

Discovering biological processes from microarray data using independent component analysis

Extended Abstract

Su-In Lee¹ and Serafim Batzoglou^{2,3}

¹Department of Electrical Engineering, Stanford University, Stanford, CA94305-9010, USA.

²Department of Computer Science, Stanford University, Stanford, CA94305-9010, USA.

³To whom correspondence should be addressed.

EMAIL serafim@cs.stanford.edu; FAX 650-725-1449

ABSTRACT

We propose a hypothesis-free methodology for discovering genome-wide expression patterns specific to underlying biological processes from DNA microarray expression data. We apply linear and nonlinear independent component analysis (ICA) as a tool for decomposing microarray data into statistically independent components. Each component represents a gene expression pattern of a putative underlying biological process. Genes that exhibit significant upregulation or downregulation within each component are grouped into clusters and putative biological meaning is assigned to each component by using the annotations of genes within the component's clusters. We test the statistical significance of enrichment of specific gene annotations within each cluster. ICA-based clustering outperforms other leading methods in constructing functionally coherent clusters on a variety of datasets, supporting our model of expression data as a mixture of statistically independent biological processes. Comparison of the performance of several ICA algorithms, including a novel nonlinear method, shows that a natural-gradient maximum-likelihood method performs well in all datasets, while the nonlinear method performs best in the smaller datasets. We conclude that ICA is a versatile technique that can extract linear and nonlinear features in gene expression, and can group genes in non-mutually exclusive clusters with strong functional coherence.

INTRODUCTION

Microarray technology has enabled high-throughput genome-wide measurements of gene transcript levels, promising to provide insight into biological processes involved in gene regulation. To aid such discoveries, mathematical and computational tools are needed that are versatile enough to capture the underlying biology, and simple enough to be applied efficiently on large datasets. Analysis tools fall broadly in two categories: supervised and unsupervised approaches. When prior knowledge can group samples into different classes (e.g., normal versus cancer tissue), supervised approaches can be used for finding gene expression patterns (features) specific to each class, and for class prediction of new samples (Ando et al. 2002, Brown et al. 2000, Golub et al. 1999, Mukherjee et al. 1999, Atul 2002). Unsupervised (hypothesis-free) approaches are important for discovering novel biological mechanisms, for revealing genetic regulatory networks and for analyzing large datasets for which little prior knowledge is available. Here we apply linear and nonlinear independent component analysis (ICA) as a versatile unsupervised approach for microarray analysis, and evaluate its performance against other leading unsupervised methods.

Unsupervised analysis methods for microarray data can be divided into three categories: clustering approaches, model-based approaches, and projection methods. Clustering approaches group genes and experiments with similar behavior (Eisen et al. 1998, Tavazoie et al. 1999, Tamayo et al. 1999, Ben-Dor et al. 1999, Kim et al. 2001), making the data simpler to analyze (Kaminski

and Friedman 2002). Clustering methods group genes that behave similarly under similar experimental conditions, assuming that such are functionally related. Most clustering methods do not attempt to model the underlying biology. A disadvantage of such methods is that they partition genes and experiments into mutually exclusive clusters, whereas in reality a gene or an experiment may be part of several biological processes. Model-based approaches first generate a model that explains the interactions among biological entities participating in genetic regulatory networks, and then train the parameters of the model on expression datasets (Friedman et al. 2000; Bussermaker et al. 2001; Lazzeroni and Owen 2002; Segal et al. 2002, 2003). Depending on the complexity of the model, one challenge of model-based approaches is the lack of sufficient data to train the parameters, and another challenge is the prohibitive computational requirement of training algorithms.

Projection methods linearly decompose the dataset into components that have a desired property. There are largely two kinds of projection methods: *principle component analysis* (PCA) and *independent component analysis* (ICA). PCA projects the data into a new space spanned by the *principal components*. Each successive principal component is selected to be orthonormal to the previous ones, and to capture the maximum information that is not already present in the previous components. Applied to expression data, PCA finds principle components, the *eigenarrays*, which can be used to reduce the dimension of expression data for visualization, filtering of noise, and for simplifying the subsequent computational analyses (Alter et al. 2000, Misra et al. 2002).

In contrast to PCA, ICA decomposes an input dataset into components so that each component is statistically as *independent* from the others as possible. A common application of ICA is in blind source separation (BSS) problems (Juttan et al. 1991): suppose that there are M independent acoustic sources—such as speech, music, and others—that generate signals simultaneously, and N microphones around the sources. Each microphone records a mixture of the M independent signals. Given N mixed vectors that are the signals received from the microphones, where $N \geq M$, ICA retrieves M independent components that are close approximations of the original signals up to scaling. ICA has been used successfully in BSS of neurobiological signals such as electroencephalographic (EEG) and magnetoencephalographic (MEG) signals (Makeig et al. 1996, Vigarío et al. 1997, 1998), and for financial time series analysis (Back 1997, Kiviluoto et al. 1998). Most applications of ICA assume that the source signals are mixed *linearly* into the input signals, and algorithms for linear ICA have been developed extensively (Bell et al. 1995, Amari et al. 1998, Cardoso 1999a, Hyvarinen 1999b, Lee et al. 1999b & 2000). In several applications *nonlinear* mixtures may provide a more realistic model, and recently several methods have been developed for performing nonlinear ICA (Burel et al. 1992, Hyvarinen and Pajunen 1999, Harmeling et al. 2001, 2002). Liebermeister (2002) first proposed using linear ICA for microarray analysis to extract expression modes, where each mode represents a linear influence of a hidden cellular variable. However, there has been no systematic analysis of the applicability of ICA as an analysis tool in diverse datasets, or comparison of its performance with other analysis methods.

