Non-Negative Matrix and Tensor Factorization Methods for Microarray Data Analysis

Yifeng Li and Alioune Ngom
School of Computer Science
University of Windsor
Windsor, Ontario, Canada N9B 3P4
Email: li11112c@uwindsor.ca; angom@cs.uwindsor.ca

***Abstract*—Microarray technique can monitor the expression level of thousands of genes at the same time. With the recent advances in microarray technology, the expression levels of genes with respect to samples can be monitored synchronically over a series of time points. Such three-dimensional microarray data, termed gene-sample-time (GST) microarray data, are gene expression matrices measured as a time-series. They have not yet received considerable attention and analysis methods such as classification algorithms, bio-marker selection or discovery techniques, and clustering approaches, to name a few, need to be devised specifically to tackle the complexity of GST datasets. Applications of microarrays are in therapeutical, environmental, pharmaceutical, functional genomics, and clinical science research. Non-negative information can benefit the analysis of microarray data. This report investigates the classification and gene selection performance of non-negative matrix factorization (NMF) over gene-sample data. We also extends it to the higher-order version for GST sample data. We have tested and compared our approaches on real gene-sample and GST datasets. Experiments show that NMF and the higher-order NMF can achieve at least comparable performance.**

I. Introduction

DNA microarray technique can monitor thousands of genes in parallel, dramatically accelerate molecular biology experiments and provide a huge amount of data to find co-regulated genes, functions of genes, genetic networks, for instance. There are two types of microarray data: gene-sample data sets, which compile the expression levels of various genes over a set of biological samples; and gene-time data sets, which record the expression levels of various genes over a series of time-points. Both types of data are represented by a two-dimensional (2D) gene expression matrix.

Machine learning methods are used to analyze microarray data. Analysis methods encompass feature extraction and classification in the application of disease prediction, gene selection to find discriminative genes and biomarkers, clustering in the application of finding co-regulated gene patterns, statistical methods to model gene regulatory networks. However, there is a plethora of problems that arise in analyzing microarray data. For instance, the measurements may contain noise due to technical issues in the measuring process, and the expressions of large number of genes are measured over a small number of samples or time points. These problems, among many other problems, substantially affect the performance of analysis algorithms devised for microarray data.

Within the last few years in medical research, the expression levels of genes with respect to biological samples have been monitored synchronically over a series of time-points [2] [1]. This corresponds to a three-dimensional (3D) data set, termed gene-sample-time (GST) microarray data [3]; which can be viewed as a collection of gene-sample data over a series of time-points, or a collection of gene-time data across some samples. GST data can be used to develop models for diagnosing diseases much more precisely than with static microarray data, or to monitor dose or drug treatment responses of patients over time in pharmacogenomics studies [4], or to determine genes or samples patterns, or to find regulatory pathways [3]. There are many problems associated with the analysis of GST data, such as missing values, noise, small number of sample and time points. Furthermore, unlike in two-dimensional microarrays, a gene or sample in a GST array is a matrix rather than a vector, and therefore GST require special methods for its analysis. Computational analysis of GST data are therefore much more difficult than their two-dimensional counterparts. All these problems, among many other problems, substantially affect the effectiveness and efficiency of analysis algorithms devised for GST data.

For sample classification, it is necessary to reduce the dimension of the data in order to avoid the problems arising from the *curse of dimensionality*. *Linear dimensionality reduction* (LDR) is a widely used linear algebraic technique mainly including *Linear dimension analysis* (LDA), *principal component analysis* (PCA), *singular value decomposition* (SVD), *independent component analysis* (ICA) and *non-negative matrix factorization* (NMF). NMF has attractions of simple implementation, good interpretation, sparsity, and using non-negative information, for its applications in microarray data analysis. The gene expression intensities and ratios are naturally non-negative. NMF becomes popular after [5]. Given a gene-sample dataset $\boldsymbol{X}$ with m genes and n samples. Matrix $\boldsymbol{X}$ can be factorized into two non-negative factors, as follows

$$\boldsymbol{X}_{m\times n} \approx \boldsymbol{A}_{m\times r}\boldsymbol{Y}_{r\times n} \quad \boldsymbol{X},\boldsymbol{A},\boldsymbol{Y} \geq 0, \tag{1}$$

where $r \leq \min(m,n)$, A and Y are the basis matrix and the coefficient matrix, respectively. Multiplicative update rules [6] [7] are widely used algorithms and are simple to implement. Other algorithms, for example *alternating least squares* (ALS), are reviewed in [7]. A novel idea of feature selection is

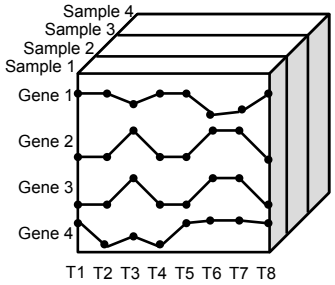

Fig. 1. An Example of a GST Dataset.

that we can analyze the basis matrix to select discriminative features, instead of analyzing the training set directly. In this report, we describe an unsupervised feature extraction methods based on NMF in order to classify microarray samples data. We also investigate the possibility of using NMF to select discriminative genes.

In multilinear algebra, a tensor of order d [8] is a d-dimensional array, and tensor algebra is the extension of vector and matrix algebra to order-d tensors. Since a GST microarray data is naturally an order-3 tensor (see Fig. 1), therefore, known theories from tensor algebra can be used to analyze such data rather than performing matrix operations on a *matricized* representation of the GST data (see Fig. 2 for a matricized example). *Tensor decomposition* is an extension of matrix factorization to tensor data and attempt to find a smaller representation describing the initial tensor data. Matrix factorizations, such as SVD, ICA, and NMF, have been extended to *higher-order SVD* (HOSVD) [9], *multilinear ICA* (MICA) [10], and *higher-order NMF* (HONMF) [11], respectively, for tensor data. Tensor decomposition is investigated in the context of sample classification. Since the original GST data are non-negative, our approach is to perform unsupervised HONMF based *multilinear dimension reduction* (MLDR) method in order to extract a small sets of discriminative and non-negative features, and then perform sample classification in the reduced space. As far as we know, this report is the first attempt at using tensor methods for classifying GST data, (see the next section for their applications on other types of microarray data).

This report is organized as follows. Related works on NMF and tensor decomposition are reviewed next section. NMF based classification and gene selection are described in section IV and section V, respectively. Section VI introduces our proposed HONMF tensor decomposition based classification methods. Finally, we draw our conclusions.

