Deris, S., Misman, M.F. and Yoshioka, M. 2009b. Selecting Informative Genes from Microarray Data by Using Hybrid Methods for Cancer Classification”, *International Journal of Artificial Life & Robotics*, 13(2).
- Mohamad, M.S., Omatu, S., Deris, S. and Yoshioka, M. 2009c. A Recursive Genetic Algorithm to Automatically Select Genes for Cancer Classification. *Proceedings of the 2nd International Workshop on Practical Application of Computational Biology & Bioinformatics*, Corchado, J.M., Paz, J.F.D., Rocha, M.P. and Juan, F.F.R, (eds.). *Advances in Soft Computing*, Berlin/Heidelberg, Springer-Verlag, 49: 166-174.
- Peng, S., Xu, Q., Ling, X.B., Peng, X., Du, W. and Chen, L. 2003. Molecular Classification of Cancer Types from Microarray Data Using the Combination of Genetic Algorithms and Support Vector Machines. *FEBS Letters*, 555(2): 358-362.
- She, Q., Su, H., Dong, L. and Chu, J. 2008. Support Vector Machine with Adaptive Parameters in Image Coding. *International Journal of Innovative Computing, Information and Control*, 4(2): 359-367.
- Saeys, Y., Inza, I. and Larranaga, P. 2007. A Review of Feature Selection Techniques in Bioinformatics. *Bioinformatics*, 23(19): 2507-2517.

7

A THREE-STAGE METHOD: SELECTION OF INFORMATIVE GENES BASED ON MIXED-LINEAGE LEUKEMIA CANCER DATA

Mohd Saberi Mohamad

Sigeru Omatu

Safaai Deris

Michifumi Yoshioka

7.1 INTRODUCTION

Microarray technology is used to measure the expression levels of thousands of genes simultaneously, and finally produce microarray data. A comparison between the gene expression levels of cancerous and normal tissues can also be done. This comparison is useful to select those genes that might anticipate the clinical behaviour of cancers. Thus, there is a need to select informative genes that contribute to a cancerous state. An informative gene is useful for cancer classification. However, the gene selection process poses a major challenge because of the following characteristics of microarray data: the huge number of genes compared to the small number of samples (higher-dimensional data), irrelevant genes, and noisy data.

To overcome the challenge, a gene selection method is used to select a subset of genes for cancer classification. The gene selection method has several advantages such as maintaining or

improving classification accuracy, reducing the dimensionality of data, and removing irrelevant and noisy genes.

There are two types of gene selection methods (Shah and Kusiak, 2007): if a gene selection method is carried out independently from a classifier, it belongs to the filter approach; otherwise, it is said to follow a hybrid (wrapper) approach. In the early era of microarray analysis, most previous works have used the filter approach to select genes because it is computationally more efficient than the hybrid approach (Armstrong *et al.*, 2002; Li *et al.*, 2003; Yang *et al.*, 2006). However, the filter approach results in inclusion of irrelevant and noisy genes in a gene subset for the cancer classification. The hybrid approach usually provides greater accuracy than the filter approach. Until now, several hybrid methods, especially a combination between a genetic algorithm (GA) and a support vector machine (SVM) classifier (GASVM), have been implemented to select informative genes (Huang and Chang, 2007; Lee, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*, 2009b; Peng *et al.*, 2003; Shah and Kusiak, 2007). The drawbacks of the hybrid methods (GASVM-based methods) in the previous works are: 1) intractable to efficiently produce a smaller subset of informative genes when the total number of genes is too large (higher-dimensional data); 2) the high risk of over-fitting problems.

In order to solve the problems derived from microarray data and overcome the limitation of the hybrid methods in the previous works (Huang and Chang, 2007; Lee, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*, 2009b; Peng *et al.*, 2003; Shah and Kusiak, 2007), we propose a three-stage gene selection method (3-SGS). This proposed method is able to perform well in the higher-dimensional data and reduce the high risk of over-fitting problems since it has three stages as follows: stage 1 for producing a subset of genes; stage 2 for resulting near-optimal subsets of genes; stage 3 for yielding a smaller (final) subset of informative genes based on the frequency of appearance for each gene in the near-optimal subsets. The diagnostic goal is to develop a medical procedure based on the least number of possible

genes that needed to detect diseases. Thus, the ultimate goal of this paper is to select a smaller subset of informative genes (minimise the number of selected genes) for yielding higher cancer classification accuracy (maximise the classification accuracy). To achieve the goal, we adopt 3-SGS. 3-SGS is evaluated on one real microarray data set, namely the leukemia cancer data set.

The outline of this paper is as follows: Sections 2 and 3 discuss previous works and the detail of the proposed three-stage method, respectively. In Section 4, microarray data sets used, experimental setup, and experimental results are described. The conclusion of this paper is provided in Section 5.

7.2 PREVIOUS WORKS

Several hybrid methods, i.e., GASVM-based methods have been proposed for genes selection of microarray data (Huang and Chang, 2007; Lee, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*, 2009b; Peng *et al.*, 2003; Shah and Kusiak, 2007). The hybrid methods usually provide greater accuracy than filter methods since genes are selected by considering relations among genes. Generally, our previous GASVM-based methods performed well in higher-dimensional data, e.g., microarray data since we proposed a modified chromosome representation and a multi-objective approach (Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*, 2009b). However, the methods yielded inconsistent results when they were run independently.

The work of Huang and Chang (Huang and Chang, 2007) can simultaneously optimise genes and SVM parameter settings by using a GASVM-based method. Next, integrated algorithms based on GASVM have been proposed by the works of Shah and Kusiak (Shah and Kusiak, 2007), and Lee (Lee, 2008) to produce a small

subset of genes. Peng *et al.* (Peng *et al.*, 2003) introduced a recursive feature elimination post-processing step after the step of a GASVM-based method in order to reduce the number of selected genes again.

Nevertheless, the GASVM-based methods of the previous works are still intractable to efficiently produce a smaller subset of informative genes from higher-dimensional data due to their binary chromosome representation drawback (Huang and Chang, 2007; Lee, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*, 2009b; Peng *et al.*, 2003; Shah and Kusiak, 2007). The total number of gene subsets produced by GASVM-based methods is calculated by $M_c = 2^M - 1$ where M_c is the total number of gene subsets, whereas M is the total number of genes. Based on this equation, the GASVM-based methods are almost impossible to evaluate all possible subsets of selected genes if M is too many (higher-dimensional data). Although the work of Peng *et al.* (Peng *et al.*, 2003) have implemented a pre-processing step to decrease the dimensionality of data, but it can only reduce a small number of genes, and many genes are still available in the data (Peng *et al.*, 2003). The GASVM-based methods also face with the high risk of over-fitting problems. The over-fitting problem that occurred on hybrid methods (e.g., GASVM-based methods) was also reported in a review paper in Saeys *et al.* (Saeys *et al.*, 2007).

7.3 THE PROPOSED THREE-STAGE GENE SELECTION METHOD (3-SGS)

In order to overcome the drawbacks of GASVM-based methods in the related previous works (Huang and Chang, 2007; Lee, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*,

2009b; Peng *et al.*, 2003; Shah and Kusiak, 2007), we propose a three-stage gene selection method (3-SGS). 3-SGS in our work differs from the methods in the previous works in one major part. The major difference is that our proposed method involves three stages, whereas the previous works usually used only one stage (using a hybrid method) (Huang and Chang, 2007; Lee, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*, 2009b; Shah and Kusiak, 2007) or two stages (using a filter method and a hybrid method) (Peng *et al.*, 2003) for gene selection. In the third stage, our method implements frequency analysis to identify the most frequently selected genes in near-optimal gene subsets, whereas the previous works (Huang and Chang, 2007; Lee, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*, 2009b; Peng *et al.*, 2003; Shah and Kusiak, 2007) rely solely on a filter method or a hybrid method in the first stage of their methods. The difference is necessary in order to produce near-optimal gene subsets from higher-dimensional data, reduce the high risk of over-fitting problems, and finally yield a smaller subset of informative genes. 3-SGS is shown in Fig. 1. The detailed stages are described as follows:

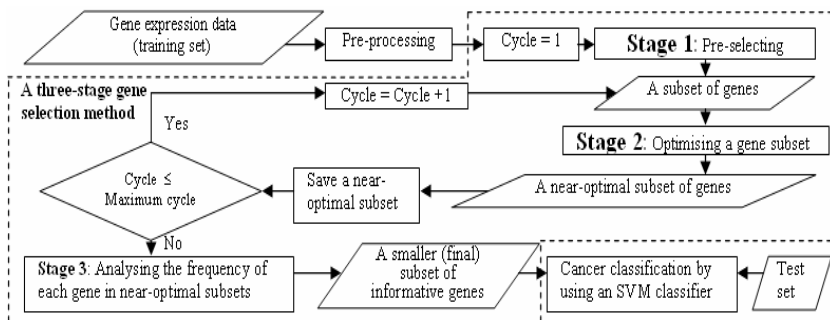


Figure 1 The proposed method (3-SGS)

7.3.1 Stage 1: Pre-selecting Genes Using a Filter Method

A filter method such as gain ratio (GR) or information gain (IG) is used to pre-select genes and finally produce a subset of genes. After the pre-select process, the dimensionality of data is also decreased. The filter method calculates and ranks a score for each gene. Genes with the highest scores are selected and put into a gene subset. This subset is then used as an input to the second stage.

Since GASVM-based methods in previous works performs poorly in higher-dimensional data, and meanwhile, we also use a GASVM-based method, i.e., a multi-objective GASVM (MOGASVM) in the second stage of 3-SGS, a filter method (GR or IG) in this first stage is used to reduce the higher-dimensional in order to overcome the drawback of GASVM-based methods. If the subset that produced by the filter method is in small-dimension, the combination of genes is not complex, and then MOGASVM can possible to produce near-optimal genes subsets.

7.3.2 Stage 2: Optimising a Gene Subset Using MOGASVM

In this stage, we develop MOGASVM to automatically optimise a gene subset that is produced by the first stage, and finally yield near-optimal subsets of genes. This stage is cycled until the maximum number of cycles is satisfied. The near-optimal subsets are identified by an evaluation function in MOGASVM that uses two criteria: maximisation of leave-one-out-cross-validation (LOOCV) accuracy and minimisation of the number of selected genes. MOGASVM selects and optimises genes by considering relations among them in order to remove irrelevant and noisy genes. The near-optimal subsets are possible to be found due to the

dimensionality and complexity of data has been firstly reduced by the first stage. The high risk of over-fitting problems can be also decreased because of the reduction. The detail of MOGASVM can be found in our previous work (Mohamad *et al.*, 2009b).

7.3.3 Stage 3: Analysing the Frequency of Each Gene in Near-optimal Subsets

The frequency of appearance for each gene in each near-optimal gene subset is examined and analysed to assess the relative importance of genes for cancer classification. The most frequently selected genes in near-optimal gene subsets are presumed to be the most relevant for the classification. Finally, a smaller (final) subset of informative genes (K genes, K is a number of genes) is produced and used to construct an SVM classifier. This subset contains a smaller number of informative genes with higher classification accuracy. This paper has produced two methods of 3-SGS obtained from combinations of two different filter methods (GR and IG) and MOGASVM. These methods are 3-SGS-GR and 3-SGS-IG.

7.4 EXPERIMENTS

7.4.1 Data Sets and Experimental Setup

The mixed-lineage leukemia (MLL) microarray data set is used to evaluate 3-SGS. The MLL cancer data set is a multi-classes data set. It has three leukemia classes: acute lymphoblastic leukemia (ALL), acute myeloid leukemia (AML), and MLL. The training set contains 57 samples (20 ALL, 17 MLL, and 20 AML). While the testing set contains 4 ALL, 3 MLL, and 8 AML samples. There are 12,582 genes in each sample. This data set was published by Armstrong *et al.* (Armstrong *et al.*, 2002). It can be downloaded at http://www.broad.mit.edu/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=63.

Table 1 Parameter settings for 3-SGS

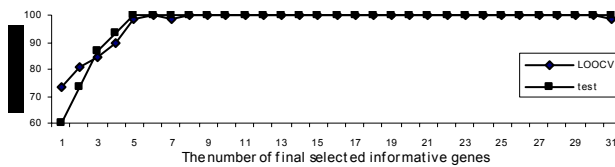
Parameters	MLL data set
Size of population	100
Number of generation	300
Crossover rate	0.7
Mutation rate	0.01
Maximum number of cycles	10
Cost for an SVM classifier	100

Table 1 contains parameter values for 3-SGS. These values are chosen based on the results of preliminary runs. Three criteria following their importance are considered to evaluate the performance of 3-SGS: test accuracy on the test set, LOOCV accuracy on the training set, and the number of selected genes. Higher accuracies and a smaller number of selected genes are needed to obtain an excellent performance. The top 200 genes are pre-selected by using GR and IG in the first stage of the 3-SGS, and are then used for the second stage.

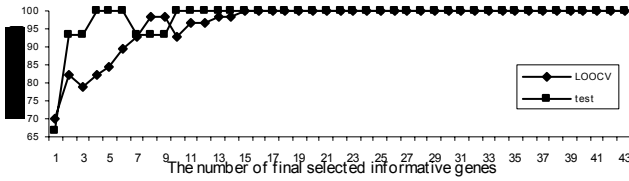
7.4.2 Experimental Results

7.4.2.1 Classification accuracies of final informative genes

As shown in Fig. 2, the best results of the MLL (100% LOOCV and 100% test accuracies) are obtained by using the only six (using 3-SGS-GR) final selected informative genes (K genes).



(a) 3-SGS-GR on the MLL data set



(b) 3-SGS-IG on the MLL data set

Figure 2 A relation between classification accuracies and the number of final selected informative genes (K genes) using 3-SGS

Many runs have achieved 100% LOOCV accuracy. This has proved that 3-SGS has efficiently selected and produced a smaller subset of informative genes from a solution space. This is due to the fact that a filter method in the first stage of 3-SGS reduces the dimensionality of the solution space in order to produce a gene subset. Next, MOGASVM in the second stage of 3-SGS optimise the subset automatically to yield near-optimal subsets of genes. These subsets are obtained since MOGASVM in 3-SGS considers and optimises a relation among genes. Finally, the first K genes appearing most frequently are selected as the final selected informative genes for cancer classification.

7.4.2.2 A list of informative genes for biological usage

The informative genes and their rank scores (frequency) of the final subsets as produced by the proposed 3-SGS and reported in Fig. 2 are listed in Table 2. These informative genes among the thousand of genes may be the excellent candidates for clinical and medical investigations. Biologists can save much time since they can directly refer to the genes that have higher possibility to be useful for cancer diagnosis and drug target in the future.