Here we apply linear and nonlinear ICA to microarray data analysis to project the samples into independent components. We cluster genes in an unsupervised fashion into non-mutually exclusive clusters, based on their load in each independent component. Each retrieved independent component is considered a putative biological process, which can be characterized by the functional annotations of genes that are predominant within the component. To perform nonlinear ICA, we introduce a methodology that combines the simplifying kernel trick (Muller et al. 2001) with a generalized mixing model. We systematically evaluate the clustering performance of several ICA methods on five expression datasets, and find that overall ICA is superior to other leading clustering methods that have been used to analyze the same datasets. Among the different ICA methods, the natural-gradient maximum-likelihood estimation (NMLE) method (Bell and Sejnowski 1995, Amari

et al. 1998) is best in the two largest datasets, while our nonlinear ICA method is best in the three smaller datasets.

RESULTS

Mathematical Model of Gene Regulation

We model the transcription level of all genes in a cell as a mixture of independent biological processes. Each process forms a vector representing levels of gene upregulation or downregulation; at each condition, the processes mix with different activation levels to determine the vector of observed gene expression levels measured by a microarray sample.

Mathematically, suppose that a cell is governed by M independent biological processes $S = (s_1, \dots, s_M)^T$, each of which is a vector of K gene levels, and that we measure the levels of expression of all genes in N conditions, resulting in a microarray expression matrix $X = (x_1, \dots, x_N)^T$. We define a model whereby the expression level at each different condition j can be expressed as linear combinations of the M biological processes: $x_j = a_{j1}s_1 + \dots + a_{jM}s_M$. We can express this model concisely in matrix notation (Equation 1).

$$X = AS, \quad \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1M} \\ \vdots & & \vdots \\ a_{N1} & \cdots & a_{NM} \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_M \end{bmatrix} \quad (1)$$

When the matrix X represents log ratios $x_{ij} = \log_2(R_{ij}/G_{ij})$ of red (experiment) and green (reference) intensities, Equation 1 corresponds to a multiplicative model of interactions between biological processes. More generally, we can express $X = (x_1, \dots, x_N)^T$ as a post-nonlinear mixture of the underlying independent processes (Equation 2, where $f(\cdot)$ is a nonlinear mapping from N to N dimensional space).

$$X = f(AS), \quad \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = f \left(\begin{bmatrix} a_{11} & \cdots & a_{1M} \\ \vdots & & \vdots \\ a_{N1} & \cdots & a_{NM} \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_M \end{bmatrix} \right) \quad (2)$$

Since we assume that the underlying biological processes are independent, we can view each of the vectors s_1, \dots, s_M as a set of K samples of an independent random source. Then, ICA can be applied to find a matrix W that provides the transformation $Y = (y_1, \dots, y_M)^T = WX$ of the observed matrix X under which the transformed random variables y_1, \dots, y_M , called the independent components, are as independent as possible (Hyvarinen 1999). Assuming certain mathematical conditions are satisfied (see Discussion), the retrieved components y_1, \dots, y_M are close approximations of s_1, \dots, s_M up to permutation and scaling.

Methodology

Given a matrix X of N microarray measurements, we perform the following steps:

Step 1. ICA-based decomposition. Use ICA to express X according to Equation 1 or 2, as a mixture of independent components y_1, \dots, y_M . Each component is a vector of K gene expression levels, or loads $y_i = (y_{i1}, \dots, y_{iK})$.

Step 2. Clustering. Cluster the genes according to their relative loads $y_{ij}/\text{mean}(y_i)$ in the components y_1, \dots, y_M . A gene may belong to more than one cluster.

Step 3. Measurement of significance. Measure the enrichment of each cluster with genes of known functional annotations.

ICA-based decomposition. Prior to applying ICA, we normalize the expression matrices X to contain log ratios $x_{ij} = \log_2(R_{ij}/G_{ij})$ of red and green intensities, and we remove any samples that

are closely approximated as linear combinations of other samples. We find as many independent components as samples in the input dataset, i.e., $M = N$ (see Discussion). The algorithms we use for ICA are described in Methods.

Clustering. Based on our model, each component is a putative genomic expression program of an independent biological process. Our hypothesis is that genes showing relatively high or low expression level within the component are the most important for the process. We create two clusters for each component: one cluster containing genes with expression level higher than a threshold, and one cluster containing genes with expression level lower than a threshold.

$$\begin{aligned} \text{Cluster}_{i,1} &= \{ \text{gene } j \mid y_{ij} > \text{mean}(y_i) + c \times \text{std}(y_i) \} \\ \text{Cluster}_{i,2} &= \{ \text{gene } j \mid y_{ij} < \text{mean}(y_i) - c \times \text{std}(y_i) \} \end{aligned} \quad (3)$$

In Equation 3, y_i is the i^{th} independent component, a vector of length K ; $\text{mean}(y_i)$ is the average, $\text{std}(y_i)$ is the standard deviation of y_i ; and c is an adjustable coefficient.

