Integration and Reduction of Microarray Gene Expressions Using an Information Theory Approach

Shamsaee, Reza; Fathy, Mahmood
Faculty of Computer Engineering, Iran University of Science and Technology (IUST), Tehran, I.R. IRAN

Masoudi-Nejad; Ali*+
Laboratory of Systems Biology and Bioinformatics (LBB), Institute of Biochemistry and Biophysics (IBB), University of Tehran, Tehran, I.R. IRAN

ABSTRACT: *The DNA microarray is an important technique that allows researchers to analyze many gene expression data in parallel. Although the data can be more significant if they come out of separate experiments, one of the most challenging phases in the microarray context is the integration of separate expression level datasets that have gathered through different techniques. In this paper, we present a general novel method for the integration of any collected data whose distributions have been linearly transformed. The new method is based on the information theory concepts. More than that, this article presents a new approach for checking of the linearity between two distributions as a validation technique. The validation technique assists in taking the feature reduction process in effect prior to the integration phase. The time complexity of the proposed algorithm is low and the new presented methods show good functionality. The experimental results are presented at the end of the paper.*

KEY WORDS: *Microarray, Microarray integration, Information theory, Feature reduction, Classification.*

INTRODUCTION

Expression levels of thousands of genes can be examined in parallel by using the DNA microarray technology. This powerful technology was first introduced in 1999 by *Patrick Brown & Vishwanath Lyer*. This technique allows scientists to perform many hybridization processes simultaneously in a single chip [1]. It has been prooved that the collected data can help scientists in discriminating tumors and normal-cells [2]. Nowadays, it has commonly used to finding genomic deviations in many diseases such as different cancers, hepatitis, and etc [3,4]. Furthermore, it has a great potential for gene-therapy in a near future.

* *To whom correspondence should be addressed.*
+ *E-mail: amasoudin@ibb.ut.ac.ir*
1021-9986/10/4/19 11/$/3.10

But as a drawback, microarray experiments are expensive, and performed over few numbers of individual samples by different labs. Consequently, differences in results are something in common between these separate datasets [5].

In contrast to few samples, microarray experiments create a lot of data around the gene expression. Each sample contains a huge number of genes whose expression levels are described by a microarray experiment in parallel. Each expression level of a gene can be seen as a feature, while a sample is imagined as a feature vector in the sense of classification task. We use these terminologies interchangeably throughout this context.

Obviously, many genes are noise ones in a specific classification task, due to this fact that not all of them take a part or equal part in the classification process of a particular genomic deviation. Omitting a noise gene, reduces the dimensions of existing genes. Some well-known techniques which are used in the feature reduction of microarray's data are as follow: Information gain [6], Neural network [7], SVM [8], and etc.

Unfortunately, these techniques are not capable of recognizing and compensating any differences in the distribution of a feature. In other word, if some feature vectors are transformed, some techniques can only investigate them as the irregular cases and put them aside. These techniques can't inversely transform these feature vectors to make them take a part in the whole process.

Integration of microarray's datasets tries to create a larger and consistent dataset out of some datasets which are gathered via dissimilar techniques by some different labs. Although, the samples in the labs are not alike, the distribution of gene expressions is expected to be similar for a specific classification task. But, as the labs use some different techniques and setups, dissimilarity can be seen in the distribution of the gene expressions inside their datasets. So, the integration of gene expression data in different microarray's datasets is not something trivial.

This paper outlines a general method for validating the features and integrating the samples. Invalid features whose distribution is disturbed by a non-linear transformation will be omitted. So, the feature vector will be robustly reduced and ready for the integration. These validations will be used not only as a robust criterion for omitting the noise genes, but also make the integration process easier. The Integration method will be applied at the next step on the genes remained whose distributions are linearly transformed and can be seen as an inverse transform. These three methods for validating, omitting, and integrating the features are well discussed in this paper by maneuvering information theory concepts and its mathematics.

The organization of this article is as follow: in the two following section we will review the related work. In the third one and in its subsections, we will present our newly proposed method and its background mathematics. The fourth section will illustrate the algorithm of our novel integration method. The experimental results and conclusion will be presented at the end of this paper.

SOME RELATED WORKS ON MICROARRAY DATA INTEGRATION

It is possible to do the integration via normalizing the data from each individual research to obtain a common ground. The normalization is performed on the most prominent set of genes. While z-score is commonly used as normalization approach by the papers, differences in the proposed methods are usually in the gene selection part.

As examples of this approach, *Lai et al.* [9] use a special statistical test of differential expression for each gene to obtain a list of test scores. Whereas, *Jiang et al.* [10] are looking for marker genes after a gene shaving method, based on random forests. *Yoon et al.* [11] found a subset of genes that has showed high expression values on a specific class and low expression values on the other by the means of informative gain called 'informative genes'.

Another method for the integration is modeling. In this approach we are trying to formalize the reality inside a model. Any subsequent action for the integration takes place based on the models. The correlation signature by *Kang et al.* [12] is one of them. In this method first a gene signature vector is created for each dataset showing the patterns of corresponding landmark genes on that dataset. By organizing the patterns in a cubic shape, one can integrate the landmark genes of datasets. Another modeling approach is Meta-analyze [2]. It is done by obtaining an effective size by examining some models. These methods are based on the probability model building. It seems that both methods suffer from high complexity and time cost.