Table 2 The list of informative genes in the final gene subsets

Data Set	Rank Score	Gene ID	Gene Description
MLL	9	M11722	human terminal transferase mRNA, complete cds
	7	M13143	nucleotide sequence of the cDNA insert of lambda
	3	U41843	human Dr1-associated corepressor (DRAP1) mRNA
	3	Z83844	vicpro2.D07.r Homo sapiens cDNA, 5' end
	2	L08895	homo sapiens MADS
	2	U59878	human low-Mr GTP-binding protein (RAB32) mRNA

7.4.2.3 3-SGS versus other previous methods

Table 3 displays the benchmark of this work and previous related works that used filter and hybrid approaches. Overall, 3-SGS in this work has outperformed the previous works on MLL the data set in terms of the test accuracy, the LOOCV accuracy, and the number of selected genes.

Table 3 The benchmark of 3-SGS with previous methods on the MLL data set

Gene Selection Method (Category) [Reference]	MLL Data Set		
	#Selected Genes	Accuracy (%)	
		CV	Test
3-SGS (Filter, hybrid, and frequency analysis)	6	100	100
GASVM (Hybrid) (Huang and Chang, 2007)	(3.5)	(100)	-
F-test and Cho's method (Filter) (Yang <i>et al.</i> , 2006)	23	97.2	-
Principal component analysis (Filter) (Armstrong <i>et al.</i> , 2002)	100	95	-
Information gain (Filter) (Li <i>et al.</i> , 2003)	-	-	100
<i>GASVM-II+GASVM</i> (Hybrid) (Mohamad <i>et al.</i> , 2009a)	(6.5)	(100)	(92)
<i>GASVM-II</i> (Hybrid) (Mohamad <i>et al.</i> , 2005)	(30)	(100)	(84.67)
<i>MOGASVM</i> (Hybrid) (Mohamad <i>et al.</i> , 2009b)	(4,465.2)	(94.74)	(90)
<i>GASVM</i> (Hybrid) (Mohamad <i>et al.</i> , 2005)	(6,298.8)	(94.74)	(87.33)

Note: The results of the best subsets shown in shaded cells. '-' means that a result is not reported in the related previous work. A result in '()' denotes an average result. CV and #Selected Genes represent cross-validation and a number of selected genes, respectively. Methods in *italics* style are experimented in this work.

Generally, filter methods in previous works (Armstrong *et al.*, 2002; Li *et al.*, 2003; Yang *et al.*, 2006) achieved poor performances since they may result in inclusion of irrelevant and noisy genes in a gene subset for the cancer classification. This situation is happen because the methods evaluate a gene based on its discriminative power for the target classes without considering its relations with other genes.

GASVM-based methods (Huang and Chang, 2007; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009a; Mohamad *et al.*, 2009b; Peng *et al.*, 2003) may be unable to produce a smaller subset of informative genes because they perform poorly in higher-dimensional data due to their chromosome representation drawback. GASVM-II (Mohamad *et al.*, 2005) method is impractical to be used in real applications because a variety number of selected genes should be tested in order to obtain the near-optimal one. On the contrary, the proposed 3-SGS that pre-selects a number of genes at the first stage can reduce the data dimensionality and produce a gene subset. This subset is then optimised by MOGASVM in the second stage of 3-SGS to yield near-optimal subsets. Finally, the first K genes appearing most frequently are selected as the final selected informative genes (a smaller subset) for cancer classification.

The gap between LOOCV accuracy and test accuracy that resulted by 3-SGS was also lower. This small gap shows that the risk of the over-fitting problem can be reduced. On the other hand, the results of LOOCV accuracy of the related previous works were much higher than their test accuracy because they were unable to avoid or reduce the risk of over-fitting problems. The previous work that used GASVM-based methods (Huang and Chang, 2007) did not provide any test accuracy results and thus, the over-fitting problem could not be investigated in their works. Over-fitting is a major problem on hybrid methods in gene selection and classification of microarray data when the classification accuracy on training samples, e.g., LOOCV accuracy is much higher than the test accuracy. This is also supported by a review paper in *Saeys*

et al. (Saeys *et al.* 2007) which reported that hybrid methods (e.g., GASVM-based methods) confront with the high risk of over-fitting problems because of the higher-dimensional data.

7.5 CONCLUSIONS

In this paper, Filter+MOGASVM has been proposed and tested for gene selection on the leukemia microarray data set. Based on the experimental results, the performance of Filter+MOGASVM was superior to the other experimental methods and related previous works. This is due to the fact that the filter method in the first stage of the proposed method can pre-select genes and reduce dimensionality of data in order to produce a subset of genes. When the dimensionality was reduced, the combination of genes and complexity of solution spaces were automatically decreased. The second stage of Filter+MOGASVM can automatically optimise the subset that is yielded by the first stage. This optimisation process is done to remove irrelevant and noisy genes, and finally produce a smaller (near-optimal) subset of informative genes. Hence, the gene selection using Filter+MOGASVM is needed to produce a smaller subset of informative genes for better cancer classification of microarray data. However, due to the application of a filter method in the first stage of Filter+MOGASVM, pre-selecting genes is difficult since it is manually done. Even though Filter+MOGASVM has classified tumours with higher accuracy, it is still not able to completely avoid the over-fitting problem. Therefore, a combination between constraint based reasoning methods and particle swarm optimisation techniques is recently developed to solve the over-fitting problem.

REFERENCES

- Armstrong, S.A., Staunton, J.E., Silverman, L.B., Pieters, R., Boer, M.L., Minden, M.D., Sallan, S.E., Lander, E.S., Golub, T.R. and Korsmeyer, S.J. 2002. MLL Translocations Specify a Distinct Gene Expression Profile that Distinguishes a Unique Leukemia. *Nature Genetics*, **30**(1): 41-47.
- Huang, H.L. and Chang, F.L. 2007. ESVM: Evolutionary Support Vector Machine for Automatic Feature Selection and Classification of Microarray Data. *BioSystems*, **90**(2): 516-528.
- Lee, Z.J. 2008. An Integrated Algorithm for Gene Selection and Classification Applied to Microarray Data of Ovarian Cancer. *Artificial Intelligence in Medicine*, **42**(1): 81-93.
- Li, J., Liu, H., Ng, S. K. and Wong, L. 2003. Discovery of Significant Rules for Classifying Cancer Diagnosis Data. *Bioinformatics*, **19**: 93-102.
- Mohamad, M.S., Deris, S. and Illias, R.M. 2005. A Hybrid of Genetic Algorithm and Support Vector Machine for Features Selection and Classification of Gene Expression Microarray. *International Journal of Computational Intelligence and Applications*, **5**: 91-107.
- Mohamad, M.S., Omatu, S., Deris, S., Misman, M.F. and Yoshioka, M. 2009a. Selecting Informative Genes from Microarray Data by Using Hybrid Methods for Cancer Classification”, *International Journal of Artificial Life & Robotics*, **13**(2).
- Mohamad, M.S., Omatu, S., Deris, S., Misman, M.F. and Yoshioka, M. 2009b. A Multi-objective Strategy in Genetic Algorithms for Gene Selection of Gene Expression Data, *International Journal of Artificial Life & Robotics*, **13**(2).

- Peng, S., Xu, Q., Ling, X.B., Peng, X., Du, W. and Chen, L. 2003. Molecular Classification of Cancer Types from Microarray Data Using the Combination of Genetic Algorithms and Support Vector Machines. *FEBS Letters*, **555**(2): 358-362.
- Saeys, Y., Inza, I. and Larranaga, P. 2007. A Review of Feature Selection Techniques in Bioinformatics. *Bioinformatics*, **23**(19): 2507-2517.
- Shah, S. and Kusiak, A. 2007. Cancer Gene Search with Data-mining and Genetic Algorithms. *Computers in Biology & Medicine*, **37**(2): 251-261.
- Yang, K., Cai, Z., Li, J. and Lin, G. 2006. A Stable Gene Selection in Microarray Data Analysis, *BMC Bioinformatics*, **7**:228-246.

8

AN ITERATIVE GA-BASED APPROACH: GENE SELECTION AND CLASSIFICATION OF LUNG CANCER DATA

Mohd Saberi Mohamad
Sigeru Omatu
Safaai Deris
Michifumi Yoshioka

8.1 INTRODUCTION

Advances in the area of microarray-based gene expression analyses have led to a promising future of cancer diagnosis using new molecular-based approaches. This microarray technology is used to measure the expression levels of thousands of genes simultaneously, and finally produce microarray data. A comparison between the gene expression levels of cancerous and normal tissues can also be done. This comparison is useful to select those genes that might anticipate the clinical behaviour of cancers. Thus, there is a need to select informative genes that contribute to a cancerous state. However, the gene selection process poses a major challenge because of the characteristics of microarray data: the huge number of genes compared to the small number of samples (higher-dimensional data), irrelevant genes, and noisy data.

To overcome the challenge, a gene selection method is

used to select a subset of genes that increases the classifier's ability to classify samples more accurately. The gene selection method has several advantages such as improving classification accuracy, reducing the dimensionality of data, and removing irrelevant and noisy genes.

There are two types of gene selection methods (Li *et al.*, 2008; Mohamad *et al.*, 2005): if a gene selection method is carried out independently from a classifier, it belongs to the filter approach; otherwise, it is said to follow a hybrid (wrapper) approach. In the early era of microarray analysis, most previous works have used the filter approach to select genes because it is computationally more efficient than the hybrid approach. However, the hybrid approach usually provides greater accuracy than the filter approach since the genes are selected by considering and optimising relations among genes (Saeys *et al.*, 2007). Until now, several hybrid methods, especially a combination between a genetic algorithm (GA) and a support vector machine (SVM) classifier (GASVM), have been implemented to select informative genes (Li *et al.*, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009; Peng *et al.*, 2003). The drawbacks of the hybrid methods (GASVM-based methods) in the previous works are: 1) intractable to efficiently produce a near-optimal subset of informative genes when the total number of genes is too large (higher-dimensional data) due to the drawback of binary chromosome representation; 2) the high risk of over-fitting problems. The over-fitting problem that occurred on hybrid methods (e.g., GASVM-based methods) was also reported in a review paper in Saeys *et al.* (Saeys *et al.*, 2007).

In order to overcome the limitations of the previous works and solve the problems derived from microarray data, we propose an iterative approach based on multi-objective GASVM (MOGASVM). The ultimate goal of this paper is to automatically select a near-optimal (smaller) subset of informative genes that is most relevant for the cancer classification. To achieve the goal, we adopt the proposed method. It is evaluated on real microarray data set, namely lung cancer data set.

8.2 THE PROPOSED ITERATIVE APPROACH BASED ON MOGASVM (I-GA)

In this paper, we propose I-GA to overcome the problems derived from the previous works and microarray data (Li *et al.*, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009; Peng *et al.*, 2003). I-GA is a hybrid approach based on MOGASVM. Details of MOGASVM can be found in Mohamad *et al.* (Mohamad *et al.* 2009).

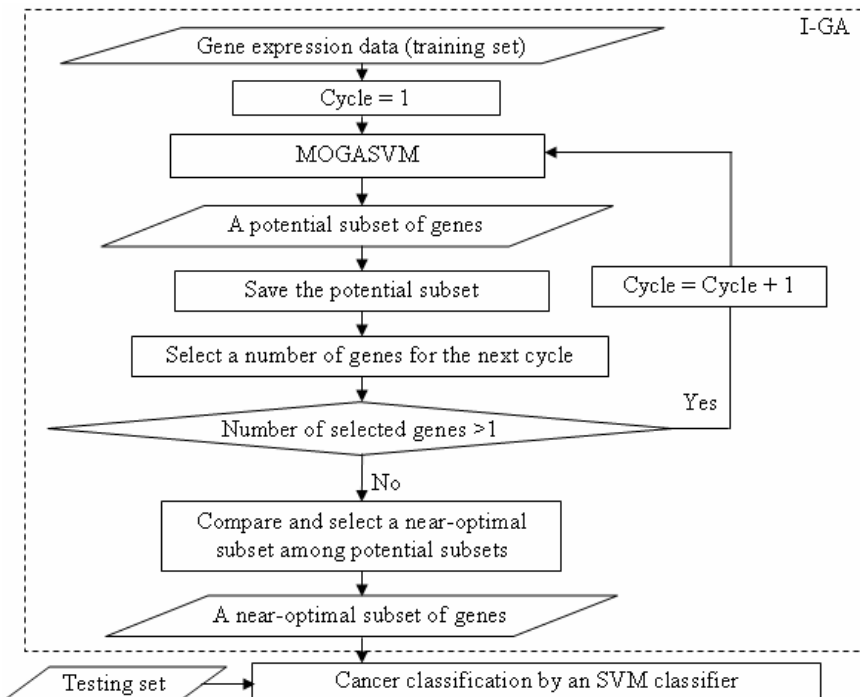


Figure 1 The flowchart of I-GA

I-GA in our work differs from the methods in the previous works in one major part (Li *et al.*, 2008; Mohamad *et al.*, 2005; Mohamad *et al.*, 2009; Peng *et al.*, 2003). The major difference is that our proposed method involves an iterative approach, whereas the previous works did not use any iterative process for gene selection. The general procedure of I-GA is shown in Fig. 1. Basically, I-GA repeats the process of MOGASVM to reduce the dimensionality of data iteratively. The description of each step is explained as follows:

- Step 1: Starting an iterative process. It is repeated until the number of selected genes in the potential subset of the current cycle c is equal or less than 1. Every cycle is started here. In each cycle of I-GA, a number of selected genes are automatically selected by MOGASVM and the dimensionality is iteratively reduced.
- Step 2: Starting MOGASVM to find and produce a potential subset of genes.
- Step 3: Producing and saving the potential subset of selected genes. This potential subset is used for the next cycle (cycle $c+1$) as an input set. The selection of genes in the next cycle (cycle $c+1$) only uses genes in the potential subset that is resulted by the previous cycle (cycle c). Therefore, the dimensionality and complexity of solution spaces can be decreased on a cycle by cycle basis.

- Step 4: A near-optimal subset is selected among the potential subsets based on the highest fitness value (the highest LOOCV accuracy with the smallest number of selected genes).
- Step 5: An iterative process (Steps 1-4) results a near-optimal subset of genes. This subset is possible to be found due to the dimensionality of data has been iteratively reduced. The near-optimal subset is then used to construct an SVM classifier, and the constructed SVM is tested by using the test set.

8.3 EXPERIMENT

8.3.1 Data Sets

The lung microarray data set is used to evaluate I-GA. This data set has two classes: malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA). There are 181 samples (31 MPM and 150 ADCA). The training set contains 32 of them (16 MPM and 16 ADCA). The rest 149 samples are used for the test set. Each sample is described by 12,533 genes. It can be obtained at <http://chest Surg.org/publications/2002-microarray.aspx>.