Measurement of biological significance. For each cluster, we measure the enrichment with genes of known functional annotations.

In our datasets we measured the biological significance of each cluster as follows. For Datasets 1–4, we used the GO (Ashburner et al. 2001) and KEGG (Kanehisa et al. 2002) annotation databases. We combined all annotations in 502 gene categories for yeast, and 996 categories for *C. elegans* (see Methods). For Dataset 5, we used the seven categories of tissues annotated by Hsiao et al. (2001). We matched each ICA cluster with every category and calculated the p -value, that is, the chance probability of the observed intersection between the cluster and the category (details in Methods). We ignored categories with p -values greater than 10^{-3} .

Evaluation of performance

We applied ICA on the following five expression datasets (see our website): (1) budding yeast during cell cycle and CLB2/CLN3 overactive strain (Spellman et al. 1998), consisting of spotted array measurements of 4579 in 22 experimental conditions; (2) budding yeast during cell cycle (Cho et al. 1998) consisting of Affymetrix oligonucleotide array measurements of 6616 genes in synchronized cell cultures at 17 time points; (3) yeast in various stressful conditions (Gasch et al. 2000) consisting of spotted array measurements of 6152 genes in 173 experimental conditions that include temperature shocks, hyper- and hypoosmotic shocks, exposure to various agents such as peroxide, menadione, diamide, dithiothreitol, amino acid starvation, nitrogen source depletion, and progression into stationary phase; (4) *C. elegans* in various conditions (Kim et al. 2001) consisting of spotted array measurements of 11917 genes in 179 experimental conditions and 17817 genes in 374 experimental conditions that include growth conditions, developmental stages, and a variety of mutants; and (5) normal human tissue (Hsiao et al. 2001) consisting of Affymetrix oligonucleotide array measurements of 7070 genes in 59 samples of 19 kinds of tissues. We used KNNimpute (Troyanskaya et al. 2001) to fill-in missing values. For each dataset, first we decomposed the expression matrix into independent components using ICA, and then we performed clustering of genes based on the decomposition.

We evaluated the performance of ICA in finding components that result in gene clusters with biologically coherent annotations, and compared against the performance of other methods that were used to analyze the same datasets. In particular, we compared with the following methods: PCA, which Alter et al. (2000) applied to the analysis of the yeast cell cycle data (Dataset 1), and Misra et al. (2002) applied to the analysis of human tissue data (Dataset 5); k-means clustering, which Tavazoie et al. (1999) applied to the yeast cell cycle data (Dataset 2); the Bayesian approach that Segal et al. (2002, 2003) applied to the dataset of yeast cells under stressful conditions (Dataset

3); the Plaid model (Lazzeroni and Owen 2002) applied on the same data (Dataset 3); and the topographical map-based method (topomap) that Kim et al. (2001) applied to the *C. elegans* data (Dataset 4). In all comparisons we applied the natural-gradient maximum-likelihood estimation (NMLE) ICA algorithm (Bell and Sejnowski 1995, Amari et al. 1998) for linear ICA, and a kernel-based nonlinear BSS algorithm (Harmeling et al. 2001, 2002) for nonlinear ICA. The single parameter in our method was the coefficient c in Equation 3, with a default $c = 1.25$.

Figure 1 summarizes the results of comparing the performance of NMLE-based clustering with the other clustering approaches. Every point in this figure is a functional category (GO or KEGG) for which there is a cluster with a p -value $< 10^{-3}$ for both NMLE and the other compared approach. The y -axis is the smallest p -value for that functional category in an NMLE cluster, and the x -axis is the smallest p -value in a cluster of the other approach. As seen from the figure, within the functional categories for which both approaches had significant clusters, NMLE performed better than PCA, k -means clustering, the Bayesian approach, and the Plaid model. NMLE performed similarly with the topomap approach. For the rest of the functional categories, overall both approaches showed low statistical significance (details in our website). For dataset 5, we generated 112 clusters and measured the enrichment of each of the seven tissue-specific categories annotated by Hsiao et al. (2001) within each cluster. The three most significant independent components were enriched for liver-specific, muscle-specific, and vulva-specific genes with p -values of 10^{-133} , 10^{-124} and 10^{-117} , respectively. The 4th most significant cluster was brain-specific (p -value = 10^{-93}). These p -values are much lower than the ones found by Misra et al. (2002) for three tissue-specific clusters (muscle, liver, and brain); however, a direct comparison was not possible because they applied their PCA analysis on an older version of the microarray dataset, which contains 40 samples compared to 59 samples in the new version.

We also applied nonlinear ICA (described in Methods) to the above datasets. Nonlinear ICA performed significantly better in the smaller datasets (1, 2, and 5), but worse in the two larger datasets (3, 4). Detailed results and gene lists for all the clusters that we obtained with our methods are provided in the web supplements in our webpage: <http://www.stanford.edu/~silee/ICA>.