II. Related Works

PCA, SVD, and ICA have been used to extract features from microarray data [12], [13]. NMF has been used to cluster samples or genes [14] [15]. It has also been used in classification problems such as, musical instrument classification [7], face recognition [7], and microarray data classification. [16] used correlation coefficient to decide which class the extracted features (by ICA and NMF) characterize. [17] proposed discriminative mix models to classify non-negative microarray gene expression data. The samples reduced by a sparse NMF are the input of the mix models. [18] used NMF or a sparse NMF to extract features, and employed SVM classifier in the feature space. [19] proposed a ranking method based on a sparse NMF in order to improve the prediction accuracy.

Ref. [1] proposed an *integrated Bayesian inference system* (IBIS) to select triplets of genes for classifying INFβ samples (a GST microarray data) but using only the first time point, and thus did not benefit from (nor consider) the full GST data. Ref. [4] used *support vector machines* based on *dynamical systems kernels* (denoted by dsSVM in this paper) to classify INFβ samples. Since each GST data sample is represented by a matrix, it is not appropriate to use the kernels which take vectorial inputs, such as, the *radial basis functions* (rbf). Dynamical systems kernels accept matrix inputs and take into account the temporal information. Two samples can be modeled by two separate *linear time invariant* (LTI) dynamical systems $X = (P, Q, R, S, x_0)$ (where x_0 is a vector, and P, Q, R, and S are matrices estimated by a SVD based approach) and $X' = (P', Q', R', S', x_0')$. The dynamic systems kernel between X and X' is defined as

$$k(X, X') = x_0^{\mathrm{T}} M_1 x_0' + \frac{1}{\mathrm{e}^{\lambda} - 1}[\mathrm{trace}(SM_2) + \mathrm{trace}(R)] \quad (2)$$

where M_1 and M_2 satisfy the Sylvester equation [4], and λ is a positive parameter of the kernel. Ref. [20] devised *generative hidden Markov models* (GenHMMs) and *discriminative HMMs* (DiscHMMs) approaches for classifying INFβ samples. Samples from the same class are used to train a GenHMM. Samples from all classes are used to train a DiscHMM, for each class. Then, in both methods, a test sample is assigned to a class based on maximum conditional likelihood. Baum-Welch algorithm is used to estimate the parameters of the models. For DiscHMMs, a backward gene selection method is first performed to find a small number of discriminative genes before training the models. [21] proposed a robust constrained mixture estimation approach to classify the INFβ data. This approach combines the constrained clustering method with a mixture estimation classification framework. Subdivision of classes and mislabeled samples can be investigated by this approach. During training, negative constraints were restricted on pairs of samples. The constrained mixture model, with linear HMMs, as components, is optimized by an EM algorithm. The supervised version of this approach (*HMMConst*) only uses training set in the estimation of parameters, while the semi-supervised version (HMMConstAll) uses all data. The emission probability for each state is modeled by mixtures of multivariate Gaussians for patient expression values, noise, and missing values, respectively. In order to select genes contributing to classification, a HMM based gene ranking method is used. Each component of the mixture model is assigned to a class. When testing, a test sample is assigned to a class according to the maximum entry in their posterior distribution.

Ref. [22] applied HOSVD on an order-5 tensor data for

face recognition. In the training phase, a basis tensor for a certain view, illumination, and expression is obtained through HOSVD and then is matricized to a basis matrix in order to obtain vector of each training sample. In the testing, the coefficient vector of a testing sample is obtained through a linear projection approach using the basis matrix. A 1-*nearest neighbor* (1-NN) classifier is used to determine the class labels of the testing samples. Ref. [23] used HOSVD to analyze the integration of DNA microarray data from different studies. They create a tensor dataset of order 3 by combining three gene time-series microarray datasets from yeast cell-cycle studies, and then decompose the tensor by HOSVD. The resulting core tensor obtained from the decomposition contains the significant features representing important biological experimental phenomena. Ref. [24] devised two different approaches based on HOSVD decomposition to classify a dataset of handwritten digits represented as a an order-3 tensor data. HOSVD is used to extract small feature sets that explains the original data but the methods differ in how the core basis tensors are obtained (i.e., either from each class separately, or from the whole data) and in how the class of a test sample is predicted (i.e, either by regression or by projection). Ref. [10] extended ICA to MICA, and used it for extracting features to be used in face recognition. Initially, facial images are vectorized then represented as an order-3 tensor data. MICA is employed to decompose this tensor into factors containing important facial features. A test sample is then multilinearly (rather than linearly, as in [22]) projected into the space spanned by the obtained core basis tensor and a nearest-neighbor classifier using cosine similarity measure is employed to predict the class of the test sample. Ref. [25] also applied MICA decomposition to classify integrated tumor gene expression data from different studies. Their working order-3 tensor is a combination of three gene-sample tumors datasets. Two core basis tensors are obtained via MICA decomposition, separately over training samples and test samples. A SVM classifier is then trained on the matricized version of the core tensor obtained from the training sample and is validated using the core tensor generated from the test data.

III. Notation

Hereafter, we use the following notations in the rest of the paper:

- A matrix is denoted by a bold capital letter, e.g. $\boldsymbol{A}$.
- A (column) vector is denoted by a bold lowercase letter, e.g. $\boldsymbol{a}$.
- A bold lowercase letter with a subscript, $\boldsymbol{a}_i$, denotes the i-th column vector in matrix $\boldsymbol{A}$.
- The italic lowercase letter with two subscripts, a_{ij}, is the (i,j)-th scalar element of matrix $\boldsymbol{A}$.
- A boldface Euler script, e.g. $\boldsymbol{\mathcal{X}}$, denotes an order-3 tensor. That is $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$.
- $\boldsymbol{X}_{(n)}$ denotes the matrix obtained through the mode-n matricization of the tensor $\boldsymbol{\mathcal{X}}$. Columns of $\boldsymbol{X}_{(n)}$ are the mode-n fibers of tensor $\boldsymbol{\mathcal{X}}$. A mode-n fiber is a vector defined through fixing every index but the nth index. This is the extension of matrix row and column in tensor algebra. $\boldsymbol{X}_{(1)}$ therefore denotes the matrix of size $I \times JK$, unfolded in mode-1 of $\boldsymbol{\mathcal{X}}$, that is $\boldsymbol{X}_{(1)} = [\boldsymbol{X}_{(1)1}, \boldsymbol{X}_{(1)2}, \cdots, \boldsymbol{X}_{(1)K}]$. See Fig. 2 as an example.
- $\boldsymbol{X}_{(1)p}$ denotes the p-th frontal slice of $\boldsymbol{\mathcal{X}}$, of size $I \times J$.
- The (i,j,k)-th scalar element of $\boldsymbol{\mathcal{X}}$ is denoted by x_{ijk}.

