Learning Statistical and Geometric Models from Microarray Gene Expression Data

Yitan Zhu

Dissertation submitted to the faculty of the Virginia Polytechnic
Institute and State University in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
In
Electrical Engineering

Yue Wang, Chairman
Jianhua Xuan
Christopher L. Wyatt
Chang-Tien Lu
Amir I. Zaghloul

(September 2, 2009)
Arlington, Virginia

Keywords: Data Clustering and Visualization, Clustering Evaluation, Blind Source Separation, Convex Analysis and Optimization, Gene Expressions.

Learning Statistical and Geometric Models from Microarray Gene Expression Data

Yitan Zhu

ABSTRACT

Analysis of microarray gene expression data is important for disease study at the molecular and genomic level. Computational data modeling and analysis are essential for extracting meaningful and specific information from noisy, high-throughput, and large-scale microarray gene expression data. In this dissertation, we propose and develop innovative data modeling and analysis methods for learning statistical and geometric models from gene expression data and subsequently discover data structure and information associated with disease mechanisms.

To provide a high-level overview of gene expression data for easy and insightful understanding of data structure relevant to the physiological event of interest, we propose a novel statistical data clustering and visualization algorithm that is comprehensive and effective for multiple clustering tasks and that overcomes some of the major limitations associated with existing clustering methods. The proposed clustering and visualization algorithm performs progressive, divisive hierarchical clustering and visualization, supported by hierarchical statistical modeling, supervised/unsupervised informative gene/feature selection, supervised/unsupervised data visualization, and user/prior knowledge guidance through human-data interactions, to discover cluster structure within complex, high-dimensional gene expression data. Applications to muscular dystrophy, muscle regeneration, and cancer data demonstrated its abilities to identify functionally enriched (co-regulated) gene groups, detect/validate disease types/subtypes, and discover the pathological relationship among multiple disease types reflected by gene expression profiles.

For the purpose of selecting suitable clustering algorithm(s) for gene expression data analysis and validating the advantage of our proposed clustering algorithm, we design an objective and reliable clustering evaluation scheme to assess the performance of clustering algorithms by comparing their sample clustering outcome to definitive ground truth, i.e. phenotype categories. Using the proposed evaluation scheme, we compared the performance of our newly developed clustering algorithm with those of several benchmark clustering methods from three aspects, i.e. algorithm functionality, clustering accuracy, and outcome reproducibility, and demonstrated the superior and stable performance of the proposed clustering algorithm.

To identify the underlying and activated biological processes that jointly form the observed (dynamic) biological event, we propose a latent linear mixture model that quantitatively describes how the observed gene expressions are generated by a process of mixing the latent active biological processes. We prove a series of theorems to show the identifiability of the noise-free model. Based on relevant geometric concepts, convex analysis and optimization, gene clustering, and model stability analysis, we develop a robust blind source separation method that fits the model to the gene expression data and subsequently identify the underlying biological processes and their activity levels under different biological conditions (or at successive time points). The proposed method provides improved model identification accuracy over several benchmark blind source separation methods, evaluated by their abilities to separate numerically mixed gene expression data. Applications to muscle regeneration and muscular dystrophy data demonstrated the applicability of the proposed method to discover critical yet hidden biological processes that form the observed biological phenomenon and are closely related to the underlying disease mechanisms.

Based on the obtained experimental results, we believe that the research work presented in this dissertation not only contributes to the engineering research areas of machine learning and pattern recognition, but also provides novel and effective solutions to potentially solve many biomedical research problems, for improving the understanding about disease mechanisms and stimulating novel hypotheses for further study. The value of our novel research for microarray gene expression data analysis has also been demonstrated by its actual utilities in biomedical research projects conducted at several well-known biomedical institutions.

Acknowledgements

First and Foremost, appreciation goes to my parents, Guanghua Zhu and Hexiang Liu, for everything they have offered me in my life. Their encouragement and support have helped me to face up to difficulties and accomplish goals in life and work. I wish they share the joy of my achievement and be proud of me.

I would like to express my deep gratitude to my advisor, Dr. Yue Wang, who has been guiding my research for more than five years and directly involved in many aspects of the research presented in this dissertation. His profound influence on me as a mentor and researcher has leaded me to enjoy the beauty of research work and also inspired my thoughts. I thank him for the countless hours he has spent with me, seeking important research topics, forming effective solutions, sharpening my skills as a competent researcher, and helping me out in research writing. The experience of working with Dr. Wang is an enjoyment that I will never forget and an asset that will always benefit me.

Many thanks go to Dr. Jianhua Xuan, Dr. Huai Li, and Dr. David J. Miller, whose guidance and suggestions have helped to improve the quality of my research and publications. Working with them gave me an opportunity to broaden my vision and learn research skills of multiple aspects.

I want to express my sincere gratitude to the other committee members, Dr. Christopher L. Wyatt, Dr. Chang-Tien Lu, and Dr. Amir I. Zaghloul. They have provided important insights that are very precious to me and their suggestions have significantly improved this dissertation.

I would like to thank all my co-workers and labmates in the Computational Bioinformatics and Bioimaging Laboratory. I will always remember the days that I worked with them on challenging and interesting problems. Their suggestions also contributed to the formation of the research solutions presented here.