8.3.3 Experimental Setup

Three criteria following their importance are considered to evaluate the performances of I-GA and other experimental methods: test accuracy, leave-one-out-cross-validation (LOOCV) accuracy, and the number of selected genes. Several experiments

are conducted 10 times on each data set using I-GA and other experimental methods such as GASVM (single-objective), MOGASVM, GASVM version 2 (GASVM-II), and SVM. Next, an average result of the 10 independent runs is obtained. A near-optimal subset that produces the highest classification accuracies with the possible least number of genes is selected as the best subset.

8.3.4 Experimental Results

Table 1 shows the classification accuracy for each run using I-GA. Interestingly, all runs have achieved 100% LOOCV accuracy. This has proven that I-GA has efficiently selected and produced a near-optimal solution in a solution space. This is due to the fact of its ability to automatically reduce the dimensionality and complexity of the solution space on a cycle by cycle basis. Therefore, I-GA yields the near-optimal gene subset (a smaller subset of informative genes with higher classification accuracy) successfully.

Table 1 Classification accuracies for each run using I-GA

Run#	Lung Data Set		
	LOOCV (%)	Test (%)	#Selected Genes
1	100	90.60	2
2	100	95.30	2
3	100	93.29	3
4	100	95.30	4
5	100	85.24	2
6	100	83.22	3
7	100	92.62	2
8	100	97.32	2
9	100	96.64	2
10	100	95.30	3
Average \pm S.D	100 \pm 0	92.48 \pm 4.80	2.5 \pm 0.71

Note: Results of the best subsets shown in shaded cells. S.D. denotes the standard deviation, whereas #Selected Genes represent a number of selected genes.

Informative genes in the best gene subset as produced by the proposed I-GA and reported in Table 1 are listed in Table 2. These informative genes among the thousand of genes may be the excellent candidates for clinical and medical investigations. Biologists can save much time since they can directly refer to the genes that have higher possibility to be useful for cancer diagnosis and drug target in the future.

Table 2 The list of informative genes in the best gene subsets

Data Set	Run#	Probe-set Name	Gene Description
		33328_at	ESTs
Lung	2	609_f_at	Highly similar to SMHU1B metallothionein 1B [H.sapiens]

According to Table 3, generally, I-GA has outperformed the other experimental methods in terms of LOOCV accuracy, test accuracy, and the number of selected genes. The gap between LOOCV accuracy and test accuracy that resulted by I-GA was also lower. This small gap shows that the risk of the over-fitting problem can be reduced. Therefore, I-GA is more efficient than other experimental methods since it has produced the higher classification accuracies, smaller number of selected genes, smaller standard deviations, and smaller gap between LOOCV accuracy and test accuracy. However, due to the iterative process, I-GA is computationally more extensive than other methods.

Table 3 The benchmark of the proposed I-GA with the previous related methods

Method	Lung Data Set (Average \pm S.D; The Best)		
	#Selected Genes	Accuracy (%)	
		LOOCV	Test
I-GA	(2.5 \pm 0.71; 2)	(100 \pm 0; 100)	(92.48 \pm 4.80; 97.32)
GASVM-II (Mohamad <i>et al.</i> , 2005)	(10 \pm 0; 10)	(100 \pm 0; 100)	(59.33 \pm 29.32; 97.32)
MOGASVM (Mohamad <i>et al.</i> , 2009)	(4,418.5 \pm 50.19; 4,433)	(75.31 \pm 0.99; 78.13)	(85.84 \pm 3.97; 93.29)
GASVM (Mohamad <i>et al.</i> , 2005)	(6,267.8 \pm 56.34; 6,342)	(75 \pm 0; 75)	(84.77 \pm 2.53; 87.92)
SVM (Mohamad <i>et al.</i> , 2005)	(12,533 \pm 0; 12,533)	(65.63 \pm 0; 65.63)	(85.91 \pm 0; 85.91)

Note: The best result shown in shaded cells. S.D. denotes the standard deviation, whereas #Selected Genes represent a number of selected genes.

8.5 CONCLUSIONS

In this paper, I-GA has been proposed and tested for gene selection on the lung microarray data set. Based on the experimental results, the performance of I-GA was superior to the other experimental methods and related previous works. This is due to the fact that I-GA can automatically reduce the dimensionality of the data on a cycle by cycle basis. When the dimensionality was reduced, the combination of genes and the complexity of solution spaces can also be automatically decreased iteratively. This iterative process is done to generate potential gene subsets in higher-dimensional data (microarray data), and finally produce a near-optimal subset of informative genes. Hence, the gene selection using I-GA is needed to produce a near-optimal (smaller) subset of informative genes for better cancer classification. Moreover, focusing the attention on the informative genes in the best subset may provide insights into the mechanisms responsible for the cancer itself. Even though I-GA has classified tumours with higher accuracy, it is still not able to completely avoid the over-fitting problem. Therefore, a combination between a constraint approach and a hybrid approach is recently developed to solve the problem.

REFERENCES

- Li, S., Wu, X. and Hu, X. 2008. Gene Selection Using Genetic Algorithm and Support Vectors Machines. *Soft Computing*, 12(7): 693-698.
- Mohamad, M.S., Deris, S. and Illias, R.M. 2005. A Hybrid of Genetic Algorithm and Support Vector Machine for Features Selection and Classification of Gene Expression Microarray. *International Journal of Computational Intelligence and Applications*, 5: 91-107.
- Mohamad, M.S., Omatu, S., Deris, S., Misman, M.F. and Yoshioka, M. 2009. A Multi-objective Strategy in Genetic Algorithms for Gene Selection of Gene Expression Data, *International Journal of Artificial Life & Robotics*, 13(2).
- Peng, S., Xu, Q., Ling, X.B., Peng, X., Du, W. and Chen, L. 2003. Molecular Classification of Cancer Types from Microarray Data Using the Combination of Genetic Algorithms and Support Vector Machines. *FEBS Letters*, 555(2): 358-362.
- Saeys, Y., Inza, I. and Larranaga, P. 2007. A Review of Feature Selection Techniques in Bioinformatics. *Bioinformatics*, 23(19): 2507-2517.

9

AN IMPROVED BINARY PARTICLE SWARM OPTIMIZATION ALGORITHM FOR GENES SELECTION AND CLASSIFICATION OF COLON CANCER DATA

Mohd Saberi Mohamad
Sigeru Omatu
Safaai Deris
Michifumi Yoshioka

9.1 INTRODUCTION

Microarray is a device that can be employed in measuring of expression levels of thousands of genes simultaneously. It finally produces microarray data that contain useful information of genomic, diagnostic, and prognostic for researchers (Knudsen, 2002). Thus, there is a need to select informative genes that contribute to a cancerous state (Mohamad *et al.*, 2009). However, the gene selection process poses a major challenge because of the following characteristics of microarray data: the huge number of genes compared to the small number of samples (higher-dimensional data), irrelevant genes, and noisy data. To overcome this challenge, a gene selection method is used to select a subset of genes that increases the classifier's ability to classify samples more accurately (Mohamad *et al.*, 2007).

Recently, several methods based on particle swarm optimization (PSO) are proposed to select informative genes from microarray data (Chuang *et al.*, 2008; Li *et al.*, 2008; Shen *et al.*, 2008). PSO is a new evolutionary technique proposed by Kennedy and Eberhart (Kennedy and Eberhart, 1995)]. It is motivated from the simulation of social behaviour of organisms such as bird flocking and fish schooling. Shen *et al.* (Shen *et al.* 2008) have proposed a hybrid of PSO and tabu search approaches for gene selection. However, the results obtained by using the hybrid method are less significant because the application of tabu approaches in PSO is unable to search a near-optimal solution in search spaces. Next, an improved binary PSO have been proposed by Chuang *et al.* (Chuang *et al.*, 2008). This approach produced 100% classification accuracy in many data sets, but it used a higher number of selected genes to achieve the higher accuracy. It uses the higher number because of all global best particles are reset to the same position when their fitness values do not change after three consecutive iterations. Li *et al.* (Li *et al.*, 2008) have introduced a hybrid of PSO and GA for the same purpose. Unfortunately, the accuracy result is still not high and many genes are selected for cancer classification since there is no direct probability relation between genetic algorithms (GA) and PSO. Generally, the proposed methods that based on PSO (Chuang *et al.*, 2008; Li *et al.*, 2008; Shen *et al.*, 2008) are intractable to efficiently produce a near-optimal (smaller) subset of informative genes for higher classification accuracy. This is mainly because the total number of genes in microarray data is too large (higher-dimensional data).

9.2 METHOD

9.2.1 A Standard Version of Binary PSO (BPSO)

Binary PSO (BPSO) is initialised with a population of particles. At each iteration, all particles move in a problem space to find the optimal solution. A particle represents a potential solution (gene subset) in an n -dimensional space (Kennedy and Eberhart, 1997). Each particle has position and velocity vectors for directing its movement. The position vector and velocity vector of the i th particle in the n -dimension can be represented as $X_i = (x_i^1, x_i^2, \dots, x_i^n)$ and $V_i = (v_i^1, v_i^2, \dots, v_i^n)$, respectively, where x_i^d is a binary bit, $i=1,2,\dots,m$ (m is the total number of particles); $d=1,2,\dots,n$ (n is the dimension of data).

In gene selection, the vector of particle positions is represented by a binary bit string of length n , where n is the total number of genes. Each vector denotes a gene subset. If the value of the bit is 1, it means that the corresponding gene is selected. Otherwise, the value of 0 means that the corresponding gene is not selected. Each particle in a generation updates its own position and velocity according to the following equations:

$$v_i^d = w * v_i^d + c_1 r_1 * (pbest_i^d - x_i^d) + c_2 r_2 * (gbest^d - x_i^d) \quad (\text{Eq. 1})$$

$$Sig(v_i^d) = \frac{1}{1 + e^{-v_i^d}} \quad (\text{Eq. 2})$$

$$\text{if } Sig(v_i^d) > r_3, \text{ then } x_i^d = 1; \text{ else } x_i^d = 0. \quad (\text{Eq. 3})$$

where w is the inertia weight. c_1 and c_2 are the acceleration constants in the interval $[0,2]$. r_1, r_2 , and r_3 are random values in the range $[0,1]$. $Pbest_i = (pbest_i^1, pbest_i^2, \dots, pbest_i^n)$ and $Gbest = (gbest^1, gbest^2, \dots, gbest^n)$ represent the best previous position of the i th particle and the global best position of the swarm (all particles), respectively. $Sig(v_i^d)$ is a sigmoid function where $Sig(v_i^d) \in [0,1]$.

9.2.2 An Improved Binary PSO (IPSO)

In this paper, we propose IPSO for gene selection. It is introduced to solve the problems derived from the microarray data, overcome the limitation of the related previous works (Chuang *et al.*, 2008; Li *et al.*, 2008; Shen *et al.*, 2008), and inline with the diagnostic goal. IPSO in our work differs from the methods in the previous works in one major part. The major difference is that we modify the existing rule (Eq. 3) for the position update, whereas the previous works used a standard rule (Eq. 3). Firstly, we analyse the sigmoid function (Eq. 2). This function represents a probability for x_i^d to be 0 or 1 ($P(x_i^d = 0)$ or $P(x_i^d = 1)$). It has the properties as follows:

$$\lim_{v_i^d \rightarrow \infty} Sig(v_i^d) = 1 \quad (\text{Eq. 4})$$

$$\lim_{v_i^d \rightarrow -\infty} Sig(v_i^d) = 0 \quad (\text{Eq. 5})$$

$$\text{if } v_i^d = 0 \text{ then } P(x_i^d = 1) = 0.5 \text{ or } \text{Sig}(0) = 0.5 \quad (\text{Eq. 6})$$

$$\text{if } v_i^d < 0 \text{ then } P(x_i^d = 1) < 0.5 \text{ or } \text{Sig}(v_i^d < 0) < 0.5 \quad (\text{Eq. 7})$$

$$\text{if } v_i^d > 0 \text{ then } P(x_i^d = 1) > 0.5 \text{ or } \text{Sig}(v_i^d > 0) > 0.5 \quad (\text{Eq. 8})$$

$$P(x_i^d = 0) = 1 - P(x_i^d = 1) \quad (\text{Eq. 9})$$

Also note that the value of x_i^d can change even if the value of v_i^d does not change, due to the random number r_3 in the Eq. 3. To propose IPSO, the following approaches are suggested:

9.2.3 Modifying the existing rule of position update (Eq. 3)

In order to support the diagnostic goal that needs the least number of genes for accurate cancer classification, the rule of position update is simple modified as follows:

$$\text{if } S(V_i) > r_3, \text{ then } x_i^d = 0; \text{ else } x_i^d = 1 \quad (\text{Eq. 10})$$

The value of particle velocity, V_i in the modified formula (Eq. 10) represents the whole of elements of a particle velocity vector, whereas the standard formula represents a single element. Moreover, V_i is also a positive real number. Based on this positive velocity value, Eq. 2, and Eq. 10, the possibility of $x_i^d = 1$ is too small. This situation causes a smaller number of genes is selected in order to produce a near-optimal gene subset from higher-dimensional data (microarray data).

9.2.4 A Simple Modification of the Formula of Velocity Update (Eq. 1)

In this formula, the calculation of the value of velocity is completely based on the whole of bits of a particle position vector, whereas the original formula (Eq. 1) is based on a single bit.

$$V_i = w * V_i + c_1 r_1 * (Pbest_i - X_i) + c_2 r_2 * (Gbest - X_i) \quad (\text{Eq. 11})$$

9.2.5 Calculation for the distance of two positions

The number of different bits between two particles relates to the difference between their positions. For example, $Gbest = [0011101000]$ and $X_i = [1100110100]$. The difference between $Gbest$ and X_i is $[-1-1110-11-100]$. A value of 1 indicates that compared with the best position, this bit (gene) should be selected, but it is not selected, which may decrease classification quality and lead to a lower fitness value. In contrast, a value of -1 indicates that, compared with the best position, this

bit should not be selected, but it is selected. The selection of irrelevant genes makes the length of the subset longer and leads to a lower fitness value. Assume that the number of 1 is a , whereas the number of -1 is b . We use the absolute value of $(a - b)$, $|a - b|$ to express the distance between two positions. In this example, $|a - b| = |3 - 4| = 1$, so the distance between G_{best} and X_i is $G_{best} - X_i = 1$.

9.2.5 Fitness function

A fitness value of a particle (a gene subset) is calculated as follows:

$$fitness(X_i) = w_1 \times A(X_i) + (w_2(M - R(X_i)) / M) \quad (\text{Eq. 12})$$

in which $A(X_i) \in [0, 1]$ is leave-one-out-cross-validation (LOOCV) accuracy on the training set using the only genes in X_i . This accuracy is provided by support vector machine classifiers (SVM). $R(X_i)$ is the number of selected genes in X_i . M is the total number of genes for each sample in the training set. w_1 and w_2 are two weights.