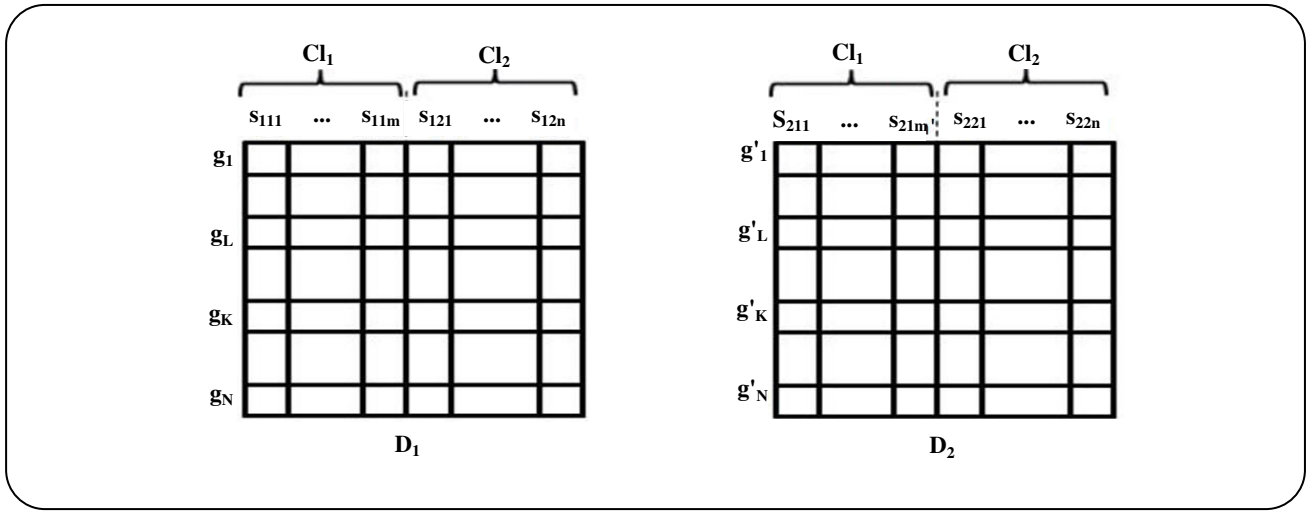

Fig 1: D1, D2 are two different gene expression datasets that have gathered for a specific field of study but by two different settings or techniques. Each column s_{xyz} represents a different sample while subscripts x, y, and z show the dataset index, class index, and number of samples correspondingly. Rows are genes.

In addition to the above researches, there are lots of papers about the integration of separate microarray studies such as [13-18], these researches more or less, can be categorized as normalization or model building methods.

PROPOSED METHODS

The current paper presents an approach based on the information theory model building. To describe the method, some terminologies are needed to be explained first. Suppose that two separate experiments were done in a same discipline like prostate cancer. But they were performed in two different settings or even with different techniques, and each of which led to separate gene expression datasets D_1, and D_2. They can be visualized as Fig. 1.

Each row represents a gene and each column shows a sample. There are two classes in each dataset: Cl_1=Normal, Cl_2=Tumor. The m value is the number of samples in the class Cl_1, and n is the number of samples in the class Cl_2 from D_1, while m' and n' are the numbers of samples in the related classes from D_2, respectively. The total number of samples in D_1 is M = m + n, and M' = m' + n' is the total number of samples in D_2. N is the number of total genes per sample in the datasets D_1 and D_2. The rows g_l and g_k show the gene expression values in D_1 while g'_l and g'_k show expression values of the same genes in D_2.

As, there isn't any presumptions about the equality of noise patterns and settings between two experiments, the equality between probability distribution of any desired gene g_l from the first experiment and its related gene g'_l in the second dataset can't be held even in one class. Particularly, in the case of different standards such as oligo chips for one gene expression dataset and cDNA chip for the other, heterogeneous datasets will be achieved which makes the direct comparison impossible. In this case normalization techniques are very likely to lead to misclassifications.

From here on, g_l, g_k, g'_l, and g'_k will be assumed as the random variables and we will denote them by X, Y, X' and Y' respectively. And these random variables can get different values from their distributions. The D_1 will be supposed as the base dataset, and remains intact.

Background theory

In this subsection we try to prove the following theorem using appendix A.

Theorem1: A similar linear transformation over any two probability distribution of the random variables X and Y (i.e. X' = S.X + c and Y' = s.Y + c), leaves the mutual information intact.

Proof: From information theory concepts [19] mutual information can be defined as:

$$I(X';Y') = H(X') - H(X' \mid Y') \quad (1)$$

Using above equation and lemma1 and lemma2 from appendix A, we can rewrite it as follow:

$$I(X';Y') = H(X) + \log(s) - \left[H(X \mid Y) - \log(s)\right] = \quad (2)$$
$$H(X) - H(X \mid Y) = I(X;Y)$$