Special thanks to our collaborators, especially Dr. Robert Clark and Dr. Eric P. Hoffman, who helped to propose the biological problems to be studied and provided biological interpretations of the obtained computational results.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

APC	Affinity Propagation Clustering
BSS	Blind Source Separation
CAM	Convex Analysis of Mixtures
cDNA	complementary DeoxyriboNucleic Acid
DCA	Discriminatory Component Analysis
DNA	DeoxyriboNucleic Acid
EM	Expectation Maximization
FOM	Figure of Merit
HC	Hierarchical Clustering
ICA	Independent Component Analysis
i.i.d.	independently identically distributed
IPA	Ingenuity Pathway Analysis
KMC	K-Means Clustering
LPP	Locality Preserving Projection
MAP	Maximum A Posteriori probability
MCLL	Mean Classification Log-Likelihood
MDL	Minimum Description Length
MLL	Mean Log-Likelihood
mRNA	messenger RiboNucleic Acid
MSC	Mean Squared Compactness
nBSS	non-negative Blind Source Separation
nICA	non-negative Independent Component Analysis
NMF	Non-negative Matrix Factorization

PCA	Principal Component Analysis
PCG	Process Common Gene
PPM	Projection Pursuit Method
PSG	Process Specific Gene
SFNM	Standard Finite Normal Mixture
SNICA	Stochastic Non-negative Independent Component Analysis
SNMF	Sparse Non-negative Matrix Factorization
SNR	Signal to Noise Ratio
SOM	Self-Organizing Maps
SRBCTs	Small, Round Blue-Cell Tumors
TOP	Tree Of Phenotypes
tRNA	transfer RiboNucleic Acid
VISDA	VIsual Statistical Data Analyzer
WGP	Well-Grounded Point

1 Introduction

1.1 Motivation

The general goal of biomedical research is to discover disease mechanisms, help medical diagnosis, and develop cures to reduce the burden of human diseases, and suggest prevention strategies to improve human health. In recent years, biomedical research has been attracting great interest and huge investments, and entered a new era largely due to the rapid development of molecular biology and genome research. Many diseases such as cancer and muscular dystrophy have genetic factors that are wholly or partially responsible for the disease. It is widely believed that genome studies will transform our understanding of the normal and abnormal mechanisms underlying the functions of cells and organisms, and revolutionize the diagnosis and management of disease by offering an unprecedented comprehensive view of the molecular underpinnings of pathology [1]. For example, disease study at the molecular and genomic level has triggered new concepts and methodologies for disease treatment such as gene therapy that modifies genome in human cells and tissues to treat a disease and personalized medicine that tailors medical care to an individual's needs according to a patient's genotype.

1.1.1 Background Knowledge

It has been one of the most exciting scientific discoveries that the hereditary information of biological organism is encoded in DeoxyriboNucleic Acid (DNA). Figure 1.1a is an illustration of the famous double helix structure of DNA that consists of two long strands of polymers formed by nucleotides, and the backbone of each strand is made of sugars and phosphate groups joined by ester bonds. Figure 1.1b shows the chemical structure of DNA double helix, from which we can also see that attached on each sugar in the phosphate-deoxyribose backbone is one of the four different nucleobases, i.e. cytosine, guanine, adenine, and thymine. It is the sequence of these four different nucleobases that encodes the hereditary information of biological organism. In the double helix of DNA, adenine pairs and bonds with thymine, and guanine pairs and bonds with cytosine. The two strands run in opposite directions to each other and are therefore anti-parallel. One strand is from 5’ end to 3’end, while the other strand is from 3’end to 5’end, as shown in Figure 1.1b.