9.3 EXPERIMENT

9.3.1 Data Sets and Experimental Setup

A real microarray data set is used to evaluate IPSO, namely the colon cancer data set. The colon cancer data set, there are 62 samples. It can be obtained at <http://chest Surg.org/publications/2002-microarray.aspx>.

Firstly, we applied the gain ratio technique to pre-select 500-top-ranked genes. These genes are then used by IPSO in the next process. In this paper, LOOCV is used to measure classification accuracy of a gene subset that produced by IPSO. The implementation of LOOCV is in exactly the same way as did by Chuang *et al.* (Chuang *et al.* 2008). Two criteria following their importance are considered to evaluate the performance of IPSO: LOOCV accuracy and the number of selected genes. A near-optimal subset that produces the highest classification accuracy with the smallest number of genes is selected as the best subset. Several experiments are independently conducted 10 times on each data set using IPSO and the standard version of binary PSO (BPSO). Next, an average result of the 10 independent runs is obtained.

9.3.2 Experimental Results

Based on the standard deviations of classification accuracy and the number of selected genes in Table 1, results that produced by IPSO were nearly consistent on the colon data set. Interestingly, all runs have consistently achieved more than 93% LOOCV accuracy with less than six selected genes. This means that IPSO has efficiently selected and produced a near-optimal gene subset from higher-dimensional data (microarray data).

Table 1 Experimental results for each run using IPSO

Run#	Colon Data Set	
	Classification Accuracy (%)	#Selected Genes
1	93.55	5
2	93.55	5
3	96.77	4
4	93.55	5
5	93.55	4
6	95.16	5
7	93.55	4
8	95.16	4
9	93.55	5
10	93.55	4
Average \pm S.D	94.19 \pm 1.13	4.5 \pm 0.53

Note: Results of the best subsets shown in shaded cells. S.D. denotes the standard deviation, whereas #Selected Genes and Run# represent a number of selected genes and a run number, respectively.

Figure 1 shows that the average of fitness values of IPSO increases dramatically after a few generations. The higher average produces a smaller subset of selected genes with higher classification rate. The condition of velocity that should always be positive real numbers provided in the initialisation method, and the new rule of position update provoke the early convergence of IPSO. In contrast, the average of fitness values of BPSO was no improvement until the last generation.

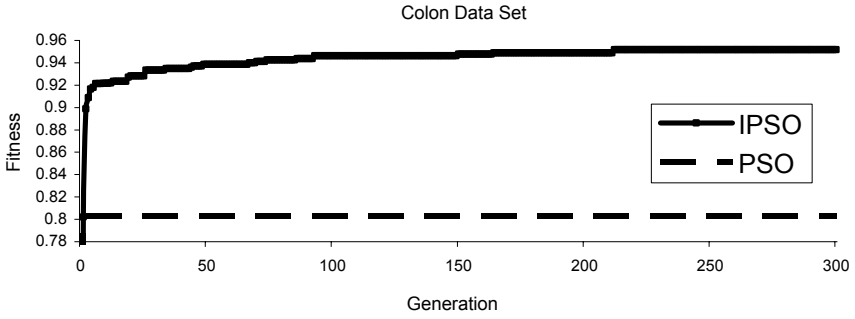


Figure 1 A relation between the average of fitness values (10 runs on average) and the number of generations for IPSO and BPSO

According to the Table 2, overall, it is worthwhile to mention that the classification accuracy and the number of selected genes of IPSO are superior to BPSO in terms of the best, average, and standard deviation results.

Table 2 A comparison in terms of statistical results of the proposed IPSO and BPSO on the colon data set

Method	Classification Accuracy (%)			#Selected Genes		
	The Best	Average	S.D	The Best	Average	S.D
IPSO	96.77	94.19	1.13	4	4.50	0.53
BPSO	87.10	86.94	0.51	214	231	10.19

Note: The best result of each data set shown in shaded cells. S.D. denotes the standard deviation, whereas #Selected Genes represents the number of selected genes.

Table 3 A comparison between our method (IPSO) and other previous methods based on PSO on the colon data set

Method	Classification Accuracy (%)	#Selected Genes
IPSO	(94.19)	(4.50)
PSOGA (Li <i>et al.</i> 2008)	(88.7)	(16.8)
PSOTS (Shen <i>et al.</i> 2008)	(93.55)	(8)

Note: The results of the best subsets shown in shaded cells. ‘-’ means that a result is not reported in the related previous work. A result in ‘()’ denotes an average result. #Selected Genes represents a number of selected genes.

PSOTS = A hybrid of PSO and tabu search

PSOGA = A hybrid of PSO and GA.

For an objective comparison, we compare our work with related previous works that used PSO in their methods (Chuang *et al.*, 2008; Li *et al.*, 2008; Shen *et al.*, 2008). It is shown in Table 3. The averages of LOOCV accuracy and the number of selected genes of our work were 94.19% and 4.5 selected genes, respectively. The latest previous work also came up with the a higher LOOCV result, but they used more than eight genes to obtain the result (Shen *et al.*, 2008). Overall, this work has outperformed the related previous works in terms of LOOCV accuracy and the number of selected genes.

According to Fig. 1 and Tables 1-3, IPSO is reliable for gene selection since it has produced the near-optimal solution from microarray data. This is due to the modification of position update that causes the selection of a smaller number of genes. Therefore, IPSO yields the optimal gene subset (a smaller subset of informative genes with higher classification accuracy) for colon cancer classification.

9.4 CONCLUSIONS

In this paper, IPSO has been proposed and tested for gene selection on the colon microarray data set. Based on the experimental results, the performance of IPSO was superior to the standard version of binary PSO and related previous works. This is due to the fact that the modified rule of position update in IPSO causes a smaller number of genes is selected in each iterative, and finally produce a near-optimal subset of genes for better cancer classification. For future works, a combination between a constraint approach and PSO is proposed to increase the classification accuracy.

REFERENCES

- Chuang, L.Y., Chang, H.W., Tu, C.J. and Yang, C.H. 2008. Improved Binary PSO for Feature Selection Using Gene Expression Data. *Computational Biology & Chemistry*, 32(1):29-38.
- Li, S., Wu, X. and Tan, M. 2008. Gene Selection Using Hybrid Particle Swarm Optimization and Genetic Algorithm, *Soft Computing*, 12(11): 1039-1048.
- Kennedy, J. and Eberhart, R. 1995. Particle Swarm Optimization. *Proceedings of the 1995 IEEE International Conference on Neural Networks*, 4, pp.1942-1948.
- Kennedy, J. and Eberhart, R. 1997. A Discrete Binary Version of the Particle Swarm Algorithm. *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics*, 5, pp.4104-4108.
- Knudsen, S. 2002. A Biologist's Guide to Analysis of DNA Microarray Data. *John Wiley & Sons*, New York.
- Mohamad, M.S., Omatu, S., Deris, S. and Hashim, S.Z.M. 2007. A Model for Gene Selection and Classification of Gene Expression Data. *International Journal of Artificial Life & Robotics*, 11(2): 219-222.
- Mohamad, M.S., Omatu, S., Deris, S., Misman, M.F. and Yoshioka, M. 2009. Selecting Informative Genes from Microarray Data by Using Hybrid Methods for Cancer Classification", *International Journal of Artificial Life & Robotics*, 13(2).
- Shen, Q., Shi, W.M. and Kong, W. 2008. Hybrid Particle Swarm Optimization and Tabu Search Approach for Selecting Genes for Tumor Classification Using Gene Expression Data. *Computational Biology & Chemistry*, 32(1): 53-60.

10

TEXT MODELING APPROACH FOR PROTEIN 3D-STRUCTURE SIMILARITY MEASUREMENT

Jafar Razmara
Safaai Deris

10.1 INTRODUCTION

Protein structure similarity measure is a very important tool to highlight the similarities and differences between protein structures. It has wide applications in protein structure analysis and classification, structure-based drug design, phylogenetic analysis and protein structure predictions which have attracted tremendous attention and have been broadly studied within the past decade. It is known that the protein structure highly indicates its functionality and the potential interactions with the other protein structures. For two given proteins, if the sequences are similar then their evolutionary relationship is obvious. Otherwise, the 3D-structure of proteins, due to structural and functional behavior placed on them, are much more evident than protein sequences. As a result, the structural similarity of proteins should be used to distinguish the differences among various proteins functionalities and their evolutionary relationships. Whenever a new protein structure is discovered, it is desired to find the structural similar proteins to predict its functions and properties.

The protein structure similarity measurement has two main problems: *Complexity* and *Curse of dimensionality* (Aghili *et al.*,

2005). In the view of complexity, the structure comparison problem is NP-hard and no exact solution exists for structural alignment of proteins. There are several algorithms proposed for optimizing the results, however, none of them can guarantee optimality within any given precision. Rapidly growing of the number of discovered protein structures, also, provides the dimensionality problem. The Protein Data Bank (PDB), currently, contains 54,956 known protein structure. The increasing number of entries in the PDB requires more efficient methods to search and find similar structural proteins.

Many protein structure comparison, retrieval and classification methods have been proposed that are divided into two main categories; sequence comparison and 3D structure comparison (Ohkawa *et al.*, 1999). The former can be considered as a sequence alignment problem of amino acids in the primary structure of the proteins. The latter is structure matching process based on three-dimensional structure of the proteins. The main goal of protein structure comparison is to superimpose two proteins over the maximum number of residues (amino acids) with a minimal distance between the matched pairs.

Several approaches to protein structure alignment have been explored over the past decade. The proposed techniques can be categorized into fine-grain and coarse-grain approaches (Chionh *et al.*, 2003). Fine-grain approaches, firstly, operate at the SSE level, and then align two proteins in amino acid level for detailed alignment. Examples of these approaches include comparison of distance matrices (DALI) (Holm and Sander, 1993), vector alignment of SSEs (VAST) (Gibrat *et al.*, 1997), combinatorial extension of alignment path (CE) (Shindyalov and Bourne, 1998) and Secondary Structure Matching (SSM) (Krissinel and Henrick, 2004). The methods of fine-grain approach usually have high accuracy but they are slow. Coarse-grain approaches, apply only SSEs as the basic elements. TOPSCAN (Martin, 2000) and SCALE (Chionh *et al.*, 2003) are two examples of these approaches that are much faster but less accurate.

Despite the maturity of the proposed methods, the study for

designing new similarity measures is still an active research area. Due to the continuous growth of protein databases and discover of new unknown proteins, the interest is renewed for designing alternative effective and reliable algorithms. Furthermore, another motivation of equal importance for establishment of similarity measure is proposition of a method without need to parameter setting by the user. The classical similarity approaches such as dynamic programming often needs a set of optional parameters to reach the best possible similarity.

Language modeling and its algorithms is a hybrid research area in protein structure analysis. The amino acid sequence of a protein consists of 20 distinct symbols of alphabet that can be treated as text written in a universal language. The mapping of a protein sequence to its structure, functional and biological role is similar to the mapping of words to their semantic meaning in natural languages. Recently (Biological Language Conference, 2003), it was suggested that this similarity motivates to apply *statistical language modeling* and *text classification techniques* in biological sequences analyzing. Within this hybrid research area, it is believed that the identification of Grammar/Syntax rules could reveal entities/relations of biological and medical sciences (Bogan-Marta *et al.*, 2005).

Here a novel method for protein structural similarity measurement based on n -gram text modeling is proposed. The method is inspired by the successful use of entropy concept for information retrieval in the field of statistical language modeling (Young and Bloothoof, 1997, Manning, 2000). N -gram modeling also stands out as superior to any formal linguistics approach and has gained high popularity due to its simplicity (Bogan-Marta *et al.*, 2005). In a very first attempt to fuse theoretical concepts from computational linguistics within the field of bioinformatics, a new general strategy for measuring similarity between primary sequences of proteins was introduced (Bogan-Marta *et al.*, 2005). In this strategy, specifically, n -gram modeling is first applied to each protein sequence and cross-entropy measures are then employed to compare pairs of proteins. Based on the fruitful

results of this attempt in using n -gram modeling, we now extend this approach to protein structural similarity measurement.

The rest of this paper is organized as follows. The next section, describes the protein structure representation in sequence form. In section 3, the n -gram modeling technique is discussed. Section 4 introduces a superposition task to find an overlap between two protein structures. Section 5 describes the novel method for protein structural similarity measurement based on n -gram modeling. Finally, the experiments results are represented and discussed in section 6.

10.2 PROTEIN STRUCTURE MODELING IN STRING FORM

Various kinds of language models can be used to capture different aspects of regularities of natural language. A variety of these alternative methods has already used for expressing similarity between biological sequences. Development of the language models to measure structural similarity of proteins needs protein 3D structure modeling in string form.

There are various databases containing structure details of proteins. The Protein Data Bank (PDB) is the worldwide repository for the processing and distribution of three dimensional biological molecular structure data. From the PDB file of each protein, the position of each residue in 3D space can be extracted using the 3D coordinates of C_α atom of each amino acid. Hence the 3D structure of a protein can be modeled in a sequence form by labeling the position of each residue with respect to the position of its previous residue in 3D coordinate. For labeling each residue i , let us suppose that the position of residue $i-1$ is centered at the origin of the spatial coordinate. Thus the position of the residue i

can be labeled according to its spatial coordinates and can be represented with a specially defined alphabet. Figure 1 shows labels defined for 18 different positions of residue i with respect to residue $i-1$. To prevent the ambiguity, the other 8 labels are not shown in the figure. Table 1 represents 26 letters used for 26 position states in spatial coordinate corresponding to its previous residue. In this table, all lengths are expressed in Angstrom.

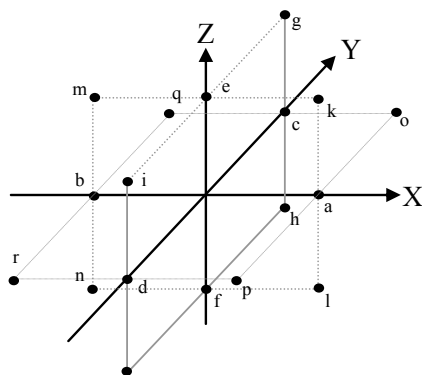


Figure 1 3D-space and labels defined for different position of residue i with respect to residue $i-1$ in the origin of the coordinate

Accordingly, the protein structure can be represented in two strings sequences: the first string represents amino acids sequence and the second string represents the position label of

each amino acid, according to table 1. From now onwards, we call the second sequence as relative residue position sequence. Figure2 represents the two sequences extracted for 1CRB chain. Having reduced the protein structure to a sequence of characters, we can apply language modeling techniques in protein structure similarity measurement problem.

```
1  PVDFNGYWKM  LSNENFEEYL  RALDVNVALR  KIANLLKPKD  EIVQDGDHMI
   zwtwxsugu  yuauktspjt  kvhsqsmqzy  wxzywxzlvz  ximieuvohh

51  IRTLSTFRNY  IMDFQVGKEF  EEDLTGIDDR  KCMTVSWDG  DKLQCVQKGE
   hkwsucvzvz  imuyzustot  xtnowiptvj  ryzynwxhqz  uvspovssy

101  KEGRGWTQWI  EGDELHLEMR  AEGVTCKQVF  KKVH
   yrxxmzxrqy  xckluououo  uywnqrnxnx  xqh
```

Figure 2 Two sequences extracted for the 1CRB protein chain

Table 1 Letters defined for labeling 3D position of each residue with respect to its previous residue.