Comparison of Different Linear and Nonlinear ICA Algorithms

We tested six linear ICA methods: (1) Natural Gradient Maximum Likelihood Estimation (NMLE) (Amari et al. 1998, Bell and Sejnowski 1995), (2) Joint Approximate Diagonalization of Eigenmatrices (JADE) (Cardoso 1999a), (3-5) Fast Fixed Point (Fast) ICA with three different measures of non-Gaussianity (Hyvarinen 1999b), and (6) Extended Information Maximization (Infomax) (Lee et al. 1999a, 2000). We also tested 2 variations of nonlinear ICA: (1) Gaussian radial basis function (RBF) kernel, and (2) polynomial kernel. For each dataset, we compared the biological coherence of clusters generated by each method. Among the six linear ICA algorithms, NMLE was the best in all datasets. Among both linear and nonlinear methods, the Gaussian kernel nonlinear ICA method was the best in Datasets 1 and 2, the polynomial kernel nonlinear ICA method was best in Dataset 5, and NMLE was best in the large datasets (3 and 4). In our website, we compare the NMLE method with three other ICA methods. We show the remaining comparisons in our web supplement. Overall, the NMLE algorithm consistently performed well in all datasets. The nonlinear ICA algorithms performed best in the small datasets, but were unstable in the two largest datasets.

The Infomax ICA algorithm can automatically determine whether the distribution of each source signal is super-Gaussian, with sharp peak at the mean and long tails (such as the Laplace distribution), or sub-Gaussian, with small peak at the mean and short tails (such as the uniform distribution). Interestingly, the application of Infomax ICA to all the expression datasets uncovered no source signal with sub-Gaussian distribution. A likely explanation is that the microarray

expression datasets are mixtures of super-Gaussian sources rather than of sub-Gaussian sources. This finding is consistent with the following intuition: underlying biological processes are super-Gaussian, because they affect sharply the relevant genes, typically a fraction of all genes (long tails in the distribution), and leave the majority of genes relatively unaffected (sharp peak at the mean of the distribution).

DISCUSSION

ICA is a powerful statistical method for separating mixed independent signals. We proposed applying ICA to decompose microarray data into independent gene expression patterns of underlying biological processes, and to group genes into clusters that are mutually non-exclusive with statistically significant functional coherence. Our clustering method outperformed several leading methods on a variety of datasets, with the added advantage that it requires setting only one parameter, namely the fraction c of standard deviations beyond which a gene is considered to be associated with a component's cluster. We observed that performance was not very sensitive to that parameter, suggesting that ICA is robust enough to be used for clustering with little human intervention. The empirical performance of ICA in our tests supports the hypothesis that statistical independence is a good criterion for separating mixed biological signals in microarray data.

Linear ICA models a microarray expression matrix X as a linear mixture $X = AS$ of independent sources. ICA decomposition attempts to find a matrix W such that $Y = WX = WAS$ recovers the sources S (up to scaling and permutation of the components). The three main mathematical conditions for a solution to exist are (Hyvarinen 1999a): (1) the number of observed mixed signals is larger than, or equal to the number of independent sources, i.e., $N \geq M$ in Equation 1; (2) the columns of the mixing matrix A are linearly independent; and (3) there is at most one source signal with Gaussian distribution. In microarray analysis, condition 1 may mean that when too few separate microarray experiments are conducted, some of the important biological processes of the studied system may collapse into a single independent component. If the number of sources is known to be smaller than the number of observed signals, PCA is usually applied prior to ICA, to reduce the dimension of the input space. Because we expect that the true number of concurrent biological processes inside a cell is very large, in our tests we attempted to find the maximum number of independent components, which is equal to the rank of X . Condition 2 is easily satisfied by removing microarray experiments that can be expressed as linear combinations of other experiments, i.e., those that make the matrix X singular. Condition 3 is reasonable for analyzing biological data: the most typical Gaussian source is random noise, whereas biological processes that control gene expression are expected to be highly non-Gaussian, sharply affecting a set of relevant genes, and leaving most other genes relatively unaffected. Moreover, the ability of ICA to separate a single Gaussian component may prove ideal in separating the experimental noise from expression data. This is a topic for future research.

ICA is a projection method for data analysis, but it can be interpreted also as a model-based method, where the underlying model explains the gene levels at each condition as mixtures of several statistically independent biological processes that control gene expression. Moreover, ICA naturally leads to clustering, with each gene assigned to the clusters that correspond to independent components where the gene has a significantly high expression level. An advantage of ICA-based clustering is that each gene can be placed in zero, one, or several clusters.

ICA is very similar to PCA, as both methods project a data matrix into components in a different space. However, the goals of the two methods are different. PCA finds the uncorrelated components of maximum variance, and is ideal for compressing data into a lower-dimensional space by removing the least significant components. ICA finds the statistically independent components, and is ideal for separating mixed signals. It is generally understood that ICA recovers

more interesting (i.e., non random) signals than PCA does (Back et al. 1997). If the input comprises a mixture of signals generated by independent sources, independent components are close approximates of the individual source signals. Otherwise, ICA is the projection-pursuit technique that finds the projection of the high-dimensional dataset exhibiting the most interesting (i.e., non-Gaussian) behavior (Hyvarinen et al. 1999a). Thus, ICA can be trusted to find statistically interesting features in the data, which may reflect underlying biological processes.

We presented a new method for performing nonlinear ICA, based on the kernel trick (Muller et al. 2001) that is usually applied in Support Vector Machine (SVM) learning (Scholkopf et al. 1999). Our method can deal with more general nonlinear mixture models than existing methods, and reduces the computation load so as to be applicable to larger datasets. Using nonlinear ICA we were able to improve performance in the three smaller datasets. However, the algorithm was still unstable in the two larger datasets. Using a Gaussian kernel, the method performed very poorly in these datasets; using a polynomial kernel, it performed comparably to linear ICA. Overall we demonstrated that nonlinear ICA is a promising method that, if applied properly, can outperform linear ICA on microarray data.