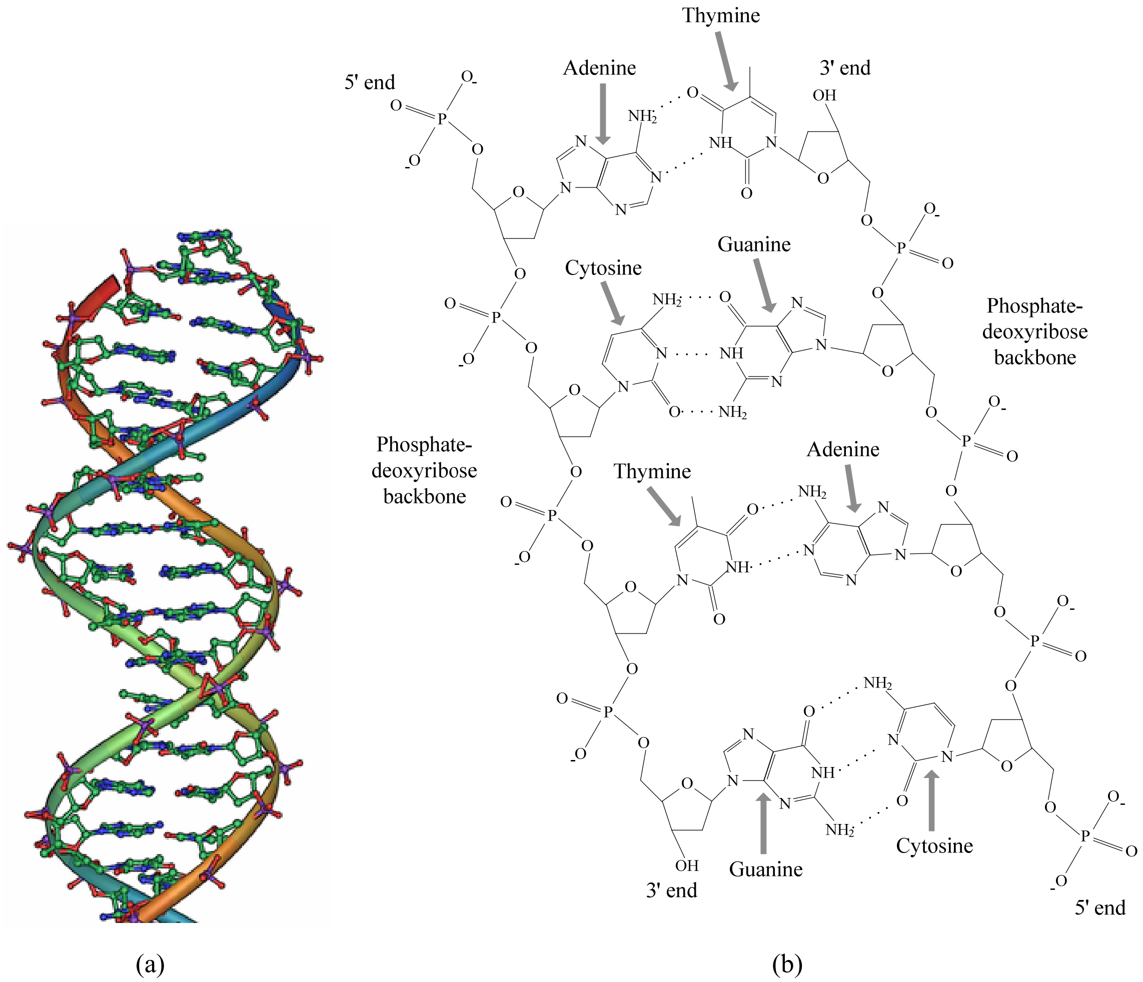


Figure 1.1 Double helix and chemical structures of DNA. (a) Double helix structure of a section of DNA. (Public domain image from Wikimedia Commons, http://commons.wikimedia.org/wiki/File:DNA_double_helix_horizontal.png) (b) Chemical structure of DNA. Hydrogen bonds are shown as dotted lines.

The most fundamental pathway in biological system is the transcription of DNA to messenger RiboNucleic Acid (mRNA) and the translation of mRNA to protein that is the building block of biological system. Figure 1.2 illustrates the processes of transcription and translation. Within the cell nucleus, when transcription starts, the double helix of DNA opens up. The sequence information of a DNA strand is complimentarily transcribed to a single-strand mRNA, which is also a long chain of nucleotides. In mRNA, the nucleobases are cytosine, guanine, adenine, and uracil. Uracil, instead of thymine, pairs and bonds with adenine in mRNA. Transcription basically transfers the coding information of DNA to mRNA. After transcription and several further processes on mRNA, e.g. capping, polyadenylation, and splicing, the mRNA matures and is transported to the cytoplasm, where amino acids are assembled according to the sequence information encoded in mRNA to form a chain, with help from other molecules and complexes, like transfer RNA (tRNA) and ribosome. The assembled amino acid chain after further processes, such as folding, becomes protein.