$((x_2, y_2, z_2)$ is the position of current residue and
 (x_1, y_1, z_1) is the position of previous residue)

Conditions for x, y, z	Symbol	Conditions for x, y, z	Symbol
$x_2 - x_1 > 0, y_2 - y_1 < 1, z_2 - z_1 < 1$	'a'	$x_2 - x_1 < 0, y_2 - y_1 < 1, z_2 - z_1 < 0$	'n'
$x_2 - x_1 < 0, y_2 - y_1 < 1, z_2 - z_1 < 1$	'b'	$x_2 - x_1 > 0, y_2 - y_1 > 0, z_2 - z_1 < 1$	'o'
$ x_2 - x_1 < 1, y_2 - y_1 > 0, z_2 - z_1 < 1$	'c'	$x_2 - x_1 > 0, y_2 - y_1 < 0, z_2 - z_1 < 1$	'p'
$ x_2 - x_1 < 1, y_2 - y_1 < 0, z_2 - z_1 < 1$	'd'	$x_2 - x_1 < 0, y_2 - y_1 > 0, z_2 - z_1 < 1$	'q'
$ x_2 - x_1 < 1, y_2 - y_1 < 1, z_2 - z_1 > 0$	'e'	$x_2 - x_1 < 0, y_2 - y_1 < 0, z_2 - z_1 < 1$	'r'
$ x_2 - x_1 < 1, y_2 - y_1 < 1, z_2 - z_1 < 0$	'f'	$x_2 - x_1 > 0, y_2 - y_1 > 0, z_2 - z_1 > 0$'s'
$ x_2 - x_1 < 1, y_2 - y_1 > 0, z_2 - z_1 > 0$	'g'	$x_2 - x_1 > 0, y_2 - y_1 > 0, z_2 - z_1 < 0$	't'
$ x_2 - x_1 < 1, y_2 - y_1 > 0, z_2 - z_1 < 0$	'h'	$x_2 - x_1 > 0, y_2 - y_1 < 0, z_2 - z_1 > 0$	'u'
$ x_2 - x_1 < 1, y_2 - y_1 < 0, z_2 - z_1 > 0$	'i'	$x_2 - x_1 > 0, y_2 - y_1 < 0, z_2 - z_1 < 0$	'v'
$ x_2 - x_1 < 1, y_2 - y_1 < 0, z_2 - z_1 < 0$	'j'	$x_2 - x_1 < 0, y_2 - y_1 > 0, z_2 - z_1 > 0$	'w'
$x_2 - x_1 > 0, y_2 - y_1 < 1, z_2 - z_1 > 0$	'k'	$x_2 - x_1 < 0, y_2 - y_1 > 0, z_2 - z_1 < 0$	'x'
$x_2 - x_1 > 0, y_2 - y_1 < 1, z_2 - z_1 < 0$	'l'	$x_2 - x_1 < 0, y_2 - y_1 < 0, z_2 - z_1 > 0$	'y'
$x_2 - x_1 < 0, y_2 - y_1 < 1, z_2 - z_1 > 0$	'm'	$x_2 - x_1 < 0, y_2 - y_1 < 0, z_2 - z_1 < 0$	'z'

10.3 TEXT SIMILARITY MEASUREMENT USING N-GRAM MODELING

Several kinds of language modeling techniques have been developed to capture different aspects of regularities of textual data. Markov chains are the more fundamental concept used in language modeling. In this approach, the dependency of the conditional probability of existing words W_k at a position k in a given text is depended only upon its immediate n predecessor words W_{k-n}, \dots, W_{k-1} . The resulting stochastic models, usually called as n -grams, widely used in formal linguistic approaches and has gained high popularity due to its simplicity (Bogan-Marta *et al.*, 2005). *Entropy* is also a useful concept in the quantification of information in a textual sequence and making connection with probabilistic language modeling. It can also be applied for

obtaining how much information is extracted by a special grammar, how a grammar matches a language, etc. A specific definition of entropy as described in (Bogan-Marta *et al.*, 2005), when a written word sequence $W = \{W_1, W_2, \dots, W_k, \dots\}$ is treated as an n -gram, is represented in the following formula:

$$\begin{aligned} H(X) &= -\sum_{w_i^n} p(w_i^n) \log_2 p(w_{i+n}|w_i^{n-1}) \\ &= -(1/N) \sum_{w_i^n} \text{Count}(w_i^n) \log_2 p(w_{i+n}|w_i^{n-1}) \end{aligned} \quad (\text{Eq. 1})$$

where the variable X is the n -gram $w_i^n = \{w_i, w_{i+1}, \dots, w_{i+n-1}\}$, the summation runs over all the possible n -length combinations of consecutive w_i , (i.e. $W^* = \{\{w_1, w_2, \dots, w_n\}, \{w_2, w_3, \dots, w_{n+1}\}, \dots\}$), $\text{Count}(w_i^n)$ is the number of occurrences of n -gram w_i^n and N is the total number of n -grams in the sequence. The second term in the summation is the conditional probability that relates the n -th element of an n -gram with the preceding $n-1$ elements and can be estimated by a counting procedure.

$$P(w_{i+n}|w_i^{n-1}) = \text{Count}(w_{i+n}) / \text{Count}(w_i^{n-1}) \quad (\text{Eq. 2})$$

As described in (Bogan-Marta *et al.*, 2005), the above entropy estimation indicates how a specific protein sequence is well predicted by the corresponding model. In the similarity measuring task, the direct comparison of the two proteins could not be facilitated by applying this measure to two distinct proteins. Cross-entropy measure is the relevant tool for this kind of comparison, where the n -gram model is, first, built based on the word-counts of one protein sequence and then the predictability, of the second sequence, by the model is measured via the formula:

$$H(X, P_M) = -\sum_{all\ w} p(w_i^n) \log_2 P_M(w_{i+n}|w_i^{n-1}) \quad (\text{Eq. 3})$$

The term $p(w_i^n)$ refers to the reference protein sequence and results from counting the words of that specific protein. The term $P_M(w_{i+n}|w_i^{n-1})$ refers to the sequence which the model has to be estimated (it results from counting the words of this protein). Variable X ranges over all the n -grams of the reference protein sequence (Bogan-Marta *et al.*, 2005).

The crux of the applied method in (Bogan-Marta *et al.*, 2005) is that both the unknown query-protein and each protein in a given database are represented via n -gram model and the cross-entropy measure is utilized to compare their representations. *Direct* method, a typical implementation of this idea, firstly, computes the perfect score PS from (3) using the query-protein both as reference and model sequence. Then the method uses (3) in the computation of the similarity score between the query-protein as the reference protein and each protein from the database as the model sequence. Therefore, N similarities are computed and applied in the calculation of the absolute differences via the formula:

$$4) \quad D(S_q, S_i) = |H(X_q, P_{M_i}) - PS| \quad (\text{Eq. 4})$$

Finally, the most similar protein in the database to the query-protein is easily identified as the one having the lowest $D(S_q, S_i)$. In another implementation of the idea, called *Alternating* method, the only difference with respect to the *Direct* method is that the protein with the shortest sequence plays the role of reference sequence when comparing the query protein with each database-protein. This was devised in order to cope with the more different length of the proteins to be compared.

10.4 SECONDARY STRUCTURE SUPERPOSITION

The application of any one of the structural alignment algorithms requires protein structure representation in some coordinate independent space to make structures comparable. One possible representation is the so-called distance matrix, which is a two-dimensional matrix containing all pairwise distances between all C_α atoms of the protein backbone (Chi *et al.*, 2004). This can also be represented as a set of overlapping sub-matrices spanning only fragments of the protein. Another possible representation is the reduction of the protein structure to the level of secondary structure elements (SSEs), which can be represented as vectors and can carry additional information about relationships to other SSEs, as well as about certain biophysical properties (Singh and Brutlag, 1997, Krissinel and Henrick, 2004, Martin, 2000). In the case of distance matrix representation, the comparison algorithm breaks down the distance matrices into regions of overlap, which are then again combined if there is overlap between adjacent fragments, thereby extending the alignment. If the SSE representation is chosen, there are several possibilities. One can search for the maximum ensemble of equivalent SSE pairs using algorithms to solve the maximum clique problem from graph theory. Other approaches employ dynamic programming or combinatorial simulated annealing (Krissinel and Henrick, 2004).

The proposed method in this paper needs an initial superposition between two proteins before encoding their structure in sequence form. In this way, the method represents the secondary structure elements of proteins as vectors and obtains a match for aligned vector pairs of query and reference proteins by computing angles between them and rotating reference protein in 3D coordinates. The secondary structures that represented in vector form are α -helices and β -strands and all types of helices (α , π , 3-10, and left handed helices) are grouped together in one class. It can easily be altered to use special classes for each type of helix. The SSEs information can be extracted from PDB file of each

protein. Following equations are used to compute the beginning and end points of the helix and strand vectors respectively where indices i and j denote the first and last residues in the SSE (Singh and Brutlag, 1997), (Krissinel and Henrick, 2004):

$$\begin{aligned} r_b &= (0.74r_i + r_{i+1} + r_{i+2} + 0.74r_{i+3}) / 3.48, \\ r_e &= (0.74r_{j-3} + r_{j-2} + r_{j-1} + 0.74r_j) / 3.48, \end{aligned} \quad (\text{Eq. 5})$$

$$\begin{aligned} r_b &= (r_i + r_{i+1}) / 2, \\ r_e &= (r_{j-1} + r_j) / 2 \end{aligned} \quad (\text{Eq. 6})$$

and then the SSEs are represented by the vectors $r_{SSE} = r_b - r_e$. Helices of length shorter than five residues and strands of length shorter than three residues are neglected (Singh and Brutlag, 1997), (Krissinel and Henrick, 2004).

Having reduced the two query and reference proteins to a set of either *Helix* or *Strand* vectors, the method now uses a dynamic programming algorithm to compare these two sets of vectors and find the best matched pairs. The scoring functions used in the algorithm are applied on the SSE type of vector, order of the vector in the protein and angles between matched vectors in 3D coordinates.

Finally, the method computes angles between each pair of matched vectors of query and reference protein and achieves a rotation angle and direction in polar coordinates. Hence, a relevant rotation-translation matrix is produced to achieve an initial overlap between two query and reference protein.

10.5 STRUCTURAL SIMILARITY MEASUREMENT USING TEXT MODELING APPROACH

Having introduced the above procedure for protein structure modeling in string form and the previous activities on n -gram modeling, a new approach for 3D-structure of proteins similarity measurement is proposed. This method works based on the above n -gram similarity measure over protein structure modeled in sequence form as discussed in section 2. The similarity measurement process uses cross-entropy formula to compute the absolute entropy (4) between each pair of query and reference protein relative residue position sequences and find the most structural similar protein in the given database to the query-protein.

In this new approach, a modification to the n -gram method introduced in (Bogan-Marta *et al.*, 2005) is done. In the counting process of the n -gram method described in (Bogan-Marta *et al.*, 2005), when all of the words have been counted once, the probability by $P_M(w_{i+n}|w_i^{n-1})$ become zero, creating problems in the calculation of $H(X, P_M)$. The new method uses a corrected entropy measurement formula:

$$H(X, P_M) = -\sum_{all\ w} p(w_i^n) \log_2 (2 + P_M(w_{i+n}|w_i^{n-1})) \quad (\text{Eq. 7})$$

Thus, if the estimated term $P_M(w_{i+n}|w_i^{n-1})$ is zero, the result of logarithm function will be 1 and the value of $p(w_i^n)$ term will be considered in the summation formula.

The procedure described above for similarity measurement has been implemented in the following steps:

- 1) Compute the cross-entropy from (7) for the relative residue position sequence of query-protein.

- 2) For each reference protein in the given database, apply steps 2-1, 2-2 and 2-3.
 - 2-1) Find the matched pairs of SSE vectors with query protein and compute the rotation-translation matrix as discussed in Section 4. Then, rotate and translate the reference protein to extract the new coordinates of atoms. Then, make the relative residue position sequence of protein as described in section 2.
 - 2-2) Apply the cross-entropy measure from the (7) to compute the absolute differences via (4), as discussed in section 3.
 - 2-3) For every atom in the query protein, find the nearest atom (within a threshold distance) on the reference protein and transform the query protein to minimize the RMSD between these pairs of atoms.
- 3) Therefore an array of N extracted similarity is created, where each element of the array contains $D_t(S_q, S_i)$ computed via (4) for the relative residue position sequence. Arrange the array according to D_t .

The input of the algorithm is the unknown query-protein structure modeled in sequence form and a protein database contains the PDB file of each protein. Furthermore, the secondary structure of each protein is represented in collection of some vectors as described in section 4 and used as the input.

10.6 EXPERIMENTAL RESULTS

In order to assess the accuracy and efficiency of the proposed method, some experiments were performed. Firstly, to measure the accuracy of the method, 53 proteins are selected from the SCOP database belonging to All Alpha, All Beta, Alpha and Beta and Alpha+Beta categories with less than 40% sequence identity, having more than 7 SSEs. The selected proteins are

shown in Table 2. The 3D structure of each selected protein is modeled by the two sequences and vector representation of its secondary structure elements, as described above.

Figure 3 represents the matrices containing all the measured dissimilarities $D(S_i, S_j)$, $i, j = 1, 2, \dots, N$ for each pair of proteins i, j in the database as grey scale images for the *Direct* and *Alternating* methods of three different n -gram models. In the figure, the first and second sequence indicates primary sequence and relative residue position sequence. In each matrix the vertical and horizontal edges represent the query and reference proteins respectively. The white and black colors in the output matrices correspond to the maximum and minimum distances between each pair of proteins. As described in (Bogan-Marta *et al.*, 2005), the ideal spatial outlay is a white matrix with only a black diagonal segment. Therefore, it is clearly evident from figure3 that 4-gram modeling which uses *Alternating* Method has a better performance in order to distinguish similar and dissimilar proteins. On the other hand, as seen from the figure, 3-gram modeling outputs represent highly similar, less similar and dissimilar proteins and it is much more informative than 4-gram. Furthermore, figure 3 shows that the results obtained from second sequence are more informative on similarity measurement than the primary sequence.