The linear mixture model that we proposed has the advantage of simplicity—it is expected to perform well in finding first-order features in the data, such as when a single transcription factor upregulates a given subset of genes. Nonlinear ICA may prove capable of capturing multi-gene interactions, such as when the cooperation of several genes, or the combination of presence of some genes and absence of others, is necessary for driving the expression of another set of genes. In future research, we will attempt to capture such interactions with nonlinear modeling, and to deduce such models from the components that we obtain with nonlinear ICA. Currently our ICA model does not take into account time in experiments such as the yeast cell cycle data. A direction for future research is to incorporate a time model in our approach, whenever the microarray measurements represent successive time points.

Finally, a direction for future research is to use ICA as a preprocessing step, followed with subsequent analyses, such as clustering or classification methods, on the transformed space. Sophisticated clustering methods may produce more coherent groups of genes than our simple clustering scheme that genes with high coefficients in each component separately. Here we demonstrated that ICA transforms the data into a space whose axes have significant functional coherence, potentially making further analyses considerably more effective than when applied to the original microarray data.

METHODS

Data Treatment

The five datasets we used in this analysis were preprocessed to contain log-ratios $x_{ij} = \log_2(R_{ij}/G_{ij})$ between red and green intensities. In Dataset 2, we variance-normalized the data as in described by Tavazoie et al. (1999). When necessary, experiments that made the expression matrix close to singular were removed. Details are shown in our webpage.

Gene Annotation Database

For the yeast and *C. elegans* datasets (Datasets 1, 2, 3, and 4), we used the functional categories defined by four different annotation systems: three tree ontologies developed by the Gene Ontology (GO) Consortium (Ashburner et al. 2001) and one from Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa et al. 2002). For budding yeast, we used 502 functional categories from GO and KEGG annotations: 243 functional categories from biological process (GO), 120 from molecular function (GO), 81 from cellular component (GO) and 58 from KEGG

biological pathway annotation. For *C. elegans*, we used 974 functional categories: 194 categories from biological process, 458 from molecular function, 231 from cellular component, and 91 from KEGG annotations. For the human dataset (Dataset 5), we used the functional annotations compiled by Hsiao et al. (2001): brain (618), kidney (91), liver (279), lung (75), muscle (317), prostate (46), and vulva (103). For each cluster, we reported functional categories with p -values smaller than 10^{-3} .

Calculating statistical significance

We measured the likelihood that a functional category and a cluster share the given number of genes by chance, based on the following quantities: the number of genes that are shared by the functional category and the cluster (k), the number of genes within the cluster that are in any functional category (n), the number of genes within the functional category that appear in the microarray dataset (f) and the total number of genes that appear both in the microarray dataset and in any functional category (g). Based on the hypergeometric distribution, the probability p that at least k genes are shared by the functional category and the cluster is given by Equation 4.

$$p = 1 - \sum_{i=0}^{k-1} \frac{\binom{f}{i} \binom{g-f}{n-i}}{\binom{g}{n}} \quad (4)$$

Algorithms for ICA

The ICA problem can be formulated as

$$Y = WX \quad (5)$$

where X represents an original expression dataset of N experiments \times K genes. The goal of ICA is to find W so that components, i.e., rows of Y , are statistically as independent as possible. To perform ICA, we used an ICA algorithm driven by maximum likelihood estimation (MLE), a statistical approach for finding estimations of unknown parameters that result in the highest probability for observations (Hyvarinen 2001). Applying a well-known principle of density of linear transformation to Equation 5 leads to

$$p_x(x) = |\det W| p_y(y) = |\det W| \prod_i p_i(w_i^T x) \quad (6)$$

where p_x and p_y are probability density functions of the original dataset X and the components Y , respectively and p_i is the probability density function of the i^{th} row of Y . The second equality is based on the assumption that the rows of Y are independent and so p_y may be factored. The log likelihood $L(W)$ of Equation 6 is given in Equation 7.

$$L(W) = \log[p_x(x)] = \log |\det W| + \sum_{i=1}^N \log p_i(w_i^T x) \quad (7)$$

Based on the gradient descent rule, a learning algorithm for finding the matrix W that maximizes the log-likelihood $L(W)$ is defined as follows.

$$\Delta W \propto \frac{\partial L(W)}{\partial W} = [W^T]^{-1} - g(Wx)x^T$$

$$\text{where, } g(y) = -\frac{\partial p(y)}{p(y)} = \left[-\frac{\partial p(y_1)}{p(y_1)}, \dots, -\frac{\partial p(y_N)}{p(y_N)} \right]^T \quad (8)$$

The above learning rule was first derived by Bell et al. (1995) from another approach called Information Maximization (*Infomax*) approach and can be derived from negentropy maximization (Girolami et al. 1997). Amari et al. (1998) proposed that the natural gradient method makes the

above learning rule more efficient and modified the above learning rule.

$$\Delta W \propto [(W^T)^{-1} - g(Wx)x^T]W^T W = [I - \{g(y)y^T\}]W \quad (9)$$

In the above learning rule, there is one unknown parameter, $g(y)$, a function of the probability density of the sources. In practice, it is enough to decide whether the distribution of the sources is super-Gaussian, or sub-Gaussian (Hyvarinen et al. 1999). Super-Gaussian distributions have a high peak at the mean and long tails (e.g., the Laplace distribution). Sub-gaussian distributions have a low peak at the mean and short tails (e.g., the uniform distribution). In our application, since independent components are expected to be genomic expression levels of biological processes, a super-Gaussian distribution is appropriate: a biological process is likely to affect a few relevant genes strongly (long tails), and the rest weakly (high peak at the mean, usually = 0). We choose $g(y)$ to be a sigmoid function, $g(u) = -2 \tanh(u)$, when we compared various ICA algorithms.