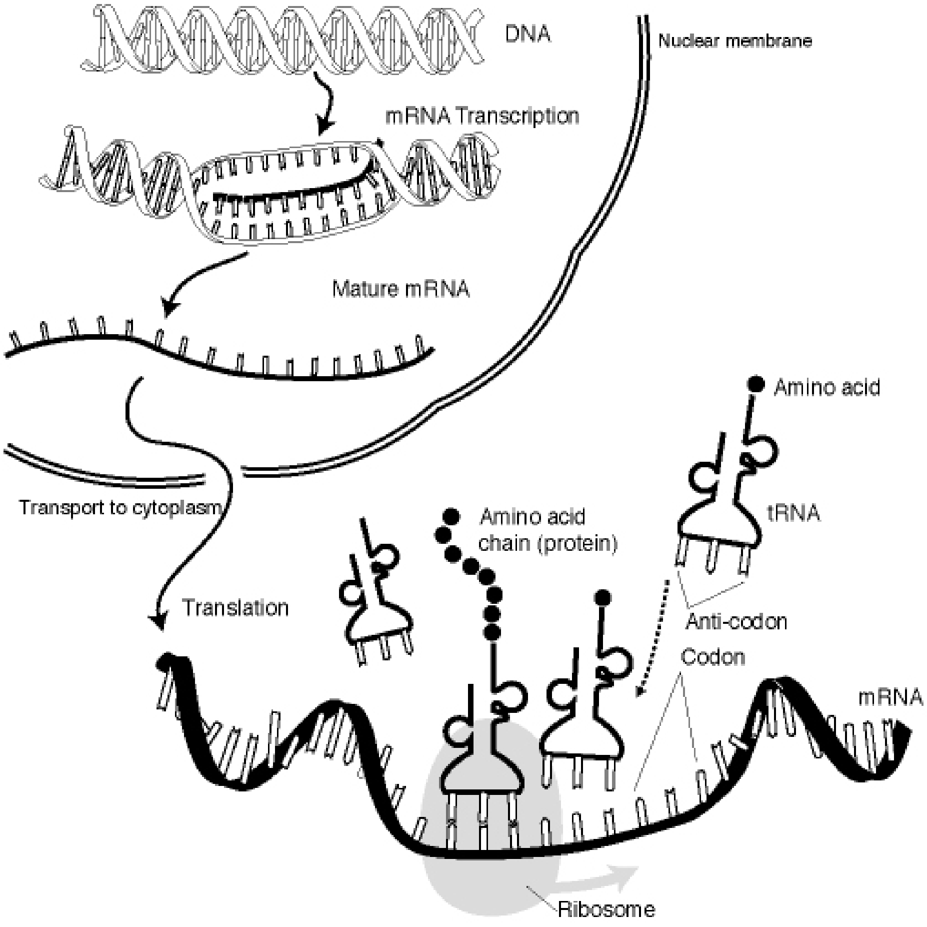


Figure 1.2 Transcription of DNA to mRNA and translation of mRNA to protein. (Public domain image from National Human Genome Research Institute, Courtesy: National Human Genome Research Institute, http://www.genome.gov/Pages/Hyperion/DIR/VIP/Glossary/Illustration/gene_expression.cfm?key=gene%20expression)

1.1.2 Microarray Technology

Since transcription of DNA is the first step in generating proteins, measuring the amount of mRNA transcripts corresponding to a gene, which is also called the expression level of a gene, is very important for studying biological systems. There exist multiple technologies for measuring gene expression levels [1-3]. Among them gene expression microarray is currently the most popular one. A gene expression microarray, as shown in Figure 1.3, is a glass or silicon chip, attached to which are arrayed series of thousands of microscopic spots of DNA oligonucleotides, each containing picomoles of a specific DNA sequence. After hybridization with fluorophore-labeled complementary DNA (cDNA, which is the DNA reversely transcribed from mRNA), the intensity of fluorescence light reflects the abundance of cDNA sequences, and thus correspondingly indicates the abundance of mRNA sequences in the original mRNA sample.

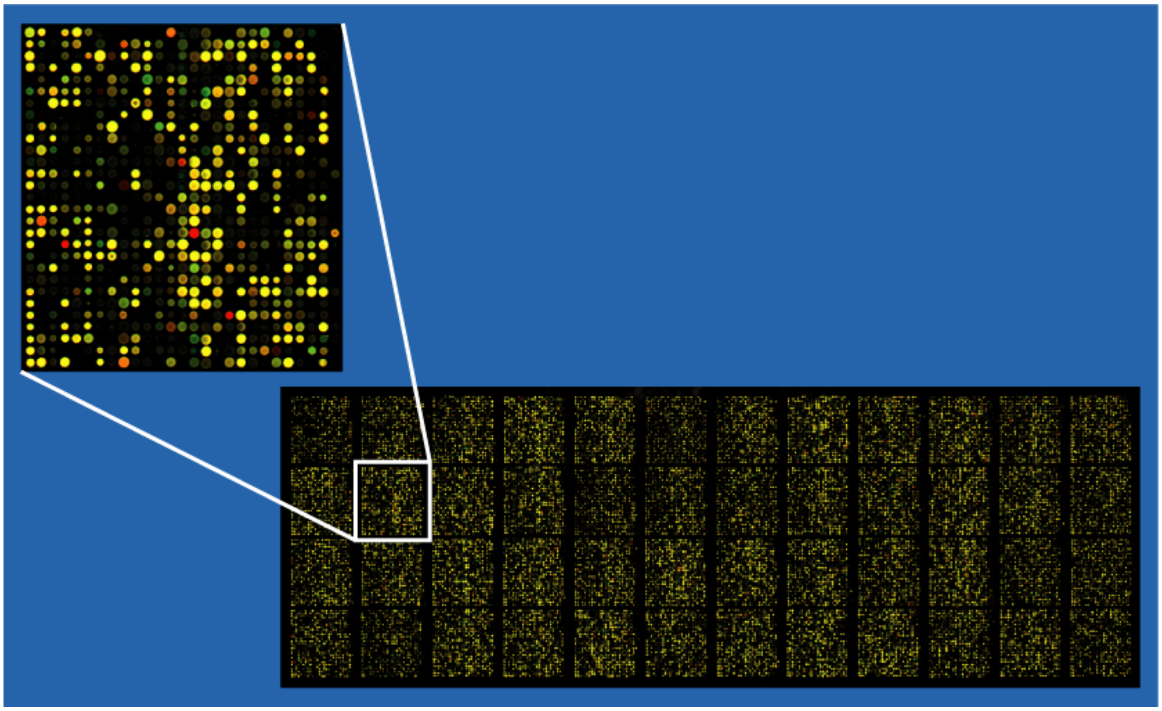


Figure 1.3 Gene expression microarray. (Public domain image from Wikimedia Commons, http://en.wikipedia.org/wiki/File:Microarray2.gif)