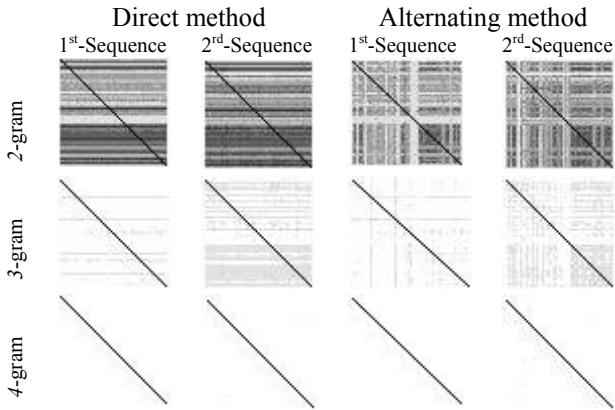


Figure 3 Gray-scale representation of the output D_p and D_t matrices containing all the possible pairwise dissimilarities for 53 proteins in the database using *Direct* and *Alternating* method

Table 2 Dataset used (53 proteins)

PDB Code	All Alpha (12 protein)
1eca, 2hbg, 2lhb 1bbh 1rtp, 1scm, 2sas, 2scp 2gst, 1glp 1cpt, 1phg	Globin-like Ferritin-like EF Hand-Like Gluthathione S-transferases, C-terminal domain Cytochrome P450
PDB Code	All Beta (16 protein)
1cd8, 1cid, 1tlk, 1cfb, 2mcm 2cas 1tie, 1hce 1arb, 2sga, 4sgb, 3rp2 1hbq, 1ftp, 1icn, 1crb	Immunoglobulin-like beta-sandwich Viral coat and capsid proteins beta-Trefoil Trypsin-like serine proteases Lipocalins
PDB Code	Alpha&Beta (21 protein)
1byb, 1ghr, 2acq, 2mnr, 4enl 3cox 3chy, 2fcr, 2fx2 1ldm 1ede, 1tca, 3tgl 5p21 2ctc, 1amp 1gca, 2lbp 1omp 3cla, 1eaf	beta/alpha (TIM)-barrel FAD/NAD(P)-binding domain Flavodoxin-like NAD(P)-binding, Rossmann-fold domains alpha/beta-Hydrolases P-loop containing small nucleotide triphosphate hydrolases Phosphorylase/hydrolase-like Periplasmic binding protein-like I Periplasmic binding protein-like II CoA-dependent acetyltransferases
PDB Code	Alpha+Beta (4 protein)
7rsa, 1onc 1frd 2pnb	Ribonuclease A-like beta-Grasp SH2-like

In order to compare the accuracy and efficiency of the method with other publicly available protein structure similarity servers, two servers were selected, namely Combinatorial Extension (CE) and Secondary Structure Matching (SSM). It is believed that none of the scores provides an absolutely reliable measure of structural similarity or statistical significance, and therefore the final decision of accepting a match should be reserved for the user (Krissinel and Henrick, 2004). Hereby, the comparison process is done by calculating three values: RMSD, N_{align} and Q -score. An intuitive understanding of structural similarity suggests contradictory requirements of achieving a lower RMSD and a higher number of aligned residues N_{align} . This contradiction may be eliminated, in the first approximation, by a score that represents a ratio of N_{align} and the RMSD. Therefore, the following function is suggested (Krissinel and Henrick, 2004):

$$Q = N_{align}^2 / ((1 + (RMSD/R_0)^2) N_1 N_2) \quad (\text{Eq. 8})$$

R_0 is an empirical parameter (chosen at 3 Å) that measures the relative significance of RMSD and N_{align} . N_1 and N_2 are the number of residues in the aligned structures. As seen from the above formula, Q reaches 1 only for identical structures ($N_{align}=N_1=N_2$ and R.M.S.D=0), and decreases to zero with decreasing similarity (increasing RMSD or/and decreasing N_{align}). Therefore, the higher Q , is the better, in general, the alignment (Krissinel and Henrick, 2004).

Figure 4 represents the results of comparing the n -gram based method with SSM and CE methods for the example of protein chain 1sar:A. The experiment is done over whole PDB chains by SSM and CE servers in order to select the top 200 chains from the list and use them to do the same experiments applying the n -gram method. The output results in figure4 are represented for 150 protein chains ordered by entropy measure of n -gram method.

Figure4 shows that n -gram method approximately fully agrees with the other servers in the identification of highly similar, less similar and dissimilar structures. As seen from the figure, all the methods reveal the same RMSD results for the first 30 protein chains, but for the rest of the protein chains there are differences. The differences are because the SSM and CE methods apply some iteration tasks to reduce RMSD value, whereas the n -gram method does not perform such a task. RMSD reduction task is a time consuming process. The n -gram method, simply, rotates and translates the reference protein in 3D-coordinates to achieve a superposition with the query protein. Therefore, from the viewpoint of running speed, the similarity measurement process has been accelerated in the n -gram method.

The alignment length of n -gram method, represented in figure4, is approximately the same as SSM. As it is described in (Krissinel and Henrick, 2004), longer alignments always come at the expense of higher RMSD and therefore the observed differences between the servers should be mostly due to the different criteria employed to balance these characteristics.

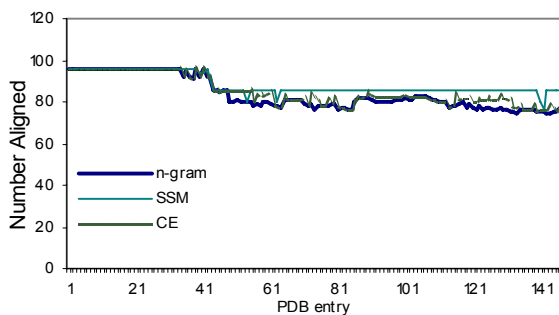
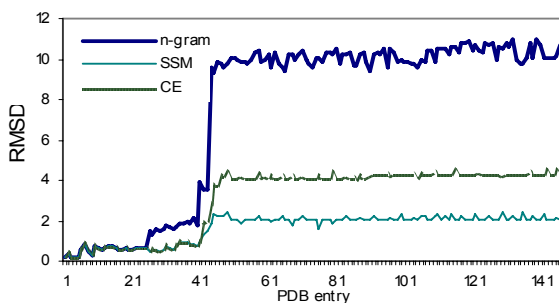
The Q -score is an indication of the balance of RMSD and the alignment length (Krissinel and Henrick, 2004). As seen from the Q -score plot in figure4, Q -score of the n -gram method is lower than those of the two other methods. This is because the n -gram method computes high RMSD value compared with the other methods.

We also performed a comparison between entropy measure computed by the n -gram method via (4) and RMSD computed by the SSM method. Figure5 represents that RMSD value increases with the increasing value of entropy. It shows that the similarity measurement results produced by the n -gram method are approximately the same as those produced by the SSM method. Therefore, the entropy measure based on n -gram modeling is a novel efficient tool for protein structural similarity measurement.

We performed a comparative study, similar to that described above, for a number of structures belonging to different

protein folds. The results represent that the outputs showed in figures 4 and 5 are of a common nature.

To evaluate the efficiency of n -gram method, an extended dataset of about 2000 proteins was prepared from the various categories in the SCOP database. The algorithm of n -gram method is implemented in C++ programming language and done on Pentium IV 2.8GHz machine with 512MB RAM running Windows- XP. Average time of similarity measurement for each query is about 30 seconds. Because the source code of SSM method was not accessible, run-time comparison of two methods could not be conceived. However, including related experiments (Singh and Brutlag, 1997), (Krissinel and Henrick, 2004), (Martin, 2000), (Aung and Tan, 2004), the efficiency of the method is established compared with the other similar methods.



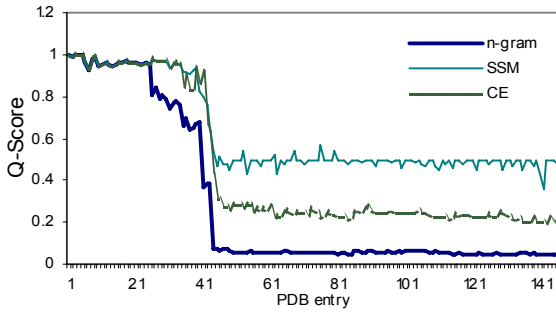


Figure 4 Comparison of the n -gram based method results with SSM and CE methods. PDB chain 1sar:A was used as a query protein for screening the whole PDB. Results in axes x were ordered by entropy value computed by n -gram method.

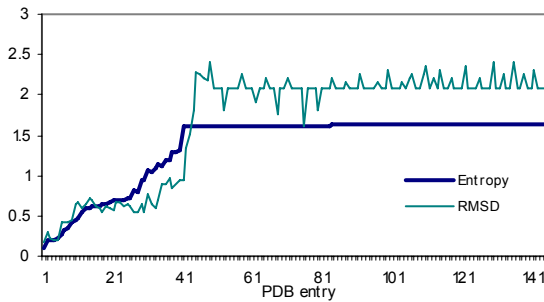


Figure 5 Comparison of the entropy measure of the n -gram method and the RMSD value of SSM method. Results in axes x were ordered by entropy value.

The proposed method in this paper uses the introduced method in (Bogan-Marta *et al.*, 2005) to apply entropy concept for information retrieval in the field of statistical language modeling for measuring the structural similarity of proteins. Specifically, the studied method, simply, applies a superposition task to achieve an initial overlap between the secondary structure elements of two proteins and then, creates relative residue position sequence for them and uses cross-entropy measure over n -gram model to compare their structures. In order to confirm the validity of the proposed method, some experiments on similar protein retrieval methods were performed which demonstrates the applicability and efficiency of this method. Also, the results of experiments represent the method is comparable with the publicly available web servers namely SSM and CE. Moreover regarding the conceptual simplicity of the approach, the preference and applicability of the method to other applied techniques is indicated.

REFERENCES

- Aghili, S. A., Agrawal, D. and Abbadi, A. E. 2005. PADS: Protein Structure Alignment using Directional Shape Signature. *Database Systems for Advanced Applications*.
- Aung, Z. and Tan, K.L. 2004. Automatic Protein Structure Classification through Structural Fingerprinting. *Proc. of the 4th IEEE Symp. on BIBE*.
- Bogan-Marta, A., Laskaris, N., Gavrielides, M.A., Pitas, I. and Lyroutdia, K. 2005. A Novel Efficient Protein Similarity Measure Based on n -gram Modeling. *Proc. of the 2nd Int. Conf. on CIMED*.

- Chi, P., Scott, G. and Shyu, C.R. 2004. A Fast Protein Structure Retrieval System Using Image-Based Distance Matrices and Multidimensional Index. *Proc. of the 4th IEEE Symp. on BIBE*.
- Chionh, C. H., Huang, Z., Tan, K. L. and Yao, Z. 2003. Augmenting SSEs with Structural Properties for Rapid Protein Structure Comparison. *Proc. of the 3rd IEEE Symp. on BIBE*.
- Gibrat, J.F., Madej, T., Spouge, J.L. and Bryant, S.H. 1997. The VAST protein structure comparison method. *Biophysical Journal*, 72:Pt2, pMP298.
- Holm, L. and Sander, C. 1993. Protein structure comparison by alignment of distance matrices. *J. of Molecular Biology*, 233, pp.123–138.
- Krissinel, E. and Henrick, K. 2004. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica Section D: Biological Crystallography*, Vol.60, No.1.
- Manning, C.D. and Schütze, H. 2000. Foundations of statistical natural language processing. *Massachusetts Institute of Technology Press*. 554 – 556;557 – 588.
- Martin, A.C.R. 2000. The Ups and Downs of Protein Topology; Rapid Comparison of Protein Structure. *Protein Engineering*, Vol.13, No.12.
- Ohkawa, T., Hirayama, S. and Nakamura, H. 1999. A Method of Comparing Protein Structures Based on Matrix Representation of Secondary Structure Pairwise Topology. *Proc. of the IEEE Int. Conf. on Information Intelligence and Systems*.
- Shindyalov, I.N. and Bourne, P.E. 1998. Protein Structure Alignment by Incremental Combinatorial Extension (CE) of the Optimal Path. *Protein Engineering*, Vol.11, pp.739-747.
- Singh, A. P. and Brutlag, D. L. 1997. Hierarchical Protein Structure Superposition using both Secondary Structure and Atomic Representations. *Proc. of the 5th Int. Conf. on*

Intelligent Systems for Molecular Biology.

Young, S. and Bloothoof, G. 1997. *Corpus-Based Methods in Language and Speech Processing*. Kluwer Academic Publishers.

11

DNA-CHIPS DATA PROCESSING AND ANALYSIS

Edin Tankovics
Ito Wasito

11.1 INTRODUCTION

Modern genetic together with bioinformatics seeks to understand the function of genes, including more than 40,000 genes in the human genome. Recently developed for genome analysis, DNA microarrays, also known as gene arrays or gene chips, are capable of determining the gene expression levels of thousands of genes simultaneously. Gene expression (often simply expression) is the process by which a gene's information is converted into the structures and functions of a cell. In other words, it is a process by which gene's information is converted into the proteins by cells. So a gene gets to express itself. In combination with classification methods, this technology can be useful in supporting clinical management decisions for patients. Experimental conditions may include types of cancers, diseased organisms, or normal tissues. To have an idea of necessity to analyze thousands of genes within a single experiment, it is enough to consider following statement: "Multicellular organisms are created from a complex organization of cooperating cells. In humans, for instance, there are 10 to the power of 14 cells comprising 200 kinds of various tissues! "

Observing any living thing we find the cells as the main building blocks of its internal structure. In other words, cells make up all living systems. Cells demonstrate an important dynamic property. Namely, they are able to adjust their behavior with respect to environmental stimulus. This is achieved by constant sampling (listening) of the multitude of molecules in their environments. Many of these molecules can actually be thought of as signal carriers that convey information to the cell. There is a large array of such molecules but they are all in form of proteins.

Therefore, overall functionality of the cell and entire organism is determined by the “instruction” encoded in the form of proteins. But how are these proteins synthesized? This is the question answered by central dogma of molecular biology which states that process of protein biosynthesis is actually a flow of genetic information from nucleus DNA, to RNA and than to protein. The RNA encodes a sequence of amino acids which define a particular protein. The sequence of amino acids of a protein dictates its 3D structure that it adopts spontaneously. With a few notable exceptions, all biological cells conform to this rule. This practically means our functionality is regulated by information stored in the molecule of DNA. DNA is a very long molecule consisting of 4 main building blocks (nucleotides), respectively: adenine, guanine, cytosine and thymine. DNA is found in shape of double helix. However it is a very long molecule and as such is broken into segment called genes for easier analysis. Genes are entities that parents pass to offspring during reproduction. They encode information essential for the construction and regulation of proteins and other molecules that determine the growth and functioning of the organism. Genetics (from the Greek *γεννώ*= give birth) is the science of genes, heredity, and the variation of organisms. The word genetics was first applied to describe the study of inheritance and the science of variation by English scientist William Bateson in a letter to Adam Sedgewick, dated April 18, 1905.