Nonlinear ICA model

Most of the studies on nonlinear blind source separation (BSS) have focused on simplified nonlinear mixtures, called *post-nonlinear* mixture models, described in Equation 10 (Hyvarinen 2001, Harmeling et al. 2002).

$$x_i = f_i\left(\sum_{j=1}^M a_{ij}s_j\right), \quad 1 \leq i \leq N \quad (10)$$

where s_i is the i^{th} source signal, x_i is the i^{th} mixture signal and $Y=f_i(Z)$ is a nonlinear function that operates component-wise, i.e., maps the i^{th} row of Z to the i^{th} row of Y . For convenience, in Equation 10 we assume that we have the same number of mixtures as that of source signals, $N = M$.

In general, an approach for BSS of post-nonlinear mixtures consists of two steps (Hyvarinen 2001, Taleb and Jutten 1999): a nonlinear stage and a linear stage. In a nonlinear stage, a rough determination of the inverse of the nonlinear function f (Equation 10) is made. A linear stage decomposes the data, obtained from the previous stage and expected to be linear mixtures, into statistically independent components. Usually, a linear ICA algorithm is used for the linear stage. Harmeling et al. (2002) proposed a kernel-based BSS algorithm, called kernel TDSEP (kTDSEP), that deals with more challenging nonlinear mixtures modeled as

$$x=f(AS) \quad (11)$$

where, $f(.)$ is a nonlinear function that maps N -dimensional vectors to N -dimensional vectors. We adopted the nonlinear stage of the approach proposed by Harmeling et al. (2002) using a *kernel trick* and used the NMLE approach for the linear stage. Our approach for nonlinear ICA is as follows.

Step 1. Construct a feature space, and map the input data to the feature space. We use the kernel trick (as described in Harmeling et al., 2002), with Gaussian RBF kernels and polynomial kernels.

Using the kernel trick, we map the expression data X of N experiments \times K genes in the N -dimensional input space into F in the L -dimensional ($L>N$) feature space.

Step 2. Apply linear ICA. We decompose the mapped data F in the feature space into statistically independent components using the NMLE algorithm.

The requirements for a valid kernel function that specifies the feature space are described by Muller et al. (2001). We choose a Gaussian radial basis function (RBF) kernel described as $k(x,y)=exp(-|x-y|^2)$ and a polynomial kernel of degree 2 described as $k(x,y)=(x^T y+1)^2$.

Determination of clustering coefficient

The only adjustable parameter in our approach is the clustering coefficient c in Equation 3.

When generating clusters, we varied the value from 0.25 to 2.0, and the result for $c=1.25$ was reported. The best settings of c for each individual dataset were: 0.5 for Dataset 1, 1.75 for Dataset 2, 1.25 for Dataset 3, 1.25 for Dataset 4 and 2 for Dataset 5. The favorable comparison of our approach to other methods was not sensitive to the value of c in this range. When comparing with principle components found by Alter et al. (2000), we generated clusters from components with the value of c that minimizes the smallest p -value from ICA or PCA clusters. The value $c=0.5$ was best for both PCA and ICA; we used that value for generating both sets of clusters.

ACKNOWLEDGEMENTS

We thank Relly Brandman, Chuong Do, Te-won Lee, and Yueyi Liu for helpful edits to the manuscript.