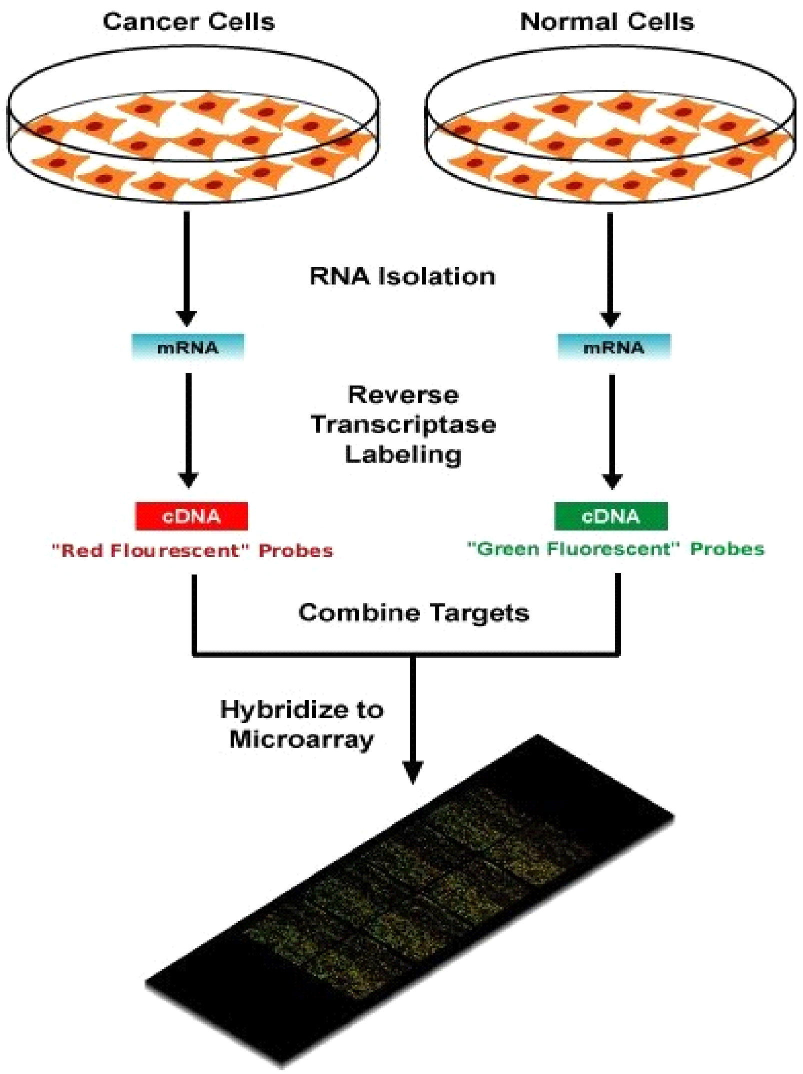


Figure 1.4 Experimental procedure of two-channel microarrays. (Public domain image from Wikipedia, http://en.wikipedia.org/wiki/File:Microarray-schema.jpg)

Microarray gene expression data can be generated in two ways, namely, two-channel microarray and one-channel microarray. Figure 1.4 shows the procedure of generating a two-channel microarray. Two mRNA samples, i.e. a control sample (normal cells in Figure 1.4) and a target sample (cancer cells in Figure 1.4), are reversely transcribed to cDNA samples that are labeled by two fluorophores of different colors, i.e. red and green. The two cDNA samples are mixed together and hybridized to a single microarray. Scanning the hybridized microarray produces an image containing colored spots (see Figure 1.3) whose color intensity indicates the relative abundance of a particular mRNA sequence. Red and green indicate up-regulation and down-regulation of the gene in the target sample compared to the control sample, respectively. In the single-channel microarray experiment, there is no control sample. The target sample is reversely transcribed from mRNA to cDNA and dyed with one fluorophore. Hence, after hybridization, the intensity of the array spots indicates the absolute value of mRNA quantities in the target sample, rather than the relative abundance in the two-channel microarray. For both the two-channel microarray and the single-channel microarray, certain post-processing (e.g. normalization) of the obtained data is needed to achieve comparability among multiple gene expression profiles and multiple experiments [4, 5].

1.1.3 Gene Expression Data and Bioinformatics

With respect to the study design of microarray experiments, gene expression data can be roughly classified as either dynamic data or static data. Dynamic data represent the gene expression levels of samples taken at successive times in a dynamic biological process, such as muscle regeneration (the generation of new muscle tissue after muscle tissue has been damaged). In dynamic data, the time order of the samples is important, and the changes of gene expression levels over time are the information of interest. Gene expression data where the samples do not have a time order belong to the category of static data, for example, phenotypic data that represent gene expression levels of samples from one or multiple phenotypes.

High-throughput microarray technology simultaneously measures the expression levels of thousands of genes and produces a vast amount of data. Gene expression data contain significant noise that exists both in the biological system to be studied and in different stages of assaying gene expression [1, 6]. Thus, extracting true and meaningful information about the biological system from noisy and large-scale microarray gene expression data is a significant challenge, which underscores the importance of computational analysis as a bridge from data generation to the formulation of new knowledge. It is at this point where bioinformatics can play a role in sifting valuable nuggets of knowledge from the widening river of data [6].

Bioinformatics approaches for gene expression data analysis use statistical and deterministic computational methods to study the data. There is no one-size-fits-all solution for the analysis and interpretation of genome-wide expression data [6], due to the complexity of biological systems, various goals of disease study, and different experiment designs. Many bioinformatics tasks have been proposed for the analysis of gene expression data, for example, detection of differentially expressed genes between different biological conditions (or time points) [7, 8], detection of co-expressed genes/samples [9, 10], classification of new samples into known disease/phenotype categories [10, 11], and inference of gene regulatory networks and pathways [12].