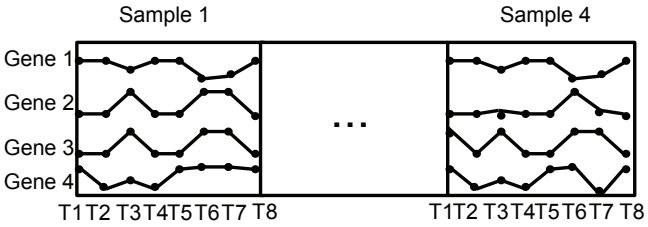

Fig. 2. The Mode-1 Matricization of the Tensor in Fig. 1

Also, $\boldsymbol{A} \otimes \boldsymbol{B}$ denotes the Kronecker tensor product [8] of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.

The mode n product of a tensor $\boldsymbol{\mathcal{X}}$ and a matrix $\boldsymbol{A}$, written as $\boldsymbol{\mathcal{X}} \times_n \boldsymbol{A}$, is:

$$\boldsymbol{\mathcal{X}} \times_n \boldsymbol{A} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N} a_{j i_n}, \tag{3}$$

where $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ and $\boldsymbol{A} \in \mathbb{R}^{J\times I_n}$. This results in a tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1\times\cdots I_{n-1}\times J\times I_{n+1}\cdots\times I_N}$.

IV. NMF Based Classification

This work will appear in the proceeding of the 2010 IEEE International Conference on Bioinformatics & Biomedicine [37].

Features must not been extracted from the overall data $\boldsymbol{X}$, because it would result in overfitting problem. There must be two steps – training on training set to generate new features and feature space, and testing on test set to project the test samples from the original space into the feature space. Given a 2-dimensional gene-sample dataset $\boldsymbol{X}$ of size $m \times n$, with samples in columns, it should be split into independent training set $\boldsymbol{X}^{\text{train}}_{m\times p}$ with p samples and test set $\boldsymbol{X}^{\text{test}}_{m\times(n-p)}$ with $n-p$ samples. First, the training set X^{train} is decomposed as follows by NMF:

$$\boldsymbol{X}^{\text{train}}_{m\times p} \approx \boldsymbol{A}^{\text{train}}_{m\times r} \boldsymbol{Y}^{\text{train}}_{r\times p}, \quad \boldsymbol{X}^{\text{train}}, \boldsymbol{A}^{\text{train}}, \boldsymbol{Y}^{\text{train}} \geq 0, \tag{4}$$

where $r \leq \min(m,p)$. Multiplicative update rule based algorithm in [14] is employed in this paper. The reason why NMF can be used in feature extraction is that a sample $\boldsymbol{x}_i$ can be approximately represented by a linear combination of the columns of $\boldsymbol{A}^{\text{train}}$ with coefficients in the ith row of $\boldsymbol{Y}^{\text{train}}$. $\boldsymbol{A}^{\text{train}}$ is the basis matrix and will be used in the test step. Each column of $\boldsymbol{A}^{\text{train}}$ is the extracted features, called *metagene* [14]. All the metagenes span the feature space, called *NMF space*. $\boldsymbol{Y}^{\text{train}}$ is the coefficients matrix. Its column $\boldsymbol{y}_i$ is the representation of $i-$th sample in the NMF space. After that, the test samples should be projected into the NMF space to obtain these presentations in this space, as follows:

$$\boldsymbol{X}^{\text{test}}_{m\times(n-p)} \approx \boldsymbol{A}^{\text{train}}_{m\times r} \boldsymbol{Y}^{\text{test}}_{r\times(n-p)}, \boldsymbol{X}^{\text{test}}, \boldsymbol{A}^{\text{train}}, \boldsymbol{Y}^{\text{test}} \geq 0 \tag{5}$$

It should be noted that this is a *non-negativity constrained least squares* (*NLS*) problem. Three projection algorithms are introduced, as follows, to solve this problem. We can use a multiplicative updating based projection method [7]. In this method, $\boldsymbol{Y}^{\text{test}}$ is iteratively updated by the multiplicative update rule while keeping $\boldsymbol{A}^{\text{train}}$ intact. The second projection method is pseudo-inverse based method [19]. The test samples is projected by a transformation matrix $\boldsymbol{A}$, as follows: $\boldsymbol{Y}^{\text{test}} = \boldsymbol{A}^{\text{train}\,\dagger}\boldsymbol{X}^{\text{test}}$, However, the drawback of this method is that the non-negative constraint of $\boldsymbol{Y}_{\text{test}}$ is violated. The improved version of this method is to enforce the negative values in Y^{test} to zeros, while holding the non-negative values [7]. This method is simple and fast, but has large fitting error. The third projection method is to use an active set method to solve Eq.(5) [26].

After the dimension of the samples is reduced and discriminative information is captured, the last step is to learn and test in the NMF space. $\boldsymbol{Y}^{\text{train}}$ is used to train a classifier, and $\boldsymbol{Y}^{\text{test}}$ is used to test the prediction performance. In this paper, due to efficiency, simplicity, and availability for multi-class problems, k-NN is employed to classify samples in NMF space. With respect to different projection methods in the test stage, we have three unsupervised feature extraction and classification approaches, named uNMFmu, uNMFpi, uNMFnls, respectively.

V. NMF Based Gene Selection

This work is partly published in the proceeding of the 2010 IEEE International Conference on Bioinformatics & Biomedicine [37]. We are preparing a manuscript titled "A New Approach to Feature Selection Using Linear Dimensionality Reduction". This paper will be submitted to the 1st IEEE International Conference on Computational Advances in Bio and Medical Sciences [41]. Computational intelligence will be used to search the optimal subset of genes with respect to our new criterion. This work will be submitted to the 2011 IEEE Congress on Evolutionary Computation [42], and the extended version will be submitted to Pattern Recognition Letter or Neurocomputing.