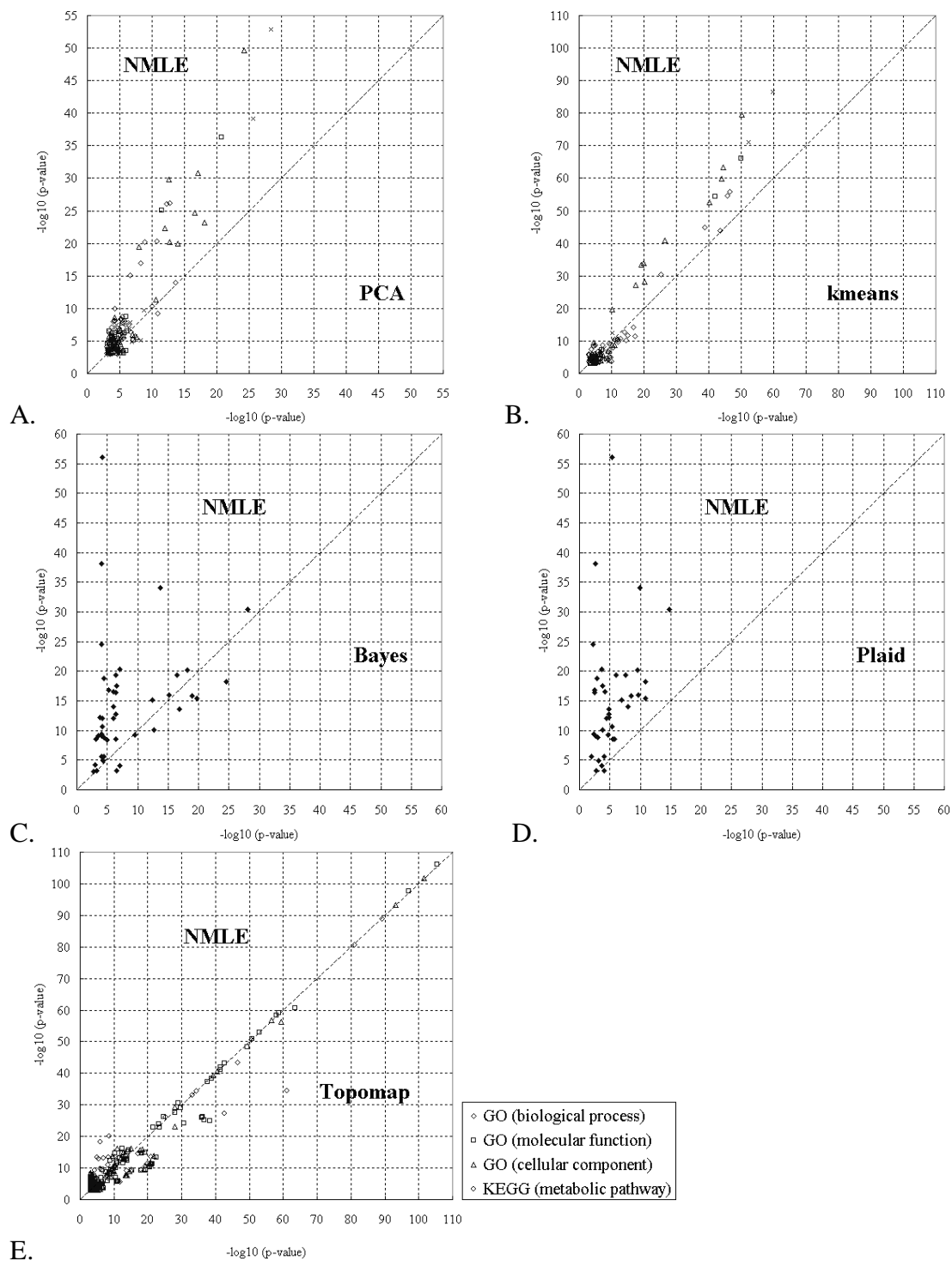
REFERENCES

- Alter O, Brown PO, Botstein D. *Proc. Natl. Acad. Sci. USA* 97:18, 2000.
- Amari S. *Neural Computation* 10:251–276, 1998.
- Ando T, Suguro M, Hanai T, Kobayashi T, Honda H, Seto M. *Jpn. J. Cancer Res.* 93(11):1207–12, 2002.
- Ashburner et al. (The Gene Ontology Consortium). *Genome Research* 11:1425–1433, 2001.
- Atul B. *Nature Drug Discovery* 1(12):951–960, 2002.
- ~~Back AD.~~ *International Journal of Neural Systems* 8:4, 1997.
- Bell AJ, Sejnowski TJ. *Neural Computation* 7:1129–1159, 1995.
- Ben-Dor A, Shamir R, Yakhini Z. *Journal of Computational Biology* 6:281–297, 1999.
- Brown M, Grundy WN, Lin D, Cristianini N, Sugnet CW, Furey TS, Ares M, Haussler D. *Proc. Natl. Acad. Sci. USA* 97:1, 2000.
- Bussermaker HJ, Li H, Siggia ED. *Nature Genetics* 27(2):167–174, 2001.
- Cardoso JF. *Neural Computation* 11(1):157–192, 1999a.
- Cho RJ, Campbell MJ, Winzeler EA, Steinmetz L, Conway A, Wodicka L, Wolfsberg TG, Gabrielian AE, Landsman D, Lockhart DJ, Davis RW. *Molecular Cell* 2:65–73, 1998.
- Eisen MB, Spellman PT, Brown PO, Botstein D. *Proc. Natl. Acad. Sci. USA* 95:14863–14868, 1998.
- Friedman N, Linial M, Nachman I, Pe’er D. In *Proc. the Fourth Annual International Conference on Computational Molecular Biology on RECOMB 2000*, pp. 127–135, 2000.
- Gasch AP, Spellman PT, Kao CM, Carmel-Harel O, Eisen MB, Storz G, Botstein D, Brown PO. *Mol. Biol. Cell* 9:4241–4257, 2000.
- Girolami M, Fyfe C. In *Proc. ICNN*, pp. 1788–1791. Houston, TX, 1997.
- Golub T, Slonim D, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD and Lander ES. *Science* 286:531–537, 1999.
- ~~Hardie DG. *Biochem. Soc. Sym.* 64:13–27, 1999.~~
- Harmeling S, Zieche A, Kawanabe M, Muller K. In Dietterich TG, Becker S, Ghahramani Z (eds.) *Advances in Neural Information Processing Systems 14*, MIT Press, Cambridge, MA, 2002.
- Hsiao L, Dangond F, Yoshida T, Hong R, Jensen R.V., Misra J, Dilon W, Lee K, Clark K, Harverty P, et al. *Physiol. Genomics* 7:97–104, 2001.
- Hyvarinen A. *IEEE Transactions on Neural Network* 10(3):626–634, 1999b.
- Hyvärinen A, Pajunen P. *Neural Networks* 12(3):429–439, 1999.
- Hyvarinen A, Karhunen J, Oja E. *Independent Component Analysis*, John Wiley & Sons, New York, 2001.
- ~~Hyvärinen A, Pajunen P. *Neural Networks* 12(3):429–439, 1999.~~
- ~~Jolliffe IT. *Principle Component Analysis*, Springer-Verlag, 1986.~~
- Juttan C, Herault J. *Signal Processing* 24:1–10. 1991.