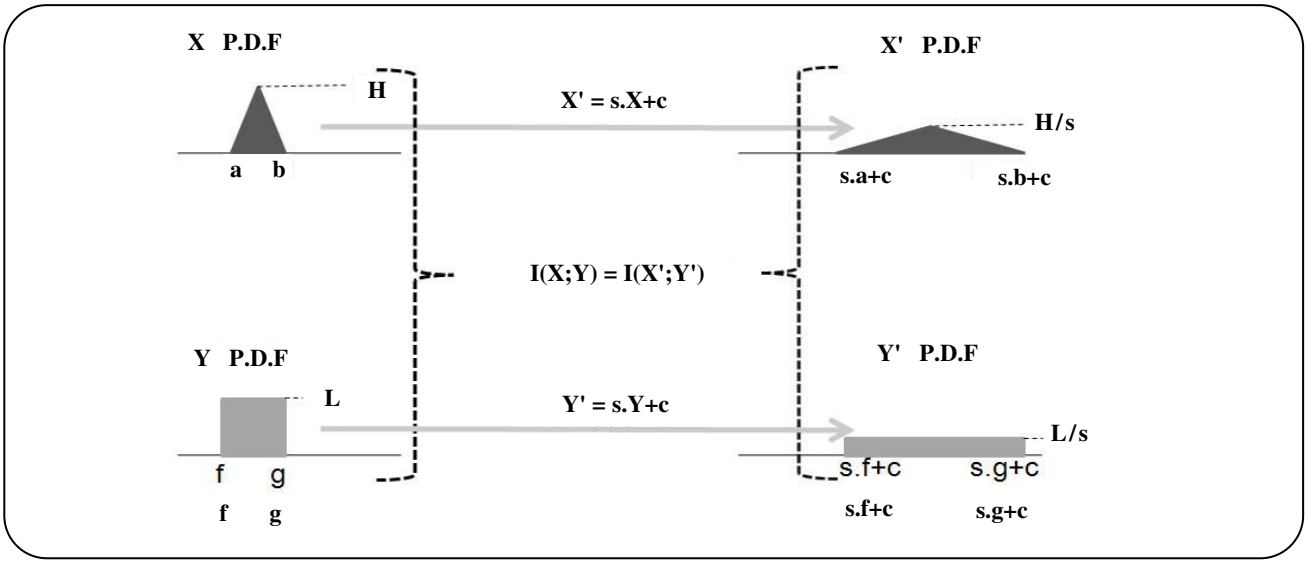

Fig. 2: The Probability Distribution Functions (PDF) of both X and Y are transformed to X', Y' by a desired T. T is described by two quantities s (scaling factor) and c (translation factor) as T(x)=s.x +c. I(X;Y)and I(X';Y') are mutual information between X, Y, and X', Y', respectively.

Due to the Eq. (2), it is proved that the mutual information between any two random variables remains intact while a similar linear transformation occurs on their distributions.

Proposed validity testing method

As it was said earlier, different labs use different techniques and setups. So, dissimilarity can be seen in the distribution of the gene expressions inside their datasets. Suppose that the distribution of X' is different from X, based on an occurred transformation T; and T is a direct outcome of the different techniques and settings between D_1 and D_2. In other word, as a result of the desired transformation T, the same gene has got different distributions in its expression level between D_1 and D_2. Fig. 2 shows the idea schematically.

Theorem1 proves that mutual information between any two desired genes' distributions remain intact through different acquisition techniques while any two given genes' expression distributions have been linearly transformed in a similar manner. In other word, if T is a linear transformation and if it takes effect on the distributions of X, Y to create distributions of X', Y', the mutual information between X, Y and X', Y' will be remained intact (Fig. 2). More than that, we can generalize Eq. 2 to any number of linear transformations. Unfortunately, if mutual information is intact, T can be linear or non-linear. But according to mathematic formula, a non-linear transformation deforms shapes. So, we are able to implicate in this way: if the mutual information is intact and the shape of the distribution is not deformed, T will be linear. But, as the computational load of shape checking is not something trivial, and both of X and X' are expressions for the same gene in a specific domain of study e.g. prostate cancer, it is supposed that T will be linear if the mutual information is intact.

Now, we are able to use Eq. (2) as a tool for identifying the transformation types between the genes of datasets. Finding any pairs of genes that have linearly transformed helps us to separate genes into two categories: linearly transformed, and non-linearly transformed. It could be done by comparing I(X, Y) any two desired genes from D_1, and I(X', Y') for their corresponding transformed genes in D_2. Our validating test process is based on finding of those genes which are linearly transformed. If there are two desired genes X, Y within D_1 and their transformed genes X', Y' within D_2, it could be said that they are linearly transformed as the followings:

$$\exists X, Y \in D_1, X', Y' \in D_2, I(X;Y) = I(X';Y') \rightarrow \quad (3)$$

$$\Delta I_{l=Xk=Y} = I(X;Y) - I(X';Y') = 0 \rightarrow$$

$$X' = sX + c,\ Y' = sY + c$$

We call this kind of genes comparable genes**.** The comparable genes are the ones that are linearly transformed. The value of ΔI_{lk} represents the difference between mutual information of two genes l and k from base dataset D_1 and mutual information of their corresponding transformed genes within D_2. The comparable genes have $\Delta I_{lk}=0$. Eq. 3 could be rewritten for a single gene as:

$$\forall X \in D_1, X' \in D_2, I(X;X) = \quad (4)$$
$$H(X), I(X';X') = H(X') \rightarrow$$
$$\Delta I_{l=Xk=X} = \log(s) \rightarrow X' = sX + c$$

In contrast, incomparability for each pair of different genes can be defined as:

$$\exists X, \forall Y; X, Y \in D_1; \exists X', \forall Y'; \quad (5)$$
$$X', Y' \in D_2, I(X;Y) \neq I(X';Y')$$

Eq. 5 can be used as a robust criterion for omitting noise genes. This equation simply expresses: if there isn't any Y and Y' for a specific X and X' that satisfies Eq. (2), it will be induced that X and X' are not linearly transformed. This means that X, and X' can be omitted and ignored. Due to this fact that the gene compared with all other existing ones is not transformed linearly, and it is disturbed, current expression values of D_1, D_2 around this gene is not valid.