NMF can also be used to select discriminative genes while filtering out redundant genes. [19] proposes a gene ranking method as follows. Elements of A^{train} less than a fixed threshold are set to 1, and the number of ones in each row is the score of the corresponding gene. Genes are then sorted in decreasing order and the top t genes are selected. In order to investigate the biological pathways, [15] also proposes a gene ranking method in which a gene is scored as follows

$$Gene\_score(i) = 1 + \frac{1}{\log_2(r)} \sum_{j=1}^{r} p(i,j) \log_2 p(i,j), \quad (6)$$

where $p(i,q) = \frac{A^{\text{train}}[i,q]}{\sum_{j=1}^{r} \boldsymbol{A}^{\text{train}}[i,j]}$. The classification performance of this measure is still not clear. For simplicity, we call the first criterion ZH , and the second IE. In both of the criteria, $\boldsymbol{A}^{\text{train}}$ is generated by a sparse NMF [15].

The aim of feature subset selection is to identify a subset of r features, where $r \ll m$, such that the subset of features is as efficient as possible. The efficiency is quantified by a measurement on a specific classifier and a validation method (e.g. cross-validation). More formally, we can use a classifier or an objective function as follows:

$$f([x_{i_1}, x_{i_2}, \ldots, x_{i_r}]) : R^{r \times n} \rightarrow R\,, \quad (7)$$

where $1 \leq i_1 < i_2 < \ldots < i_r \leq n$.

For the sake of simplicity in the notation we use $\mathbf{A}$ instead of $\mathbf{A}_{\text{train}}$. In accordance with the above discussion, our idea is to find an unordered *standard basis vector matrix* $\mathbf{C} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_r]$ which is closest to $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_r]$. $\mathbf{A}$ can be obtained by the LT or MF method. Each column of $\mathbf{C}$ is a standard basis vector (coordinate vector) containing a single value "1". Also, the columns and rows of $\mathbf{C}$ must be distinct.

We can give a definition of *unordered standard coordinate* (USC) basis matrix: if a matrix $\mathbf{C}_{m \times n}$ satisfies the following constraints,

1) $\mathbf{C}_{ij} = 0$ or 1,
2) $\sum_{i=1}^{m} \mathbf{C}_{ij} = 1$,
3) $\sum_{j=1}^{n} \mathbf{C}_{ij} = 0$ or 1,

then $\mathbf{C}$ is a USC basis. If $\mathbf{C}$ is a USC basis, an obvious property of it is $m \geq r$. Based on the definition of basis: a basis of the space $\mathbf{S}$ of R^m, is a collection of linearly independent vectors $\mathbf{V}$ that spans $\mathbf{S}$. Thus, we can use matrix angle to measure the closeness of a pair of bases. The aim of seeking the closest USC basis $\mathbf{C}_{m \times r}$ to a basis $\mathbf{A}_{m \times r}$ is equivalent to

$$\arg_{\mathbf{C}} \max\{f(\mathbf{C}) = \cos(\mathbf{A}, \mathbf{C})\}. \quad (8)$$

VI. Tensor Based Classification

The general tensor decomposition based classification approaches are submitted to the Bioinformatics Journal [34], and is published in the proceeding of the 7th International Meeting on Computational Intelligence Methods for Bioinfomatics and Biostatistics [35]. Our proposed methods are compared with other methods in our paper that is submitted to the 2011 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology [36]. Our paper that extends NMF based classification to HONMF based classification will appear in the proceeding of the 2010 IEEE International Conference on Bioinformatics & Biomedicine [37].

We extended the above NMF based approach into the higher-order version for GST data. Tensor decomposition mainly includes PARAFAC and Tucker decompositions [8]. HONMF is a tucker3 decomposition with non-negativity constraints. Fig. 3 is an example of Tucker3 decomposition. HONMF factorizes a non-negative tensor $\boldsymbol{\mathcal{X}}_{I \times J \times K}$ into a non-negative core tensor $\boldsymbol{\mathcal{C}}_{P \times Q \times R}$ and 3 non-negative mode

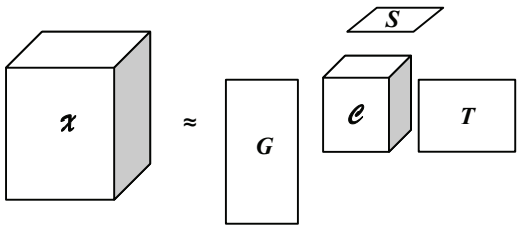

Fig. 3. An Example of Tucker3 Decomposition of the Tensor in Fig. 1.

matrices (factors) $\boldsymbol{G}_{I\times P}$, $\boldsymbol{T}_{J\times Q}$, and $\boldsymbol{S}_{K\times R}$ as follows:

$$\begin{aligned}\mathcal{X} &\approx \mathcal{C} \times_1 \boldsymbol{G} \times_2 \boldsymbol{T} \times_3 \boldsymbol{S} = [\![\mathcal{C}; \boldsymbol{G}, \boldsymbol{T}, \boldsymbol{S}]\!] \\ &= \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} c_{pqr}\boldsymbol{g}_p \circ \boldsymbol{t}_q \circ \boldsymbol{s}_r, \end{aligned} \tag{9}$$

where P,Q, and R are the number of mode vectors (or ranks) of the mode matrices $\boldsymbol{G}$, $\boldsymbol{T}$, and $\boldsymbol{S}$, respectively. $\boldsymbol{a} \circ \boldsymbol{b}$ is the outer product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. In light of Eq. 9, it is clear that an element of core tensor $\mathcal{C}$ indicates the degree of interaction among the corresponding mode vectors from different mode matrices. Algorithm based on multiplicative update rules can converge to local optimum [11].

Next, we describe our unsupervised dimension reduction approaches based on HONMF. Let $\mathcal{X}$ be a training set, from a GST dataset, with I genes, J time points, and K samples. Through Eq.(9), we can obtain

$$\mathcal{X} \approx \mathcal{B} \times_3 \boldsymbol{S}, \tag{10}$$

where $\mathcal{B} = \mathcal{C} \times_1 \boldsymbol{G} \times_2 \boldsymbol{T}$ is a non-negative tensor. Making use of multilinear operations, we have

$$\begin{aligned}\boldsymbol{X}_{(1)} &\approx \boldsymbol{I}_G \boldsymbol{B}_{(1)} (S \otimes \boldsymbol{I}_T)^T \\ &= [\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_R] \begin{bmatrix} \boldsymbol{s}_{11}\boldsymbol{I}_T & \cdots & \boldsymbol{s}_{k1}\boldsymbol{I}_T \\ \vdots & \vdots & \vdots \\ \boldsymbol{s}_{1R}\boldsymbol{I}_T & \cdots & \boldsymbol{s}_{kR}\boldsymbol{I}_T \end{bmatrix}, \end{aligned} \tag{11}$$