Very related to genetics is another branch of science called Bioinformatics or computational biology which uses techniques from applied mathematics, informatics, statistics, and computer science to solve biological problems. Major research efforts in the field include prediction of gene expression, sequence alignment, protein structure prediction, etc.

11.2 DNA MICRO-CHIPS

"If you want to learn what words mean in a foreign language you look at how they are used. It's the same for genes. Microarrays as a way of seeing how genes express themselves will be the most widely used application of arrays."

The concept emerged 10 years ago. A research paper by Fodor opened the way for the entire microarray industry. DNA microarrays are microscopic groups of thousands of DNA molecules of known sequences attached to a solid surface such as a nylon membrane or a simple glass microscope slide. Each array consists of an orderly organization of samples that provides a medium for matching known and unknown samples based on base-pairing rules and automating the process of identifying the unknowns. Microarrays come in several varieties, each of which has specific advantages for research and screening. Depending on the size of each DNA spot on the array, DNA arrays can be categorized as microarrays when the diameter of DNA spot is less than 250 microns, and macroarrays when the diameter is bigger than 300 microns.

Brown's team at Stanford, in collaboration with Mark Schena and Ron Davis, working as consultants to Affymetrix, developed the basic technology for what scientists now regard as the traditional type of microarray. It uses lengths of complementary DNA (or cDNA) produced from cellular

messenger RNA using the reverse transcriptase polymerase chain reaction (RT-PCR). Stretches of cDNA about 500 to 5,000 bases long are immobilized onto a substrate and exposed to a set of targets either separately or in a mixture.

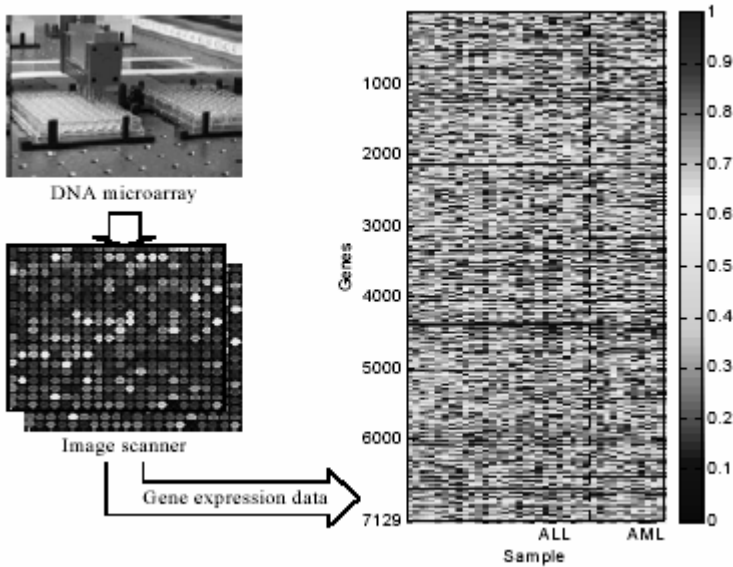


Figure 1 General process of acquiring the gene expression data from DNA microarray

DNA microarrays are composed of thousands of individual DNA sequences printed in a high density array on a glass microscope slide using a robotic arrayer as shown in Fig. 1. The relative abundance of these spotted DNA sequences in two DNA

or RNA samples may be assessed by monitoring the differential hybridization of the two samples to the sequences on the array. For mRNA samples, the two samples are reverse-transcribed into cDNA, labeled using different fluorescent dyes mixed (red-fluorescent dye Cy5 and green-fluorescent dye Cy3). After the hybridization of these samples with the arrayed DNA probes, the slides are imaged using scanner that makes fluorescence measurements for each dye. The log ratio between the two intensities of each dye is used as the gene expression data (Lashkari et al. 1997, Derisi et al. 1997, Eisen et al. 1998).

$$gene_expression = \log_2 \frac{Int(Cy5)}{Int(Cy3)} \quad (Eq. 1)$$

Where $Int(Cy5)$ and $Int(Cy3)$ are the intensities of red and green colors.

11.3 SUPERVISED VERSUS UNSUPERVISED LEARNING

It is essential to efficiently analyze DNA microarray data because the amount of DNA microarray data is usually very large. Many machine learning and data mining methods have been applied to solve them.

11.3.1 Classification (Supervised versus Unsupervised)

There are two main divisions of classification:

11.3.1.1 Supervised classification or discrimination

Unsupervised classification simply referred to as classification or clustering. In supervised classification we have a set of data samples with associated labels, class types. An example of application is radar target recognition of objects. Related to our course of study we use classification to select genes related to cancer, also to select cancer genes from normal tissues.

In unsupervised classification the data are not labeled and we seek to find the gropes in the data and the features that distinguish one object from another. In our example, we use clustering to read the results from the microarray. We want to se which are genes that are dominant and we don't care if they are from the cancer tissue or normal tissue. We perform an unbiased observation.

11.3.1.2 Unsupervised classification or Clustering

Clustering methods are used in for data exploration and to provide prototypes for use in supervised classifiers. Available methods operate both on dissimilarity matrices and measurements on individuals, each imposing its own structure on data.

Cluster analysis is the groping of individuals in population in order to discover structure in data. In some sense we would like the individuals within the group to be close or similar to one another, but dissimilar from individuals in other groups.

Clustering is fundamentally a collection of methods of methods of data exploration.

We can roughly distinguish among following clustering methods:

1. Hierarchical methods
2. Quick partitions
3. Mixture models
4. Sum-of-squares
5. Cluster validity

11.3.1.2.1 Hierarchical Clustering Methods

Hierarchical Clustering Methods are the most commonly used methods of summarizing data structure. A hierarchical tree is a nested set of partitions represented by a tree diagram or dendogram.

- Single link method
- Complete link method
- Sum-Of Squares method (Ward)
- General agglomerative algorithm (Centroid distance, Median distance, group average link)

11.3.2 Statistical Learning and Analysis

Each DNA hybridization array experiment generates thousands of data points, and each study (containing many experiments) can result in millions of data points (Sherlock, 2000). Therefore, the interpretation and verification of array databases present a major challenge. While no universal algorithm exists for array data management, many experiments undergo a process of normalization, unsupervised analysis, and supervised analysis (Young, 2000).

Unsupervised analysis is commonly used for exploratory tasks, such as an unbiased discovery of gene expression patterns. Data grouped in an unsupervised analytic strategy are termed "clusters" (Tamayo et al., 1999). Several mathematical models exist for the clustering. Cluster algorithms group similar profiles based on a distance metric, usually by the statistical correlation coefficient or Euclidean distance (Freeman et al., 2000).

11.4 DNA-CHIP DATA PROCESSING

A common procedure of analysis in microarray technology is to conduct several experiments across the same genes, measuring gene expression during each trial (e.g. different patients, time points, etc.). The end result is often expression arrays of high dimensionality. For example, if you have 10 trials measured across 10,000 genes, you have a 10 by 10,000 matrix (10,000 genes in 10 dimensions). In order to detect a pattern in the data, researchers traditionally use methods that reduce the dimensionality to just two dimensions (along an x and y axis).

Many methods exist to accomplish this task, but I will focus on just one: cluster analysis.

11.4.1 Loading Data

The first step is to import data. Currently, our program only reads tab delimited text files in a particular format, described below. Such tab-delimited text files can be created and exported in any standard spreadsheet program, such as Microsoft Excel. (Create the excel file and save by .txt extension)

By convention, input tables rows represent genes and columns represent samples or observations (e.g. a single microarray hybridization). For a simple time-course, a minimal input file would look like this:

YORF	0 minutes	30 minutes
YAL001C	1	1.3
YAL002W	0.9	0.8
YAL003W	0.8	2.1
YAL005C	1.1	1.3
YAL010C	1.2	1

Figure 2 Time-course microarray data

This is to satisfy multiple experiment data sets, however single ones are also supported and in that case we would have only two columns. Each row has an identifier (in green) that always goes in the first column. It actually represents the name of the gene of interest. Each column (sample) has a label (in blue) that is always in the first row; here the labels describe the time at which a sample was taken. It can be any other trial of interest such as tumor cells of different patients etc.

The first column of the first row contains a special field (in red) that tells the program what kinds of objects are in each row. In this case, YORF stands for yeast open reading frame. This field can be any alpha-numeric value.

This format of input file was first proposed by M. Eisen at Stanford University and today represents a widely accepted standard.

To test the output of my program I was using the test data set provided by the same author and available from <http://rana.lbl.gov/downloads/data/demo.txt>

1.4.2 Filtering data

Once data set is uploaded, we need to reduce the size of data set by excluding unwanted variables such as genes that do not show a significant difference in expression values between the control and experimental groups. This is aimed to improve computation speed since we are dealing with very large data sets.

Example of filtering methods recommended and proven to be useful in recent papers are:

- Removing all genes that have missing values in percentage greater than $(100 - X)$ of the columns.
- Removing all genes that have standard deviations of observed values less than X . Removing all genes that all genes that do not have at least X observations with absolute values greater than .
- Removing all genes whose maximum minus minimum values are less than X .

These are fairly self-explanatory and it is trivial thing to code these filters in Matlab.

11.4.3 Adjusting data

Adjusting data actually perform a number of operations that alter the underlying data in the imported table. These operations that have been proposed by number of authors include:

- Log Transform Data (replace all data values x by $\log_2(x)$).
The results of DNA microarray experiments are fluorescent ratios. Ratio measurements are most naturally processed in log space.

Illustrative Example given by M. Eisen: “Consider an experiment where you are looking at gene expression over time, and the results are relative expression levels compared to time 0. Assume at time point 1, a gene is unchanged, at time point 2 it is up 2-fold and at time point three is down 2-fold relative to time 0. The raw ratio values are 1.0, 2.0 and 0.5. In most applications, you want to think of 2-fold up and 2-fold down as being the same magnitude of change, but in an opposite direction. In raw ratio space, however, the difference between time point 1 and 2 is +1.0, while between time point 1 and 3 is -0.5. Thus mathematical operations that use the difference between values would think that the 2-foldup change was twice as significant as the 2-fold down change. Usually, you do not want this. In log space (we use log base 2 for simplicity) the data points become 0,1.0,-1.0. With these values, 2-fold up and 2-fold down are symmetric about 0.”

- Normalize Genes and/or Arrays: Multiply all values in each row and/or column of data a scale factor S so that the sum of the squares of the values is in each row and/or column is 1.0 (a separate S is computed for each row/column).

- Mean Center Genes and/or Arrays: Subtract the row-wise or column-wise mean from the values in each row and/or column of data, so that mean value of each row and/or column is 0.

Explanation by M. Eisen (Stanford University): “Consider a now common experimental design where you are looking at a large number of tumor samples all compared to a common reference sample made from a collection of cell-lines. For each gene, you have a series of ratio values that are relative to the expression level of that gene in the reference sample. Since the reference sample really has nothing to do with your experiment, you want your analysis to be independent of the amount of a gene present in the reference sample. This is achieved by adjusting the values of each gene to reflect their variation from some property of the series of observed values such as the mean or median. This is what mean and/or median centering of genes does. “

However, these operations are not associative, so the order in which these operations are applied is very important to preserve the meaning of the data.

- The order of operations is
- Log transform all values.
- Mean center rows.
- Normalize rows.
- Mean center columns.
- Normalize columns.

Again, Matlab is used as a sufficient tool to perform all these operation.

11.4.3 Distance/Similarity Measures (Creating Similarity Matrix)

The first choice that must be made is how similarity (or dissimilarity or distance) between gene expression data is to be defined. There are many ways to compute how similar two series of numbers are.

11.4.4 Pearson Correlation

Usually conceived of as applicable to situations where X and Y are interval or ratio scales (quantitative variables). In micro array's case we have log ratios of colors and therefore the most commonly used similarity metrics are based on Pearson correlation.

The Pearson correlation coefficient between any two series of numbers $x = \{x_1, x_2, \dots, x_n\}$ and $y = \{y_1, y_2, \dots, y_n\}$ is defined as:

$$r = \frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{\sigma_x} \right) \left(\frac{y_i - \bar{y}}{\sigma_y} \right), \quad (\text{Eq. 2})$$

where \bar{x} is mean (average) value, calculated as $(x_1 + x_2 + x_3 + \dots + x_n)/n$ and σ is standard deviation

$$\sigma_x = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i)^2};$$
$$\sigma_y = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i)^2}.$$

(Eq. 3)

There are many ways of conceptualizing the correlation coefficient. Using a scatterplot of the values of x against y (pairing x1 with y1, x2 with y2, etc. The simplest way to think about the correlation coefficient is to plot x and y as curves, with r telling how similar the shapes of the two curves are. The Pearson correlation coefficient is always between -1 and 1, with 1 meaning that the two series are identical, 0 meaning they are completely uncorrelated, and -1 meaning they are perfect opposites. The correlation coefficient is invariant under linear transformation of the data. That is, if you multiply all the values in y by 2, or add 7 to all the values in y, the correlation between x and y will be unchanged. Thus, two curves that have identical shape, but different magnitude, will still have a correlation of 1.

We have used Pearson Coefficient in my program to obtain similarity matrix which is displayed in grid box. Here is the source code:

```

function res = pearson(matrix)
D = size(matrix)
D
for i = 1:D(1)
    for j = (i+1):D(1)
        x = matrix(i, :);
        xx = matrix(j, :);

        A = 0;

        for k=1:size(D(2))
            A = A + ((x(k)-mean(x))/std(x)) * (xx(k)-mean(xx))/std(xx);
        end

        res(i,j) = (1/D(2)) * A;
        res(j,i) = (1/D(2)) * A;
    end
end
end

```

Figure 3 Pearson coefficient pseudo-code

A newly added distance function is the Euclidean distance, which is defined as:

$$d(x, y) = \sum_{i=1}^n (x_i - y_i)^2 .$$

The Euclidean distance takes the difference between two gene expression levels directly. It should therefore only be used for expression data that are suitably normalized, for example by converting the measured gene expression levels to log-ratios. Unlike the correlation-based distance measures, the Euclidean distance takes the magnitude of changes in the gene expression levels into account. It therefore preserves more information about the gene expression levels than the other distance measures mentioned above. An example of the Euclidean distance applied to k-means clustering can be found in De Hoon, Imoto, and Miyano (2002).