Practically, there are lots of noise sources, and there are few samples compared with the number of genes. Hence, using the above exact equations, few or no linear transformed genes can be reported by the Eqs. (3), (4), and (5). So, it is reasonable to compare two datasets, based on these equations somehow not crisply. For example, majority testing for the lowest or highest, putting some threshold, or even establishing of some fuzzy relations instead of the above crisp equations can be used. In this way, we are able to report those genes which are linearly transformed or near to it. We use simple threshold policy in our experiments that is explained in the last section.

Proposed integration method

If the comparable genes are held only in D_1 and D_2, they can be easily integrated. As D_1 is the base dataset, the integration can be obtained by inverse transformation over D_2. Understanding this, the important job is to find the scale s, and the translation factor c. This subsection determines equations to help us in this way.

By equations 4 and A-4, scaling factor can be calculated. The following formula is obtained by the equation A-4:

$$H(X') - H(X) = \log(s) \rightarrow s = \exp((H(X') - H(X)) \quad (6)$$

The quantities of H(X), H(X') are entropies of X, and X' correspondingly. This equation demonstrates the relationship between scaling factor s with the entropy values of a same gene in both datasets D_1 and D_2.

The translation value c can be found by checking the same gene expression distributions between two data sets D_1 and D_2. Suppose that D_1 is the reference dataset, and D_2 contains those samples that should be modified in a way integratable with samples in D_1. Finding it, the expected value of X in D_1 and X' in D_2, c could be obtained. The following equation shows relationship between the expected values of the expression distributions and the scaling s with translation factor c.

$$c = E\{X'\} - sE\{X\} \quad (7)$$

Two quantities E{X}, E{X'} are the expected values of X, and X' respectively. The values s and c represent scaling and translation factors.

Using s ($s \neq 0$) and c, the expression values of X' from D_2 can be integrated with D_1 as:

$$\tilde{X} = \frac{X' - c}{s} \quad (8)$$

The value X' is a gene expression value in D_2, $\tilde{X}$ is modified gene expression values inversely transformed to integrate with the same gene expression in D_1.

As not the whole of genes can exactly be transformed with same s and c as other ones in the practice, there will be vectors for scaling and translation:

$$s \in \{s_i, 1 \leq i \leq N\}, c \in \{c_i, 1 \leq i \leq N\} \quad (9)$$

If the values of s_i are similar and close to each other, their average can be used instead of different s_i for each gene. The same story stands for c_i. But if the time of process isn't too much important, it would be better to work with each s_i , c_i separately.

ALGORITHM

Our method is expressed in algorithm 1. Based on algorithm the first ΔI_{lk}, is calculated for all genes using

Algorithm 1: The Proposed algorithm for integrating of microarray expression data.

1) Calculate ΔI from samples of one class.
2) Create V by row summation of ΔI.
3) Calculate T.
4) Omit genes lower than T.
5) Calculate s and c for each remained gene.
6) Inverse Translation for each gene.
7) Adding Inverted samples to reference dataset.

Eq. 3 and 4. The indexes l and k represent any two desired genes. Calculating ΔI_{lk} results in forming symmetric matrix as:

$$\Delta I = \{\Delta I_{lk}, 1 \le l \le N, 1 \le k \le N\} = \begin{bmatrix} \log(s_1) & \Delta I_{12} & \cdots & \Delta I_{1N} \\ \Delta I_{21} & \log(s_2) & \cdots & \Delta I_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ \Delta I_{N1} & \Delta I_{N2} & \cdots & \log(s_N) \end{bmatrix}_{N\times N} \quad (10)$$

The value of ΔI_{lk} is defined as an absolute difference between mutual information by Eq. (3), so $\Delta I_{lk}=\Delta I_{kl}$ which is a symmetric matrix. The diagonal value of i-th gene is equal to $\log(s_i)$ calculated by Eq. (4). If the value of ΔI_{lk} in any other locations except those in the diagonal is zero, the l-th and k-th genes are linearly transformed between D_1 and D_2 with the same s and c. If a value in diagonal is zero, the corresponding gene is not scaled. In the best case it is a hope to see a zero matrix, meaning X, X' have identical distributions and no transformation has taken effect. As the calculations in this phase takes place for all combination of two genes, it has quadric complexity $O(N^2)$ regarding N genes.

As it was mentioned earlier, the Eq. (5) describes a gene which should be omitted. Based on this equation, if the l-th gene has $\Delta I_{lk} \neq 0$ for all other genes, the l-th gene shall be omitted. In the matrix of Eq. (10), this can be interpreted as the omittion of those rows that have non-zero values. The more non-zero values the l-th row of the matrix has, the more non-linear characteristics the l-th gene shows. And as the values of ΔI_{lk} are non-negative, so the greater summation the l-th row of the matrix has, the more non-linear specifications the l-th gene illustrates.