where $\boldsymbol{I}_G$ and $\boldsymbol{I}_T$ are identity matrices of sizes $I \times I$ and $J \times J$, respectively. Thus the k-th frontal slice of $\mathcal{X}$, that is, the k-th sample, can be fitted by the additive summation of the frontal slices of $\mathcal{B}$:

$$\boldsymbol{X}_{(1)k} \approx \sum_{i=1}^{R} \boldsymbol{B}_r \boldsymbol{s}_{kr}, \tag{12}$$

where the coefficients are in the k-th row of $\boldsymbol{S}$. Thus, $\mathcal{B}$ is the basis matrix for the samples and $\boldsymbol{S}$ is the encoding matrix. We can define the matrix space spanned by $\mathcal{B}$ as the *HONMF feature space*, and $\boldsymbol{s}_k$ as the non-negative *representation* of the k-th sample in the feature space. These non-negative matrix slices of $\mathcal{B}$ are the *HONMF features*. This reduces the original sample slice to a vector $\boldsymbol{s}_k$ in the feature space.

In the test phase, each test sample $\boldsymbol{Y}_l$ is projected into the HONMF feature space. $\boldsymbol{Y}_l$ should also be an additive linear combination of the basis matrices in $\mathcal{B}$:

$$\boldsymbol{Y}_l = \sum_{r=1}^{R} \boldsymbol{B}_r \alpha_r, \tag{13}$$

where $\mathcal{B}$ is obtained in the training step and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_R]^T$ is the representation of $\boldsymbol{Y}_l$ in the feature space. Finding α is equivalent to solve the following generalized least squares problem:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{Y}_l - \sum_{r=1}^{R} \boldsymbol{B}_r \alpha_r\|_F^2, \quad \boldsymbol{\alpha} \geq 0, \tag{14}$$

where $\| \bullet \|_F$ is Frobenius norm of a matrix. The general solution to this problem is $\alpha_r = \frac{<\boldsymbol{Y}_l, \boldsymbol{B}_r>}{<\boldsymbol{B}_r, \boldsymbol{B}_r>}$ [24] [27], where $< \bullet, \bullet >$ is the inner product of two matrices. For different test samples, we put $\boldsymbol{\alpha}$'s in the corresponding rows of a non-negative coefficient matrix $\boldsymbol{A}$.

Alternatively, given the test samples $\mathcal{Y}$, we can fix $\mathcal{C}$, $\boldsymbol{G}$, and $\boldsymbol{T}$ to calculate the coefficient matrix $\boldsymbol{A}$ of $\mathcal{Y}$. We need to find $\boldsymbol{A}$ that satisfies

$$\mathcal{Y} \approx \mathcal{C} \times_1 \boldsymbol{G} \times_2 \boldsymbol{T} \times_3 \boldsymbol{A}. \tag{15}$$

Higher-order orthogonal iterations (HOOI) is an ALS Tucker3 decomposition algorithm which restricts orthogonality on factors [8]. For HOSVD and HOOI, the mode matrices are orthogonal and $\boldsymbol{A}$ is the R leading left singular vectors of $\boldsymbol{Z}_{(3)}$. $\boldsymbol{Z}_{(3)}$ is matricized from $\mathcal{Z}$ which is calculated by the following equation:

$$\mathcal{Z} = \mathcal{Y} \times_1 \boldsymbol{G}^T \times_2 \boldsymbol{T}^T. \tag{16}$$

But, for HONMF, the constraint on the mode matrices is non-negativity rather than orthogonality. Instead of deriving an equation similar to Eq.(16), we can just update $\boldsymbol{A}$ iteratively using its multiplicative update rule in [11], while keeping $\mathcal{C}$, $\boldsymbol{G}$, and $\boldsymbol{T}$ constant.

These feature extraction methods are termed multilinear dimension reduction (MLDR), extended from LDR in linear algebra. Once $\boldsymbol{A}$ is obtained, we do not need to learn on the training samples and classify the test samples represented by the matrices. Instead, any classifier can be trained and tested in the feature space where a sample is represented by a vector, that is any classifier can be trained on $\boldsymbol{S}$ and predict the classes of the test samples (the rows of $\boldsymbol{A}$). The same scheme is also implemented using HOSVD and HOOI on the purpose of comparison. With respect to different projection methods in the test phase, these tensor based classification methods are denoted by uHONMFgls, uHOSVDgls, uHOOIgls, and uHONMFmu, uHOSVDtf, uHOOItf.

The unsupervised MLDR techniques above can be modified in a supervised manner. Let m be the number of classes in the data. The idea is to first partition the training set into m subsets $\mathcal{X}^1$, $\cdots$, $\mathcal{X}^m$, where each subset $\mathcal{X}^i$ contains only samples of class i. Next, m core tensors $\mathcal{B}^1$, $\cdots$, $\mathcal{B}^m$ are obtained through decomposition using Eq.(10). The resulting basis matrices are then normalized using the Frobenius norm. A normalized test sample can be fitted by these basis tensors, respectively, through Eq.(14). This sample is assigned to the class where the minimal fitting residual is obtained. For simplicity, we denote the supervised version of HONMF, HOSVD, and HOOI based classification

methods by sHONMF, sHOSVD, and sHOOI. This supervised decomposition approach is described in [24] for hand written recognition using HOSVD.

VII. Experiments

A. NMF Based Feature Extraction and Classification

We test the performance of the NMF based methods over three binary-class and two multi-class gene-sample datasets as summarized in Table I. Gene selection is not applied in this part. PCA is used as a benchmark method. The Euclidean distance based 3-NN is used. 7-fold *cross-validation* (CV) is employed to partition a dataset into training sets and test sets. The optimal number of metagenes r is searched through line search according to the classification performance. The mean performances of 20 runs are shown in Table II. *Specificity*, *sensitivity*, and *accuracy* are defined by $\frac{TN}{TN+FP}$, $\frac{TP}{TP+FN}$, and $\frac{TN+TP}{TN+TP+FN+FP}$, respectively, where TN, TP, FN, FP are the numbers of true negative, true positive, false negative, and false positive samples, respectively. It can be seen that the NMF based approaches obtains better accuracies in general. Also, the NMF based approaches tend to extract a small number of features.

TABLE I
Gene-Sample Datasets

Dataset	#Classes	#Genes	#Samples
Binary-class Leukemia [14]	2	5000	27+11=38
Medulloblastoma [14]	2	5893	39+21=60
Colon [28]	2	2000	40+22=62
Multi-class Leukemia [14]	3	5000	19+8+11=38
SRBCT [29]	4	2308	23+8+12+20=63

TABLE II
Performance of Various NMFs and PCA.