Analysis of gene expression data faces its own challenges, due to the unique data characteristics. First, gene expression data usually have tens of samples but thousands of genes. When the samples are objects to be modeled and analyzed, computational methods will usually face the problem of insufficient data for an accurate estimate of the model parameters, due to the "curse of dimensionality". Second, although microarrays simultaneously measure the expression levels of thousands of genes, usually only a small portion of the genes are actually involved in the physiological event of interest, while the other genes are not related to the goal of study. Identification of the genes related to the study is a difficult but necessary step. Third, the existence of significant noise in the gene expression data requires that the computational methods to be robust/resistant against noise impact.

1.2 Objectives and Statement of Problems

1.2.1 Data Clustering for High-level Overview of Dataset

Although molecular biology and genome research are experiencing rapid development, existing knowledge about disease initiation, progression, and response to therapy at the molecular and genomic level is still very limited. When facing a dataset without much prior knowledge, the first task is usually to generate a high-level overview of the dataset so that the data analyzers (usually biologists) can quickly understand/grasp the major data structures within the dataset. The basic strategy for providing a high-level overview of a dataset is to group similar data points together, so that we can discover what and how many different patterns the data points have, as well as which part of the data follows a particular pattern.

A popular unsupervised learning solution for discovering similar data points and distinct data patterns is data clustering that groups data points into clusters so that data points within a cluster are similar while data points in different clusters are dissimilar [13]. The obtained cluster exemplars (e.g. cluster centers), whose number is usually much smaller than that of the data points, can be taken as a representation of the whole dataset for a high-level overview. Data clustering can be practically taken as a data compression step to reduce the size and complexity of data, for the ease of understanding data. Because the discovered patterns are cluster exemplars calculated based on all the data points within the cluster, they are usually more reliable and robust than individual data points that may be heavily influenced by noise or outlier factors. Usually, cluster discovery is taken as the first step to analyze a new dataset without much prior knowledge, and other analytical methods may be subsequently involved [6].

In the context of gene expression data analysis, being similar means being co-expressed. Data clustering on gene expression data, including gene clustering and sample clustering, discovers co-expressed genes and co-expressed samples. The biological conjectures driving the discovery of co-expressed genes and co-expressed samples are the following: co-expressed genes may be co-regulated by certain genomic regulation mechanisms and thus have similar genomic functions; co-expressed samples may represent a known/unknown phenotype (e.g. disease type). To give a rough idea of data clustering on gene expression data, Figure 1.5 shows the heat map of sample clustering result obtained on the Small, Round Blue-Cell Tumors (SRBCTs) microarray gene expression dataset [14].

Data clustering has been widely applied to gene expression data to find co-expressed genes and samples [9, 10, 15-17]. However, most existing clustering methods used in gene expression data analysis have some major limitations, including heuristic or random algorithm initialization, the potential of finding poor local optima, the lack of cluster number detection, an inability to incorporate prior/expert knowledge, black-box and non-adaptive designs, in addition to the curse of dimensionality and the discernment of uninformative, uninteresting cluster structure associated with confounding variables, and we will give a more detailed review on existing clustering methods and their limitations in Section 2.1. Thus there is an urgent demand for developing novel clustering methods that can at least partially overcome some of the limitations. Besides gene clustering and sample clustering that are data clustering tasks targeted by most existing clustering methods, we propose phenotype clustering that groups similar known phenotypes (groups of samples with the same phenotype label) together for gene expression data analysis. If the clustering algorithm can characterize the relationship among the detected phenotype clusters, phenotype clustering may discover important pathological relationships among the phenotypes reflected at the mRNA level.