The merit of a gene omitted is described by the simple threshold in this research. To calculate the threshold value T, the algorithm performs steps 2, 3 and 4. The second step is about the creation of vector V by ΔI summation in the rows, which is done with the complexity of O(N).

$$V = \left\{ v_i, (1 \le i \le N) v_i = \sum_{j=1}^{N} \Delta I_{ij} \right\} \quad (11)$$

The third step finds those genes that have v_i with greater values than T, and the fourth step omits them. The threshold value T is calculated by Eq. (12). This equation shows that the one- tenth of the greatest row summation is used as the threshold value in this research. The one tenth is the threshold percentage which was chosen by experiments.

$$T = 0.1 \times \mathrm{Max}\{v_i; 1 \le i \le N\} \quad (12)$$

In the fifth step, the values s and c are calculated for the remaining genes by the Eqs. (6) and (7). To find s, an exponential function effects on the diagonal values of ΔI matrix. Based on the calculated s and Eq. (7), the translation factor c is calculated.

Step six deals with finding the inverse transformed expression values for the genes by the Eq. (8), and the step seven ends the process by merging the newly calculated values with corresponding genes in D_1.

The complexity of phases between 3 and 7, aren't greater than O(N). It makes that the overall complexity of the algorithm will be limited in quadric $O(N^2)$ that is suitable in sense of the time to be spent.

EXPERIMENTAL RESULTS

Our experiments fall into two categories: a) finding the threshold percentage, b) performing the whole algorithm on some real datasets.

Finding the threshold percentage

A Random reference dataset D_1 was created with 10 features (i.e. genes) whose distributions were selected to be uniform PDF with different mean μ and standard deviation σ. The μ and σ were selected equal to the index of each gene. Creating the modified dataset D_2, the initial PDF of each gene from the reference dataset linearly is transformed by different s, c. Hence, transformed PDF was sampled to create D_2. The s and c were selected

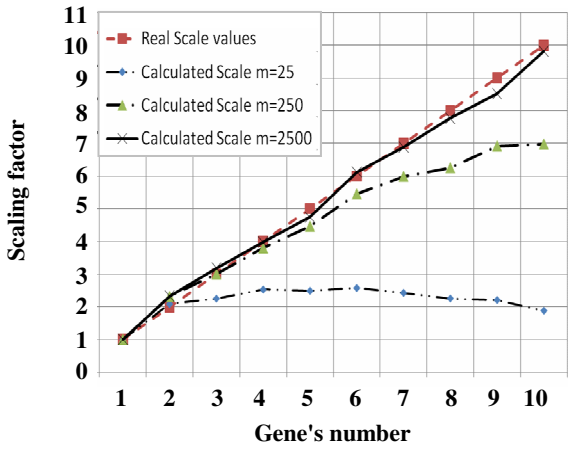

Fig. 3: The results of proposed algorithm in order to find scaling factor (s). Horizontal axis determines the gene number in datasets. Vertical axis shows discovered scaling factor(s). Real effected scaling factors(s) are illustrated by red dashed lined. The other line types and their marks show the discovered (s) which were calculated by different dataset's number of samples (m). The examined values of m are m=25, 250, and 2500. Each marked point on the discovered values (s) is the average of ten independent runs.

equal to the index of each gene. The number of samples in D_1, D_2 which is denoted by m, was equal.

As an example, the fourth feature g_4 in D_1 was created by sampling a uniform PDF with μ=4 and σ=4 that means $g_4 \sim (a= -2.9282, b = +10.9282)$. The distribution of g'_4 in D_2 was calculated by a linear transformation on the PDF of the g_4. Due to the scaling and translation s = c = 4, the g'_4 had μ=20 and σ=16 or in the other word $g'_4 \sim (a = -7.7128, b = +47.7128)$.

In order to discover the threshold percentage, the D_1, D_2 were fed into the software which was implemented to find the s, c without any threshold value. The software was implemented only based on steps 1 and 5 of the algorithm. The procedure of sampling D_1, D_2 and looking for s, c was performed ten times for each dataset's number of samples m and examined by different values m=25, 250, and 2500.

The discovered results of s and c can be seen in Fig. 3, 4 respectively. The horizontal axis shows the corresponding gene number and the vertical axis represents the corresponding studied values which are the scaling factor s in Fig. 3 and the translation values c in Fig. 4. The discovered values of s and c in Figs. 3 and 4 are the average of ten independent runs of the algorithm.

The Figs. 3 and 4 show some facts. The greater value in the scaling factor s and the translation factor c the gene is affected by, the more deviation in finding the correct values the algorithm has. The deviation can be reduced by increasing the dataset's number of samples m.

Although, increasing in m makes the discovered values s and c close to the actual values, the number of samples in a real dataset, is fixed and not more than 250 samples. If the relations between the valid values of s, c m are created, the greatest one will be 0.1(i.e. s=2/m=25). It means that we are mostly sure about the deviation beyond this range. In other word, this technique is capable of discovering valid s and c whose values are less than a tenth of the number of samples.

Integration on real datasets

In order to perform the algorithm to do the integration on the real datasets, the prostate cancer microarray datasets which are publicly available were used. These datasets have been presented in some highly valuable papers. For simplicity, each dataset is represented by the abbreviation of the first author of the paper, such as *Singh* [3], and *Welsh* [20]. The platform of these datasets is Affymatrix HG-95AV2. Table 1 shows the data-sets' specifications in brief.

As it is shown, the datasets aren't equal in their number of genes; so before presenting these datasets within the algorithm, a preparation step is needed in order to balance them. The input datasets and integrated dataset, which is the output of the algorithm, are classified in order to discover the robustness of the algorithm in omitting the noise genes.

Our classification approach for the evaluation is SVM [21] with linear kernel. Despite its complication in adjusting the proper parameters, SVM which works based on machine learning, is a powerful tool not only in the classification tasks but also in the regression and other fields of study. As SVM is a machine learning technique, it is needed to be fed by some samples as the training cases t, and the remaining as the test ones. The Leave One Out Cross Validation (i.e. LOOCV) is used as the training and testing method.