Data	Methods	Optimal r	Specificity	Sensitivity	Accuracy
Binary Class Leukemia	uNMFmu	3	**0.993**	**0.925**	**0.977**
	uNMFpi	**2**	0.980	0.907	0.959
	uNMFnls	**2**	0.980	0.907	0.959
	PCA	4	0.989	0.879	0.957
Medulloblastoma	uNMFmu	6	0.933	**0.793**	**0.895**
	uNMFpi	6	0.912	0.661	0.839
	uNMFnls	6	0.910	0.668	0.839
	PCA	**5**	**0.985**	0.418	0.837
Colon	uNMFmu	**5**	0.788	0.897	0.858
	uNMFpi	6	**0.814**	0.917	**0.881**
	uNMFnls	6	0.810	0.909	0.874
	PCA	10	0.706	**0.950**	0.863
Multi-class Leukemia	uNMFmu	**3**	-	-	**0.973**
	uNMFpi	**3**	-	-	0.950
	uNMFnls	**3**	-	-	0.953
	PCA	4	-	-	0.935
SRBCT	uNMFmu	**6**	-	-	0.954
	uNMFpi	12	-	-	0.942
	uNMFnls	10	-	-	0.942
	PCA	19	-	-	**0.958**

B. NMF Based Gene Selection

We also compared the performance of the gene selection criterion IE with ZH. The classification performance is shown in Table III. Here, IE (or ZH) indicates gene selection criterion IE (or ZH) is used and then 3-NN classification is performed. IE+NMF (or ZH+NMF) means gene selection criterion IE (or ZH) is employed prior to uNMFmu. We can find that, in general, gene selection criterion IE is much effective than ZH. With hundreds of selected genes by criterion IE, the accuracies do not decrease dramatically on some datasets, while the accuracies are increased on some datasets, comparing to these in Table II.

TABLE III
Comparison of Two NMF Based Gene Selection Methods

Data	Methods	Number of Selected Genes					
		100	200	300	400	500	600
Binary Class Leukemia	ZH	0.780	0.775	0.771	0.786	0.782	0.767
	ZH+NMF	0.718	0.746	0.745	0.732	0.765	**0.793**
	IE	0.912	0.929	0.930	0.939	0.949	0.939
	IE+NMF	0.925	0.939	0.939	0.956	**0.975**	0.961
Medullo-blastoma	ZH	0.547	0.562	0.583	0.634	0.668	**0.678**
	ZH+NMF	0.604	0.554	0.598	0.591	0.585	0.612
	IE	0.668	0.692	0.691	0.691	0.704	0.719
	IE+NMF	0.673	0.713	0.722	0.702	0.716	**0.743**
Colon	ZH	0.799	0.791	0.797	0.803	0.811	0.826
	ZH+NMF	0.831	0.821	0.834	0.836	0.844	**0.854**
	IE	0.827	0.848	0.857	**0.875**	0.868	0.864
	IE+NMF	0.832	0.851	0.860	0.870	0.874	0.870
Multi-class Leukemia	ZH	0.501	0.579	0.585	0.589	0.600	**0.627**
	ZH+NMF	0.468	0.483	0.486	0.455	0.447	0.450
	IE	0.907	0.927	0.942	0.943	0.956	0.933
	IE+NMF	0.921	0.925	0.912	0.951	0.959	**0.968**
SRBCT	ZH	0.968	0.976	0.980	0.979	**0.980**	0.979
	ZH+NMF	0.951	0.972	0.976	0.975	0.977	0.977
	IE	0.982	0.979	0.977	0.977	**0.982**	0.981
	IE+NMF	0.980	0.980	0.978	0.974	0.977	0.971

C. Tensor Decomposition Based Feature Extraction and Classification

We used our HONMF approaches to predict good or bad responders to *Interferon beta* (INFβ) treatments. INFβ is a protein used for treating patients afflicted with multiple-sclerosis (MS). Some MS patients after INFβ therapy do not respond well to the drug and the reasons are still not clear [1]. Baranzini *et al.* [1], among others researchers, applied Bayesian learning method on a clinical time-series dataset to determine pairs or triplets of genes that can discriminate between bad and good INFβ responders. The initial dataset is a GST data sampled from 53 MS patients who were initially treated with equal dose of INFβ over a time period. This initial dataset contains the expression measurements for 76 genes at 7 time points for each patient, with 31 patients responding well and the remaining 22 responding bad to the treatment. This dataset contains genes with missing expression measurements at some time points. Those genes and corresponding samples were removed from our analysis, and hence, the resulting "complete" data contains 53 genes and 27 samples (18 good responders and 9 bad responders).

Our proposed HONMF based methods were applied to this INFβ data. They were compared with GenHMMs and DiscHMMs approaches [20], dsSVM [4] approaches (described in the related work section), uHOSVDls, uHOOIls, uHOSVDtf, uHOOItf, and the supervised tensor based methods. We also run our experiment on a SVM classifier based on rbf kernels (rbfSVM) over the vectorized samples. Due to technical problems, we can not run HMMConst and HMMConstAll

[21]. Matlab codes for HONMF and HOOI from [11] and [30], respectively, are used. We used 3-NN with Euclidean distance in the classification phase. Due to the small number of samples, 9-fold CV is employed. All our methods are performed for 20 runs, and the mean performances and standard deviations are reported in Table IV. Specificity is the prediction accuracy of the good responders, while sensitivity is for the bad responders. The parameter for GenHMMs and DiscHMMs is the number of selected genes; absence of such parameter means gene selection is not used. The first parameter of dsSVM is the number of hidden states, and the second one is the value of λ. The parameter of rbfSVM is the value of λ in the rbf function. The parameter of the tensor decomposition based approaches are rank-(P, Q, R). All of the parameters in the above approaches are obtained through line/grid search with respect to the classification performance.