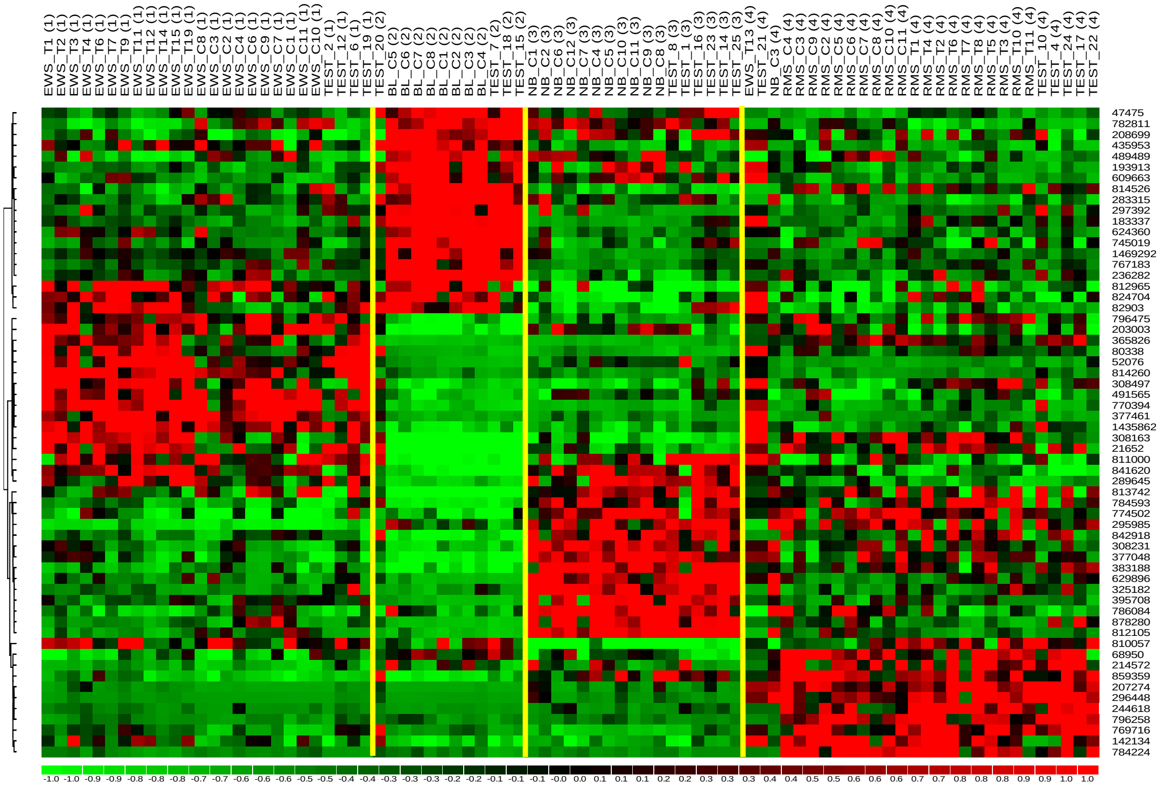


Figure 1.5 Heat map showing a sample clustering result obtained on the SRBCTs gene expression dataset. Cluster boundaries are indicated by yellow lines. Red color indicates up-regulation and green color indicates down-regulation. Each column is a sample, whose sample ID is shown on the top. Each row is a gene, whose probe set ID is shown on the right.

In this dissertation, **the first research topic** to be addressed is that by developing an effective data clustering method for pattern discovery on gene expression data, we want to seek biologically meaningful solutions to the following tasks:

(1) Provide a high-level overview of the dataset for easy understanding.

(2) Identify co-expressed genes and their expression patterns.

(3) Identify co-expressed samples and their expression patterns.

(4) Identify co-expressed phenotypes and discover the relationship among the phenotypes at the mRNA level.

1.2.2 Clustering Evaluation

A problem always associated with data clustering is performance evaluation, i.e. assessing how well the clustering algorithm performs. Given the variety of available clustering methods for gene expression data analysis, it is important to develop an appropriate and rigorous validation scheme to assess the performance and limitations of the most widely used clustering algorithms to guide users in selecting a suitable clustering algorithm for the analysis task at hand.

Evaluations of clustering of gene expression data have been proposed and tackled by a number of research groups, as we will review in Section 3.1. However, these evaluation schemes assess clustering algorithms based on either the stability of clustering outcomes obtained in the presence of data perturbation, or the consistency between gene clustering outcome and gene annotations like GO classes [18] that are knowledge-based annotations including genes' genomic function categories. The assessment based on some form of stability analysis does not constitute a ground truth based gold standard that is usually needed for algorithm outcome evaluation. Knowledge-based evaluation schemes use gene annotations as the gold standard, but gene annotations are prone to significant "false positive evidence" when used under biological contexts different from those that produced the annotations, and most GO-like databases only provide partial gene annotations. Thus results provided by such evaluation schemes cannot be considered as conclusive.

Therefore, there is an urgent need for developing an objective and reliable clustering evaluation scheme that utilizes concrete and complete ground truth as the gold standard, which leads to **the second research topic** that will be addressed in this dissertation. It can be stated as the following: what is a reliable, statistically rigorous and biologically meaningful clustering evaluation scheme to guide users in the selection of clustering algorithm for gene expression data analysis?

1.2.3 Gene Expression Dissection for Identification of Underlying Biological Processes

Although data clustering can provide a high-level overview of the gene expression dataset, it is not sufficient for revealing the underlying mechanism of observed biological phenomenon that is often a joint effect of multiple underlying activated biological processes. A biological process is a collection of molecular events specifically pertinent to the functioning of integrated living units: cells, tissues, organs, and organisms [18]. Each biological process is characterized by a group of genes that are key players for the biological process to fulfill its specific biological functions. Dissection of observed gene expression data into gene expression patterns that correspond to underlying biological processes, identification of the key genes of the biological processes, and estimation of the activity levels of the biological processes in different biological conditions (or at successive time points), will help understand the complex mechanisms of biological system and will be useful for multiple biomedical research purposes. For example, based on the estimated time activity curves of the underlying biological processes, we may figure out the most critical time period of the dynamic biological event and design further experiments that focus on such specific time period to study the events of interest.

Accordingly, **the third research topic** that will be addressed in this dissertation is to develop a computational method to dissect the gene expression data, for answering the following questions:

(1) What are the underlying active biological processes that form the observed biological phenomenon?

(2) What are the key genes involved in each biological process?

(3) What are the activity levels of biological processes in different biological conditions (or at successive time points)?