11.5 CLUSTERING

Since we already have similarity matrix created by function `person()`, logical next step is clustering of these data based on their similarity.

Clustering can be considered the most important unsupervised learning problem; so, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data. A loose definition of clustering could be “the process of organizing objects into groups whose members are similar in some way”. A cluster is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters.

There are several clustering algorithms generally proposed for use in DNA micro array experiments. Such as: Hierarchical Clustering, SOM (self organizing maps), k-means clustering, support vector machine Hierarchical clustering methods organize genes in a tree structure, based on their similarity. Four variants of hierarchical clustering are studied in this project: Single link method, Complete link method, Sum-Of Squares method (Ward) and average link method.

11.5.1 Single Linkage Clustering

In Single Linkage Clustering the distance between two items x and y is the minimum of all pairwise distances between items contained in x and y . In single linkage clustering no further distances need to be calculated once the distance matrix is known.

11.5.2 Complete Linkage Clustering

In Complete Linkage Clustering the distance between two items x and y is the maximum of all pairwise distances between items contained in x and y . As in single linkage clustering, no other distances need to be calculated once the distance matrix is known.

11.5.3 Average Linkage Clustering

In average linkage clustering, the distance between two items x and y is the mean of all pairwise distances between items contained in x and y .

11.5.4 Ward (Sum-of-Squares) Linkage Method

The Sum-of-Squares method is appropriate for the clustering of points in Euclidian space. The aim is to minimize total within group sum of squares.

11.5.5 Graphical Observation of Results (Dendrogram)

For graphical observation we use a dendrogram—a binary tree in which subtrees are each a cluster and the leaves are individual genes. The distance from the root to a subtree indicates the similarity of subtrees—highly similar nodes or subtrees have joining points farther from the root.

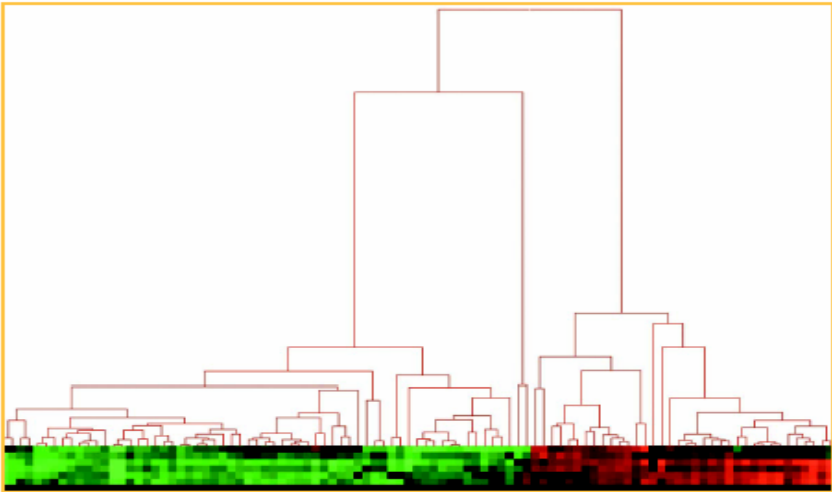


Figure 3 An example of dendrogram

11.6 DISCUSSION

DNA array technology and bioinformatics are rapidly evolving and becoming better able to address important biologic questions. Microarray-based genomic surveys and other high-throughput approaches play a part in the process. As a result, we need to

develop our ability to "see" the information in the massive tables of quantitative measurements that these approaches produce.

Our approach to this problem can be generalized as follows. First, we use a common-sense approach to organize the data, next we filter these data, optionally we apply various transformations on the data, then we use correlation techniques to create similarity matrix, then apply clustering algorithm. And all of this just to be able to display data graphically in a form of dendrogram. This is important because human brains are not well adapted to assimilating quantitative data by reading digits; we represent the quantitative values in a form of hierarchical tree by using a naturalistic color scale rather than numbers. This alternative encoding preserves all the quantitative information, but transmits it to our brains by way of a much higher-bandwidth channel than the "number-reading" channel.

11.7 CONCLUSION

A natural way of viewing complex data sets is first to scan and survey the large-scale features and then to focus in on the interesting details. What we have found to be the most valuable feature of the approach described here is that it allows this natural and intuitive process to be applied to genomic data sets. The approach is a general one, with no inherent specificity to the particular method used to acquire data or even to gene-expression data. It is therefore likely that very similar approaches may be applied to many other kinds of very large data sets. In each case, it may be necessary to find alternative algorithms and computation methods to bring out inherent structures in the data, and, equally important, to find dense naturalistic visual representations that convey the quantitative information effectively. We recognize that the particular clustering algorithm we may use is not the only, or even the best, method available.

REFERENCES

- Alevizos I., Mahdevappa M., Ohyama H., Zhang X., Kohno Y., Colevas D, *et al.* (2001). Head and neck/oral cancer gene expression profiling assisted by laser capture microdissection and GeneChip analysis. *Oncogene*20:6196–6204.
- Aschary M.P., Jaggernauth W., Gross E., Alfieri A., Klinger H.P., Vikram B. (2000). Cell lines from the same cervical carcinoma but with different radiosensitivities exhibit different cDNA microarray patterns of gene expression. *Cytogenet Cell Genet*91:39–43.
- Brown M.P., Grundy W.N., Lin D., Cristianini N., Sugnet C.W., Furey T.S., *et al.* (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc Natl Acad Sci USA*97:262–267.
- Butte A.J., Tamayo P., Slonim D., Golub T.R., Kohane I.S. (2000). Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proc Natl Acad Sci USA*97:12182–12186.
- Calaluce R., Kunkel M.W., Watts G.S., Schmelz M., Hao J., Barrera J., *et al.* (2001). Lamin-5-mediated gene expression in human prostate carcinoma cells. *Mol Carcinog*30:119–129.
- Califano J., van der Riet P., Westra W., Nawroz H., Clayman G., Piantadosi S., *et al.* (1996). Genetic progression model for head and neck cancer: implications for field cancerization. *Cancer Res*56:2488–2492.

- Califano A., Stolovitzky G, Tu Y (2000). Analysis of gene expression microarrays for phenotype classification. *Proc Int Conf Intell Syst Mol Biol*8:75–85.
- Chang D.D., Park N.H., Denny C.T., Nelson S.F., Pe M. (1998). Characterization of transformation related genes in oral cancer cells. *Oncogene*16:1921–1930.
- Clark E.A., Golub T.R., Lander E.S., Hynes R.O. (2000). Genomic analysis of metastasis reveals an essential role for RhoC. *Nature*406:532–535.
- Coller H.A., Grandori C., Tamayo P., Colbert T., Lander E.S., Eisenman R.N., *et al.* (2000). Expression analysis with oligonucleotide microarrays reveals that Myc regulates genes involved in growth, cell cycle, signaling, and adhesion. *Proc Natl Acad Sci USA*97:3260–3265.
- DeRisi J., Penland L., Brown P.O., Bittner M.L., Meltzer P.S., Ray M., *et al.* (1996). Use of a cDNA microarray to analyze gene expression patterns in human cancer. *Nat Genet*14:457–460.
- Duan Z., Feller A.J., Penson R.T., Chabner B.A., Seiden M.V. (1999). Discovery of differentially expressed genes associated with paclitaxel resistance using cDNA array technology: analysis of interleukin (IL) 6, IL-8, and monocyte chemotactic protein 1 in the paclitaxel-resistant phenotype. *Clin Cancer Res*5:3445–3453.
- Edman C.F., Raymond D.E., Wu D.J., Tu E., Sosnowski R.G., Butler W.F., *et al.* (1997). Electric field directed nucleic acid hybridization on microchips. *Nucleic Acids Res*25:4907–4914.
- Emmert-Buck M.R., Strausberg R.L., Krizman D.B., Bonaldo M.F., Bonner R.F., Bostwick D.G., *et al.* (2000). Molecular

Ito W., Siti Zaiton M. H. and Sri S. (2007.) Iterative local Gaussian clustering for expressed genes identification linked to malignancy of human colorectal carcinoma. *Bioinformation*, Vol. 2, No. 5.

INDEX

- accuracy, 1, 9, 15, 20, 27, 40, 42, 45, 46, 51, 53, 56, 65, 70, 76, 89, 92, 105, 107, 108, 112, 118, 121, 123, 125, 129, 132, 134, 147, 158
- test, 9, 27, 42, 45, 46, 53, 56, 70, 76, 92, 112, 123, 125, 172
- algorithm, 4, 9, 15, 22, 27, 36, 40, 42, 53, 56, 65, 76, 89, 105, 121, 156, 158, 172
- gain ratio, 92, 108, 134
- GASVM-II, 45, 53, 56, 76, 92, 112, 125
- genetic, 40, 53, 56, 65, 89, 105, 121, 132, 170
- information gain, 92, 108
- MOGASVM, 40, 42, 45, 46, 76, 89, 92, 108, 112, 118, 121, 123, 125
- particle swarm optimisation, 92, 118
- approach, 1, 9, 15, 20, 27, 36, 40, 42, 45, 46, 51, 53, 56, 65, 67, 68, 76, 89, 92, 105, 107, 121, 123, 129, 132, 134, 147, 150, 158, 172
- chromosome, 42, 53, 56, 67, 70, 76, 89, 92, 107, 112, 121
- classification, 4, 9, 15, 22, 27, 40, 42, 45, 46, 51, 56, 65, 76, 89, 92, 105, 108, 112, 118, 121, 125, 129, 132, 134, 147, 170, 172, 190
- cancer, 40, 45, 46, 51, 53, 56, 65, 76, 89, 92, 105, 108, 112, 118, 121, 125, 129, 132, 134, 172, 190
- classifier, 9, 27, 40, 42, 51, 53, 56, 65, 70, 76, 89, 92, 105, 108, 112, 121, 123, 132
- COLON**, 132
- computational, 1, 9, 20, 27, 76, 147, 170
- cross-validation, 9, 27, 42, 53, 76, 92, 108, 112, 125, 134
- LOOCV, 42, 45, 46, 53, 56, 70, 76, 92, 108, 112, 123, 125, 134
- data, 1, 4, 9, 15, 20, 22, 27, 40, 42, 45, 46, 51, 53, 56, 65, 67, 68, 70, 76, 89, 92, 105, 107, 108, 112, 118, 121, 123, 125, 129, 132, 134, 150, 172, 190
- dimensionality, 51, 53, 65, 67, 70, 76, 89, 92, 105, 107, 108, 112, 118, 121, 123, 125, 129, 147, 172
- high-dimensional, 89

- noisy, 51, 53, 56, 65, 76, 89, 92, 105, 108, 112, 118, 121, 132
- biomarker, 150, 163
 - samples, 40, 42, 45, 51, 53, 56, 65, 76, 89, 92, 105, 112, 121, 125, 132, 134, 172
 - training, 1, 4, 9, 15, 20, 22, 27, 36, 42, 45, 53, 56, 76, 92, 112, 125, 134
- design, 147, 172
 - drug, 76, 112, 125, 147
- diagnosis, 51, 56, 65, 76, 112, 121, 125

- entropy, 147, 150, 158
- filter
 - approach, 40, 51, 65, 76, 89, 92, 105, 107, 108, 112, 118, 121, 172
- fitness, 42, 53, 70, 76, 123, 132, 134
- functions, 1, 15, 20, 36, 42, 147, 156, 170

- gene, 1, 15, 20, 27, 36, 40, 42, 45, 46, 51, 53, 56, 65, 67, 68, 70, 76, 89, 92, 105, 107, 108, 112, 118, 121, 123, 125, 129, 132, 134, 170, 172, 190
 - expression, 1, 20, 40, 42, 45, 46, 50, 51, 53, 56, 65, 67, 68, 76, 89, 92, 105, 121, 132, 170, 172, 190
 - informative, 9, 27, 51, 53, 56, 65, 76, 89, 92, 105, 107, 108, 112, 118, 121, 125, 129, 132, 134, 158
- genomes, 1, 15, 20, 36

- hydrophobicity, 1, 4, 9, 15, 20, 22, 27
- interactions, 1, 3, 4, 9, 15, 20, 22, 27, 36, 56, 147
- kernel, 4, 9, 22, 27

- leukemia, 45, 76, 92
 - acute lymphoblastic, 45, 56, 92, 112
 - acute myeloid, 45, 53, 56, 92, 112

- machine learning, 1, 4, 20, 22, 172
- method
 - clustering, 172
 - experimental, 1, 9, 20, 27, 45, 53, 56, 65, 70, 76, 89, 92, 105, 118, 125, 129, 134, 172
 - hybrid, 1, 20, 40, 42, 51, 53, 56, 65, 67, 68, 76, 89, 92, 105, 107, 108, 112, 121, 123, 129, 132, 134, 147
 - unsupervised, 172
- methods, 1, 20, 40, 51, 53, 56, 65, 67, 68, 70, 76, 89, 92, 105, 107, 108, 112, 118, 121, 123, 125, 129, 132, 134, 147, 150, 158, 170, 172
- microarray, 40, 51, 53, 56, 65, 76, 89, 92, 105, 107, 112, 118, 121, 123, 125, 129, 132, 134, 172, 190
- molecular, 1, 20, 40, 51, 56, 65, 121, 150, 170
- multi-objective, 40, 42, 46, 56, 67, 76, 89, 107, 108, 121

- optimize, 9, 27
- organism, 170

- parameter, 4, 22, 42, 67, 76, 89, 92, 107, 112, 147, 158
- Pearson Coefficient, 172
- phenotypes, 40, 51
- population, 53, 76, 92, 112, 134, 172
- prognostic, 132
- protein, 1, 3, 4, 9, 15, 20, 22, 27, 36, 40, 56, 76, 112, 147, 150, 156, 158, 170, 190
 - amino acids, 4, 22, 147, 150, 170
- ROC, 1, 9, 20, 27
- selection, 40, 42, 45, 46, 51, 53, 56, 65, 67, 68, 70, 76, 89, 92, 105, 107, 108, 112, 118, 121, 123, 129, 132, 134
- sensitivity, 9, 27
- sequence, 1, 3, 4, 9, 20, 22, 27, 112, 147, 150, 156, 158, 170
- single-objective, 42, 45, 46, 56, 76, 92, 125
- specificity, 9, 27, 172
- structure, 1, 4, 9, 15, 20, 22, 27, 36, 147, 150, 156, 158, 170, 172
 - 3D, 147, 150, 156, 158, 170
- subset, 40, 42, 45, 46, 51, 53, 56, 65, 67, 70, 76, 89, 92, 105, 107, 108, 112, 118, 121, 123, 125, 129, 132, 134
- supervised, 4, 15, 22, 36, 172
- support vector machine, 4, 22, 40, 65, 89, 105, 121, 134, 172
- yeast, 1, 4, 9, 20, 22, 27, 172