TABLE IV
ACCURACY ON COMPLETE INF-BETA DATA

Methods	Param.	Specificity	Sensitivity	Accuracy
GenHMMs	-	0.861±0.036	0.556±0.000	0.759±0.044
DiscHMMs	-	0.861±0.036	0.556±0.000	0.759±0.044
GenHMMs	7	0.861±0.063	0.561±0.008	0.761±0.047
DiscHMMs	7	0.861±0.063	0.561±0.008	0.761±0.047
dsSVM	1,5	0.972±0.082	0.422±0.013	0.789±0.023
rbfSVM	1	**1.000±0.000**	0.000±0.000	0.667±0.000
uHOSVDgls	7,3,3	0.839±0.039	0.594±0.020	0.757±0.050
uHOOIgls	4,3,10	0.900±0.031	0.500±0.012	0.767±0.035
uHONMFgls	3,5,3	0.897±0.079	0.306±0.034	0.700±0.052
uHOSVDtf	4,2,3	0.764±0.053	0.550±0.041	0.693±0.046
uHOOItf	3,7,3	0.811±0.048	0.661±0.055	0.761±0.050
uHONMFmu	3,5,3	0.789±0.029	**0.867±0.154**	**0.815±0.040**
sHOSVD	4,3,8	0.831±0.054	0.633±0.012	0.765±0.044
sHOOI	3,4,4	0.761±0.045	0.667±0.000	0.730±0.039
sHONMF	3,4,6	**0.958±0.110**	0.006±0.069	0.641±0.075

As shown in Table IV, uHONMFmu obtains the highest mean prediction accuracy (0.8148). This is better than GenHMMs and DiscHMMs without and with gene selection (0.7593 and 0.7611, respectively), and dsSVM (0.789). uHONMFmu also outperforms uHOSVDgls, uHOOIgls, and uHOOItf. The reasons why uHONMFls and uHOOItf do not performed well needs further investigation. The supervised sHOSVD, sHOOI and sHONMF did not achieve good results. The small parameters of the tensor methods indicate that only few genes pathways and biological stages respond to the INFβ treatment. The MLDR techniques are able to dramatically reduce the dimension of the original dataset and transform the sample matrices into new "equivalent" short vectors which are used for classification. In uHONMFmu, a 53 by 7 test sample can be represented by a vector of size 1 by 3 in feature space; thus reducing the data by 99.19% while preserving discriminative information. The execution times (in seconds) were recorded for each method. Tab.V shows the results. The tensor decomposition based approaches use the same parameter $(3, 5, 3)$. The number of selected genes is 7. It can be seen that HONMF based approaches are faster than the HMMs based methods while giving at least comparable classification results, though slower than HOSVD and HOOI methods.

TABLE V
RUNNING TIME ON COMPLETE INF-BETA DATA

Methods	DiscHMMs	uHOSVDgls	uHOOIgls	uHONMFtf
Time (s.)	2.117×10^3	1.321	1.057	1.662×10^3

VIII. CONCLUSION

Non-negative information can help in analysis of microarray gene expression data. This paper investigated the performance of NMF based classification scheme for binary and multi-class microarray datasets, and extended it for GST data, which is our main contribution. Methods devised specifically for the analysis of GST data will be very useful in the near future, as many recent clinical data are given in the form of tensor data of order 3 or more. In this regards, we have implemented a HONMF-based scheme for classifying sample GST data from INFβ data. We have shown that our approach are faster and still comparable in classification performances to two recent methods developed for analyzing the same dataset. More research need to be done, however, to improve the classification performances of the tensor-based methods (for instance, SVM classifiers will be investigated deeply), and in particular to devise methods that can deal with missing values. We also plan to investigate gene selection methods such as gene-pairs or gene-triplets search algorithms for bio-marker selection. Beside classification, bi-clustering and tri-clustering approaches for GST data will be studied for determining pattern of genes or samples given certain doses (in dose-response GST data) or time intervals (in drug-response GST data).

APPENDIX A
OTHER TOPICS

An important concern when analyzing microarray data is how to handle missing values. There are three choices to deal with this problem. The safest way is to remove vectors (for 2D microarray data) or matrices (for 3D microarray data) with missing values. But this remove the other useful information as well. If missing values appear randomly, we can estimate/impute them using statistical and machine learning methods. We extended the KNN and SVD based missing value imputation methods (KNNimpute and SVDimpute, respectively) for 2D microarray data into methods (3KNNimpute and 3SVDimpute, respectively) for 3D microarray data [35]. If we encounter systemical missing values, we need to avoid touching missing values, while using the other existing values. In NMF and tensor decomposition, weighted least squares based methods are proposed in literature to handle missing values [32] [33].

For clustering microarray gene time-series data, we proposed multiple alignment based clustering methods, such as spectral clustering, to find gene patterns [39] [38]. Integral distance is used as dissimilarity measure between gene expression profiles. We also presented new index to evaluate the number of clusters. Experiments results show that our proposed clustering methods outperform the other methods for mining gene time-series data.

ACKNOWLEDGMENT

This research has been partially funded by IEEE CIS Walter Karplus Summer Research Grant 2010, and Canadian NSERC Grants #RGPIN228117-2006.

REFERENCES

[1] S.E. Baranzini, P. Mousavi, J. Rio, S.J. Caillier, A. Stillman, and P. Villoslada, "Transcription-based prediction of response to INFβ using supervised computational methods," *PLOS Biology*, vol. 3, no. 1, pp. e2, 2005.

[2] B. Weinstock-Guttman, D. Badgett, K. Patrick, L. Hartrich, R. Santos, D. Hall, M. Baier, J. Feichter, and M. Ramanathan, "Genomic effects of IFN-β in multiple sclerosis patients" *J Immunol.*, vol. 171 no. 5, pp. 2694-2702, 2003.

[3] A. Zhang, "Advanced Analysis of Gene Expression Microarray Data," World Scientific, Singapore, 2009.

[4] K.M. Borgwardt, S.V.N., Vishwanathan, and H.P. Kriegel, "Class prediction from time series gene expression profiles using dynamical systems kernels," *Proc. Pacific Symposim on Biocomputing(PSB)*, HI, 2006, pp. 547-558.

[5] D.D. Lee and S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Science*, vol. 401, pp. 788–791, 1999.

[6] D.D. Lee, S. Seung, "Algorithms for Non-negative Matrix Factorization," in *NIPS*, Breckenridge, 2000, pp.556-562.

[7] A. Cichocki, *et al*., *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*, Wiley, West Sussex, 2009.

[8] T.G. Kolda and B.W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455-500, 2009.

[9] L.D. Lathauwer, B.D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253-1278, 2000.

[10] M.A.O. Vasilescu and D. Terzopoulos. "Mulitlinear independent component analysis," *Proc. CVPR*, San Diego, 2005, vol. 1, pp. 547–553.

[11] M. Mørup, L.K. Hansen, and S.M. Arnfred, "Algorithms for sparse nonnegative tucker decompositions," *Neural Computation*, vol. 20, no. 8, pp. 2112-2131, 2008.