Giving available N samples, LOOCV uses (N-1) samples to build and train the classifier, while the remaining sample inputs are used by the classifier to measure its merit. This process takes place N times and

Table 1: The data-sets' specifications.

Dataset	Number of Probes	Number Of Normal Samples	Number of Tumor Samples	Total Number of Samples
Singh	12600	50	52	102
Welsh	12626	9	24	33

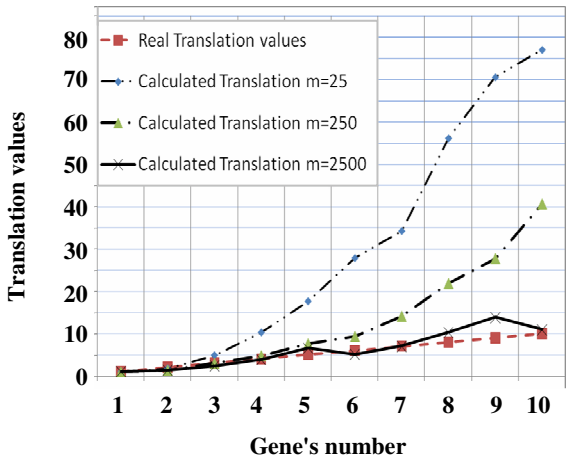

Fig. 4: The results of proposed algorithm in finding proper Translation value. Horizontal axis determines the gene number in data sets vertical axis shows effected Translation values to PDF. Real Translation values are illustrated by dashed lined. For example 4th gene has effected by c = 4. The algorithm has found c = 4.2741.

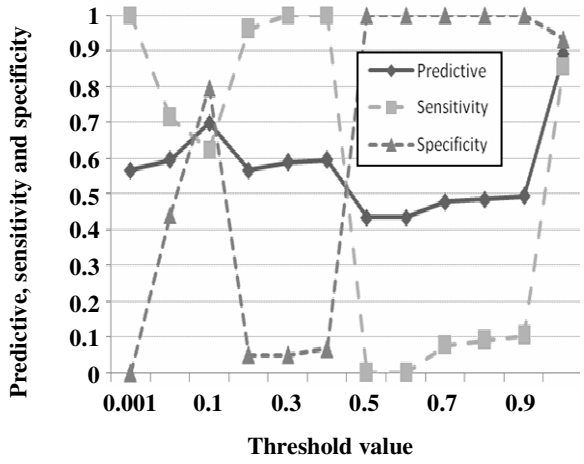

Fig. 5: The results of proposed algorithm with different threshold values. The vertical axis shows the predictive accuracy, sensitivity, and specificity which are a number between 0 and 1. The horizontal axis is threshold values. Each point is reported by LOOCV.

the average of the measured criteria are reported after each time. The evaluated criteria are accuracy, sensitivity, and specificity defined as:

$$\text{Accuracy} = \frac{\text{The number of samples that is correctly predicted}}{\text{The number of total samples}} \quad (13)$$

$$\text{Sensitivity} = \frac{\text{The number of samples that is correctly predicted as tumor}}{\text{The number of tumor samples}} \quad (14)$$

$$\text{Sensitivity} = \frac{\text{The number of samples that is correctly predicted as Normal}}{\text{The number of normal samples}} \quad (15)$$

The behavior of the algorithm with different threshold values is shown in Fig. 5. It can be seen that increasing the threshold value up to 0.1 makes the predictive accuracy better. Due to the few numbers of samples, threshold values greater than 0.1 make the result worse. Deviation and reduction in predictive accuracy is performed based on what has been presented previously. If the integration algorithm is performed without any reduction in genes (T=1), SVM will classify the sample with best predictive accuracy.

The SVM performance over *Singh*, *Welsh*, and the integrated dataset (*Singh*+*Welsh*) which is the output of the new integration algorithm without any threshold for reduction, are illustrated in Fig. 6. It shows that the integrated dataset is classified by SVM with highest predictive accuracy and specificity. And the sensitivity is 0.8571 that is less than the others.

Generally speaking, the integrated dataset is classified better than the others. The two out of three evaluated criteria are satisfied by the proposed integration algorithm. Although, its sensitivity is less than the other, but the most important factor in comparison of different methods is predictive accuracy which is satisfied by proposed integration method. It is due to this fact that the algorithm integrating separate datasets creates more relevant samples than any single dataset.

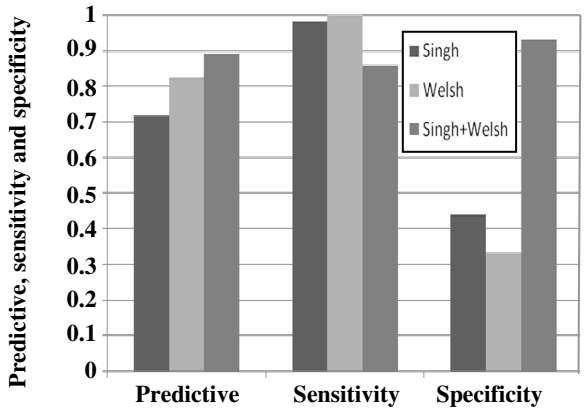

Fig. 6: It shows the classification results over Singh, Welsh, and integrated dataset. The integrated dataset is created by the algorithm without any threshold value. Each result is reported by LOOCV. The vertical axis shows the predictive accuracy, sensitivity and specificity which are between 0 and 1.