- Kaminski N, Friedman, N. *American Journal of Reproductive and Cell Molecular Biology* 27:125–132, 2002.
- Kanehisa M, Goto S. In *Current Topics in Computational Molecular Biology*, pp. 301–315. MIT-Press, Cambridge, MA, 2002.
- Kim SK., Lund K, Kiraly M, Duke K, Jiang M., Stuart, J.M., Eizinger, A., Wylie, B.N. and Davidson, G.S. *Science* 293:2087–2092, 2001.
- Lazzeroni L, Owen A. *Statistica Sinica* 12(1):61-86, 2002.
- Lee TW, Lewicki MS, Sejnowski TJ. *IEEE Transactions on Pattern Recognition and Machine Intelligence* 22(10):1–12, 2000.
- Liebermeister W. *Bioinformatics* 18(1):51–60, 2002.
- Makeig S, Bell AJ, Jung TP, Sejnowski TJ. In *Advances in Neural Information Processing Systems* 8:145–151, MIT Press, Cambridge, MA, 1996.
- Misra J, Schmitt W, Hwang D, Hsiao L, Gullans S, Stephanopoulos G., Stephanopoulos, G. *Genome Research* 12:1112–1120, 2002.
- Mukherjee S, Tamayo P, Mesirov JP, Slonim D, Verri A, Poggio T, CBCL Paper #182/AI Memo #1677, MIT, Cambridge, MA, December, 1999.
- Muller KR, Mika S, Ratsch G, Tsudat K, Scholkopf B. *IEEE Transaction on Neural Networks* 12(2):181–201, 2001.
- ~~Quackenbush J. *Nature Genetics* 32:496–501, 2002.~~
- Scholkopf B, Burges CJC, Smola AJ. *Advances in kernel methods—Support Vector Learning*, MIT press, Cambridge, MA, 1999.
- Segal E, Barash, Y., Simon, I., Friedman, N., and Koller, D. "From Promoter Sequence to Expression: A Probabilistic Framework." RECOMB, 2002.
- Segal E, Battle A, Koller D. In *Proc. Pacific Symposium on Biocomputing*, Kauai, Hawaii, 2003.
- ~~Segal E, Barash, Y., Simon, I., Friedman, N., and Koller, D. "From Promoter Sequence to Expression: A Probabilistic Framework." RECOMB, 2002.~~
- Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, Eisen MB, Brown PO, Botstein D, Futcher B. *Mol. Biol. Cell* 9:3273-3297, 1998.
- Taleb, A. and Jutten, C. 1999. *IEEE Transaction on Signal Process* 47(10):2807-2820.
- Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan E, Dmitrovsky E, Sander ES, Golub TR. *Proc. Natl. Acad. Sci. USA* 96:2907–2912, 1999.
- Tavazoie S, Hughes JD, Campbell MJ, Cho RJ, Church GM. *Nature Genetics* 22(3):281–285, 1999.
- Troyanskaya O, Cantor M, Sherlock G, Brown PO, Hastie T, Tibshirani R, Botstein D, Altman RB. *Bioinformatics* 17:520–525, 2001.
- ~~Vigario R. *Electroenceph. Clin. Neurophysiol.* 103(3):395–404, 1997.~~
- Vigario R, Jousmaki V, Hamalainen M, Hari R, Oja E. In *Advances in Neural Information Process (Proc. NIPS'97)* 10:229–235, Cambridge, MA, MIT Press, 1998.

WEB SITE REFERENCES

- <http://www.stanford.edu/~silee/ICA/> (web supplement).
- <http://cellcycle-www.stanford.edu> (Dataset 1).
- http://arep.med.harvard.edu/network_discovery/ (Dataset 2).
- http://www-genome.stanford.edu/yeast_stress (Dataset 3).
- <http://cmgm.stanford.edu/~kimlab/topomap> (Dataset 4).
- <http://www.hugeindex.org> (Dataset 5).
- <http://www.cnl.salk.edu/~tewon/ICA/code.html> (Infomax software).
- <http://www.cis.hut.fi/~aapo/> (Fast ICA software).



E.

Figure 1. Comparison of linear ICA (NMLE) with other approaches, in four datasets from yeast and *C.elegans*. ICA is compared against PCA, k-means clustering, the Bayesian approach, the Plaid model, and the topomap approach. For each functional category within GO and KEGG, the value of $-\log_{10}(\text{p-value})$ with the smallest p-value from one method is plotted against the corresponding value from the other method. Overall, the clusters found by NMLE show lower p-values than those found by the other approaches, except for the topomap approach which performed slightly better than NMLE on the *C. elegans* dataset. More detailed comparison of ICA with other approaches is found in our webpage, <http://www.stanford.edu/~silee/ICA>.