[12] O. Alter, P.O. Brown, and D. Botstein, "Singular value decomposition for genome-wide expression data processing and modeling," *PNAS*, vol. 97, no. 18, pp. 10101-10106, 2000.

[13] D. Huang, C. Zheng, "Independent component analysis-based penalized discriminant method for tumor classification using gene expression Data," *Bioinformatics*, vol. 22, 1855-1862, 2006.

[14] J.P. Brunet, *et al*., "Metagenes and molecular pattern discovery using matrix factorization," *PNAS*, vol. 101, no. 12, pp. 4164-4169, 2004.

[15] H. Kim and H. Park, "Sparse non-negatice matrix factorization via alternating non-negative-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495-1502, 2007.

[16] R. Schachtner, *et al*., "Exploring matrix factorization techniques for classification of gene expression profiles," in *WISP*, Alcala de Henares, 2007, pp. 1-6.

[17] W. Liu, K. Yuan, and Z. Xie, "Sparse nonnegative matrix factorization for classification of gene expression data," in Proc. BIBE-ICBBE, Wuhan, 2007, pp. 180-183.

[18] P. Zhang, *et al*., "Tumor classification using non-negative matrix factorization," *ICIC 2008, CCIS 15*, pp. 236-243, 2008.

[19] C. Zheng, *et al*., "Gene expression data classification based on non-negative matrix factorization," in Proc. IJCNN, Atlanta, 2009, pp. 3542-3547.

[20] T. Lin, N. Kaminski, and Z. Bar-Joseph, "Alignment and classification of time series gene expression in clinical studies," *Bioinformatics*, vol. 24 ISMB 2008, pp. i147-i155, 2008.

[21] I.G. Costa, A. Schönhuth, C. Hafemeister, and A. Schliep, "Constrained mixture estimation for analysis and robust classification of clinical time series," *Bioinformatics*, vol. 25 ISMB 2009, pp. i6-i14, 2009.

[22] M.A.O. Vasilescu and D. Terzopoulos, "Multilinear image analysis for facial recognition," *Proc. the International Conference on Pattern Recognition*, Quebec, 2002, vol. 3, pp. 511-514.

[23] L. Omberg, G.H. Golub, and O. Alter, "A tensor higher-order sigular value decomposition for integrative analysis of DNA microarray from different studies," *PNAS 2007*, vol. 104, no. 47, pp. 18371-18376, 2007.

[24] B. Savas and L. Eldén, "Handwritten digit classification using higher order singular value decomposition," *Pattern Recongtion*, vol. 40, pp. 993-1003, 2007.

[25] M. Du, S. Zhang, and H. Wang, "Tumor classification using higher-order gene expression profiles based on multilinear ICA," *Advances in Bioinformatics*, vol. 2009, doi:10.1115/2009/926450, 2009.

[26] M.H. van Benthem and M.R. Keenan, "Fast algorithm for the solution of large-scale non-negative-constrained least squares problems," *J. Chemometrics*, vol. 18, pp. 441-450, 2004.

[27] B. Savas, "Analyses and tests of handwritten digit recognition algorithms," Master thesis, Dept. Mathathmatics Scientific Computing, Linköping University, Linköping, Sweden, 2003.

[28] U. Alon, *et al*., "Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by oligonucleotide arrays," *PNAS*, vol. 96, no. 12, pp. 6745-6750, 1999.

[29] J. Khan, *et al*., "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine*, vol. 7, no. 6, pp. 673-679, 2001.

[30] C.A. Andersson and R. Bro, "The n-way toolbox for MATLAB," *Chemometr. Intell. Lab. Syst.*, vol. 52, pp. 1-4, 2000.

[31] C.A. Andersson and R. Bro, "Improving the speed of multi-way algorithms: Part I. Tucker3," *Chemometrics and Intelligent Laboratory Systems*, vol. 42, pp. 93-103, 1998.

[32] S. Zhang, W. Wang, J. Ford, and, F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," *Proc. the 6th SIAM Conference on Data Mining*, ML, 2006, 548-552.

[33] E. Acar, D. M. Dunlavy, T. G. Kolda and M. Mrup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, 2010, in press.

[34] Y. Li and A. Ngom, "Classification of gene-sample-time microarray expression data via tensor decomposition methods," *Bioinfomatics*, under review, submitted on Sep.06, 2010, manuscript ID: BIOINF-2010-1329.

[35] Y. Li, A. Ngom, "Classification of clinical gene-sample-time microarray expression data via tensor decomposition methods," *The 7th International Meeting on Computational Intelligence Methods for Bioinfomatics and Biostatistics*, Palermo, Sep., 2010, (CD publication, http://cibb10.pa.icar.cnr.it/technical-program).

[36] Y. Li and A. Ngom, "Tensor decomposition methods for the classification of three-dimensional microarray data," *Proc. the 2011 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, under review, submitted on Oct.29, 2010, manuscript ID: 1569376277.

[37] Y. Li and A. Ngom, "Non-negative matrix and tensor factorization based classification of clinical microarray gene expression data," *IEEE International Conference on Bioinformatics & Biomedicine*, Hong Kong, Dec. 2010, in press, paper ID: B343.

[38] Y. Li, N. Subhani, A. Ngom, and L. Rueda, "Alignment-based versus variation-based transformation methods for clustering microarray time-series data," *Proc. the 2010 ACM International Conference On Bioinformatics and Computational Biology*, Niagara Falls, NY, Aug. 2010, pp.53-61.

[39] N. Subhani, Y. Li, A. Ngom, and L. Rueda, "Alignment versus variation vector methods for clustering microarray time-series data," *Proc. the 2010 IEEE Congress on Evolutionary Computation*, Barcelona, Spain, Jul. 2010, pp. 818-825.

[40] Y. Li, A. Ngom, and L. Rueda, "Missing value imputation methods for gene-sample-time microarray data analysis", *Proc. the 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology* Montreal, Canada, May 2010, pp.183-189.

[41] Y. Li, L. Rueda, M. Herrera, and A. Ngom, "A new approach to feature selection using linear dimensionality reduction," *The 1st IEEE International Conference on Computational Advances in Bio and Medical Sciences*, Orlando, Feb., 2011, in preparation.

[42] Y. Li, L. Rueda, M. Herrera, and A. Ngom, "A new approach to feature selection using linear dimensionality reduction and genetic algorithm," *The 2011 IEEE Congress on Evolutionary Computation*, New Orleans, June, 2011, in preparation.