CONCLUSIONS

We have presented a new integration method and a validation technique by means of information theory concepts. The method is general for integration of any kind of data whose distributions have been linearly transformed. The validation method is for checking the linearity between two distributions, as the integration method can only cope with linear transformations. Additionally, this article presents new heuristic feature reduction method. The method omits those genes which are not linearly transformed. These methods have been applied in DNA microarray as a special case of their usages. The time complexity of proposed algorithm is in quadric order that is something suitable for practical usages.

APPENDIX A

This assumption will not decrease from the generality of further topics due to this fact that, the number of possible values for each random variable can be increased toward infinite and integral notations can be used instead of sigma form, or one can discreteize these random variables first.

Upon these, we will use some terminology as follow: p(X) and p(Y) are probability mass functions of g_l and g_k in D_1. And p(X') and p(Y') are probability mass functions of g_l, g'_k in D_2, and p(X,Y) is joint probability mass function of g_l, g_k and so on. $p(x_i)$ is probability of a specific event x_i out of an events' list that is related to random variable X and l is number of these events. $p(y_j)$ and $p(x_i,y_j)$ are similarly defined, m is possible number of y_j.

Suppose a given gene as a discrete random variable X that can get different possible values $\{x_1,..,x_l\}$ and $c,s\in\Re$, are scaling and translation factors that scale and transform a given gene's expression distribution uniformly between D_1 and D_2. Thus it can be written as follow:

$$0 \le p(x_i) \le 1, x_i \in X \equiv \{x_1,...,x_1\}, \quad \text{(A-1)}$$

$$\sum_{i=1}^{l} p(x_i) = 1 \xrightarrow{X'=sX+c}$$

$$0 \le p(x_i') \le 1, x_i' \in X' \equiv \{x_1',...,x_{sl}'\},$$

$$\sum_{i=l+c}^{sl+c} p(x_i') = 1 \rightarrow p(x_i') = \frac{p(x_i)}{s}$$

$$0 \le p(y_j) \le 1, y_j \in Y \equiv \{y_1,...,y_m\}, \quad \text{(A-2)}$$

$$\sum_{j=1}^{m} p(y_j) = 1 \xrightarrow{Y'=sY+c}$$

$$0 \le p(y_j) \le 1, y_j \in Y' \equiv \{y_1',...,y_{sm}'\},$$

$$\sum_{j=l+c}^{sm+c} p(y_j') = 1 \rightarrow p(y_j') = \frac{p(y_j)}{s}$$

Lemma1: Linear transformation X'=sX+c over a random variable increases the total ambiguity equal to logarithm of effected scaling factor.

Proof: Entropy of random variable X can be calculated upon its definition [19] and Eq. (1) as follow:

$$H(X) = -\sum_{i=1}^{l} p(x_i)\log(p(x_i)) \xrightarrow{X'=sX+c} \quad \text{(A-3)}$$

$$H(X') = -\sum_{i=l+c}^{sl+c} p(x_i')\log(p(x_i'))$$

Using index shifting, A-1, and A-2 we are able to change it as follow:

$$H(X') = -\sum_{i=1}^{sl} p(x_i')\log(p(x_i')) = \quad \text{(A-4)}$$

$$-\sum_{i=1}^{sl} \frac{p(x_i)}{s}\log(\frac{p(x_i)}{s}) =$$

$$\frac{1}{s}\left[-\sum_{i=1}^{sl} p(x)\log(p(x)) + \sum_{i=1}^{sl} p(x)\log(s)\right] =$$

$$\frac{1}{s}\left[sH(X) + s\log(s)\right] = H(X) + \log(s)$$

Example: We have a dice with six faces, the total ambiguity or entropy is 2.5849(bit). If a special 12 faces dice is designed, the entropy would be equal to 3.5849(bit), equation A-4 is hold with the scaling factor 2**.**

Lemma2: Linear transformations X'=sX+c and Y'=sY+c over two random variables increase the conditional entropy equal to logarithm of effected scaling factor.

Proof: For conditional entropy we can write as follow:

$$H(X \mid Y) = -\sum_{i=1}^{l}\sum_{j=1}^{m} p(x_i, y_j)\log(p(x_i \mid y_j)) = \quad \text{(A-6)}$$

$$-\sum_{i=1}^{l}\sum_{j=1}^{m} p(x_i, y_j)\log(\frac{p(x_i, y_j)}{p(y_j)})$$

$$H(X' \mid Y') = -\sum_{i=1+c}^{sl+c}\sum_{j=1+c}^{sm+c} p(x_i', y_j')\log(\frac{p(x_i', y_j')}{p(y_j')})$$

For joint probability we have the following relationship:

$$p(x_i, y_j) \xrightarrow{X'=sX+c, Y'=sY+c, x_i \in X, x'_i \in X', y_j \in Y, y'_j \in Y'} \quad \text{(A-7)}$$

$$p(x_i', y_j') = \frac{p(x_i, y_j)}{s^2}$$

From joint entropy definition we have:

$$H(X, Y) = H(Y) + H(X \mid Y) = \quad \text{(A-8)}$$

$$\sum_{i=1}^{l}\sum_{j=1}^{m} p(x_i, y_j)\log(p(x_i, y_j))$$

Based on the above equations A-2, A-6, A-7, and using index shifting it could be written:

$$H(X' \mid Y') = -\sum_{i=1}^{sl}\sum_{j=1}^{sm} \frac{p(x_i, y_j)}{s^2}\log(\frac{p(x_i, y_j)}{s.p(y_j)}) = \quad \text{(A-9)}$$

$$\frac{-1}{s^2}\left[\sum_{i=1}^{sl}\sum_{j=1}^{sm} p(x_i, y_j)\log(p(x_i, y_j)) - \right.$$

$$\left.\sum_{i=1}^{sl}\sum_{j=1}^{sm} p(x_i, y_j)\log(s.p(y_j))\right] =$$

$$\frac{1}{s^2}\left[s^2 H(X, Y) + s.\sum_{j=1}^{sm} p(y_j)\log(s.p(y_j))\right] =$$

$$\frac{1}{s^2}\left[s^2 H(X, Y) + s.\sum_{j=1}^{sm} p(y_j)\log(s) + s.\sum_{j=1}^{sm} p(y_j)\log(p(y_j))\right] =$$

$$\frac{1}{s^2}\left[s^2 H(X, Y) + s^2\log(s) - s^2 H(Y)\right] =$$

$$H(X, Y) + \log(s) - H(Y) = H(X \mid Y) + \log(s)$$

Received : March 15, 2010 ; Accepted : Jan. 17, 2011

REFERENCES

[1] Iyer V.R. et al., The Transcriptional Program in the Response of Human Fibroblasts to Serum, *Science*, **283**, 83, (1999).

[2] Choi J.K., Yu U., Kim S., Yoo O.J., Combining Multiple Microarray Studies and Modeling Interstudy Variation, *Bioinformatics*, **19**, p. 84, (2003).

[3] Singh D. et al., Gene Expression Correlates of Clinical Prostate Cancer Behavior, *Cancer Cell*, **1**, p. 203 (2002).

[4] Chen W.B., Zhang C., Liu W.L., An Automatic and Robust Method for Microarray Image Analysis and the Related Information Retrieval for Microarray Databases, "(ICDE 2007) IEEE 23rd International Conference″, 85 (2007).

[5] Katzer F.K.M., Methods for Automatic Microarray Image Segmentation, *IEEE Transactions on nanobioscience*, **2**, p. 202 (2003).

[6] Knudsen S., "Guid to Analysis DNA Microarray Data", John Wiley, (2007).

[7] Conde L., Mateos A., Herrero J., Dopazo J., Unsupervised Reduction of the Dimensionality Followed by Supervised Learning with a Perceptron Improves the Classification of Conditions in DNA Microarray Gene Expression Data, "Neural Networks for Signal Processing″, 77 (2002).

[8] Huang T.-M., Kecman V., Kopriva I., Feature Reduction with Support Vector Machines and Application in DNA Microarray Analysis, "Kernel Based Algorithms For Mining Huge Data Sets", Springer, 95 (2007).

[9] Lai Y., Eckenrode S.E., She J., A Statistical Framework for Integrating Two Microarray Data Sets in Differential Expression Analysis, "17th Asia Pacific Bioinformatics Conference (APBC2009)", (2009).

[10] Jiang H. et al., Joint Analysis of Two Microarray Gene-Expression Data Sets of Select Lung adenocarcinoma marker genes, *BMC Bioinformatics*, **5**, p. 8 (2004).

[11] Yoon Y., Lee J., Park S., Building a Classifier for Integrated Microarray Datasets Through Two-Stage Approach, "BioInformatics and BioEngineering, 2006. BIBE 2006. Sixth IEEE Symposium", 94, (2006).

[12] Kang J., Yang J., Xu W., Chopra P., Integrating Hetrogeneous Microarray Data Sources using Correlation Signatures, *Data Integration in the Life Sciences (DILS),* **3615/2005**, p. 105 (2006).

[13] Xu L., Tan A., Naiman D., Geman D., Winslow R., Robust Prostate Cancer Marker Genes Emerge from Direct Integration of Inter-Study Microarray Data, *Bioinformatics*, **21**, p. 3905 (2005).

[14] Conlon E., Song J., Liu J., Bayesian Models for Pooling Microarray Studies with Multiple Sources of Replications, *BMC Bioinformatics*, **7**, p. 247, (2006).

[15] Hong F., A Comparison of Meta-Analysis Methods for Detecting Differentially Expressed Genes in Microarray Experiments, *Bioinformatics*, **24**, p. 374, (2008).

[16] Xu L., Tan A., Winslow, R. Geman D., Merging Microarray Data from Separate Breast Cancer Studies Provides a Robust Prognostic Test, *BMC Bioinformatics*, **9**, p. 125 (2008).

[17] Borozan I. et al., MAID: An Effect Size Based Model for Microarray Data Integration Across Laboratories and Platforms, *BMC Bioinformatics*, **9**, p. 305 (2008).

[18] Cahan P. et al., List of Lists-Annotated(LOLA): a Database for Annotation and Comparison of Published Microarray Gene Lists, *Gene*, **360**, p. 78 (2005).

[19] Cover T.M., Thomas, J.A., "Elements of information theory", Wiley, (2006).

[20] Welsh J.B. et al., Analysis of Gene Expression Identifies Candidate Markers and Pharmacological Targets in Prostate Cancer, *Cancer Research*, **61**, p. 5974 (2001).

[21] Chang C.C., Lin. C.J., http://www.csie.ntu.edu.tw/ `cjlin/libsvm, (2001).