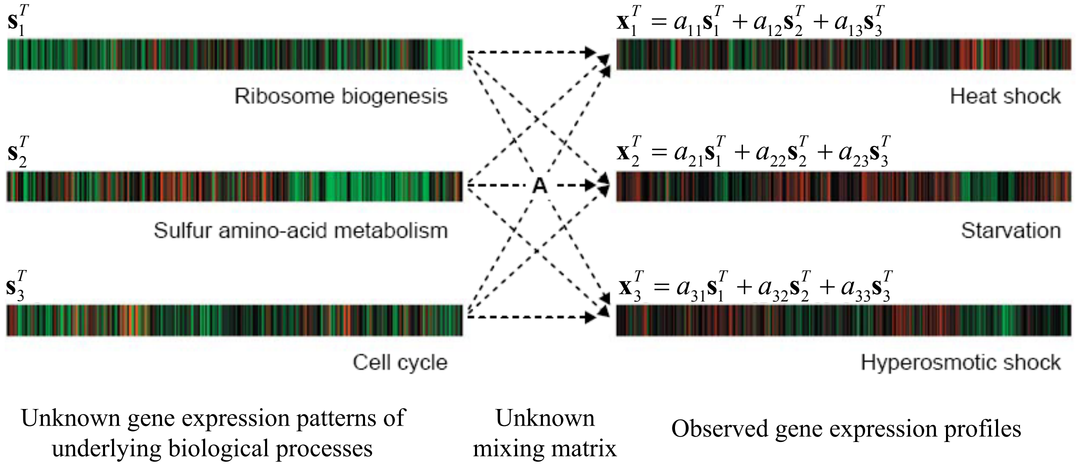

$\mathbf{s}_1^T$
Ribosome biogenesis
$\mathbf{s}_2^T$
Sulfur amino-acid metabolism
$\mathbf{s}_3^T$
Cell cycle
A
$\mathbf{x}_1^T = a_{11}\mathbf{s}_1^T + a_{12}\mathbf{s}_2^T + a_{13}\mathbf{s}_3^T$
Heat shock
$\mathbf{x}_2^T = a_{21}\mathbf{s}_1^T + a_{22}\mathbf{s}_2^T + a_{23}\mathbf{s}_3^T$
Starvation
$\mathbf{x}_3^T = a_{31}\mathbf{s}_1^T + a_{32}\mathbf{s}_2^T + a_{33}\mathbf{s}_3^T$
Hyperosmotic shock
Unknown gene expression patterns of underlying biological processes
Unknown mixing matrix
Observed gene expression profiles

Figure 1.6 Illustration of the relationship between the observed gene expression profiles and the gene expression patterns of underlying active biological processes [19]. Red color indicates up-regulation. Green color indicates down-regulation.

The relationship between the observed gene expression data and the underlying active biological processes can be described by a linear mixture model **X** = **AS** [19, 20], where **X** is a known data matrix containing observed mixtures (i.e. observed gene expression profiles in the context of gene expression data analysis), **A** is an unknown mixing matrix, and **S** is an unknown non-negative data matrix containing underlying sources (i.e. gene expression patterns of underlying active biological processes in the context of gene expression dissection). The mixing coefficients in **A** indicate the activity levels of the biological processes under different biological conditions that generate the observed gene expression profiles. Figure 1.6 is an illustration of using linear mixture model to describe the relationship between the observed gene expression profiles and the gene expression patterns of underlying biological processes [19]. This illustration figure was drawn based on yeast gene expression data [19]. It shows the case that there are three observed gene expression profiles and three underlying biological processes, which can be mathematically formulated as

$$\mathbf{AS} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1^T \\ \mathbf{s}_2^T \\ \mathbf{s}_3^T \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \mathbf{x}_3^T \end{bmatrix} = \mathbf{X}, \tag{1.1}$$

where the superscript T indicates the transpose of vector, $\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{s}_3$ denote the gene expression patterns of three underlying biological processes, and $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ denote the three observed gene expression profiles. On the left side of the figure are the unknown gene expression patterns of underlying biological processes, i.e. ribosome biogenesis, sulfur amino-acid metabolism, and cell cycle. Their linear combinations are shown on the right side of the figure, which are the observed gene expression profiles of yeast under different stimuli, i.e. heat shock, starvation, and hyperosmotic shock.

In the linear mixture model, the only available information comes from the observed mixtures, while the underlying sources and their mixing coefficients need to be discovered. Such a modeling forms a typical Blind Source Separation (BSS) problem [21]. Because the gene expression patterns of underlying biological processes are non-negative, gene expression dissection is actually a non-negative Blind Source Separation (nBSS) problem [22]. As we will review and summarize in Sub-section 4.1.1, existing nBSS solutions make stringent model assumptions that may be inconsistent with the biological system, or have non-unique solutions leading to the possibility of convergence to a poor local optimum, or turn to be very sensitive to noise in data. Therefore, there is an urgent need for developing an nBSS solution with model assumptions characterizing the biological system, while the model can be well (and uniquely) identified through a mathematically rigorous and noise-resistant way.

1.3 Organization of Dissertation

The main research topics discussed in this dissertation include data clustering and visualization, performance evaluation of clustering algorithms, and gene expression dissection for identification of underlying biological processes. Many research problems, such as hierarchical mixture model fitting, supervised/unsupervised feature/gene selection, supervised/unsupervised data projection for visualization, cluster number detection on high-dimensional data, evaluation of the biological relevance of clustering outcome, evaluation of the reproducibility of clustering outcome, identifiability of the latent linear mixture model, noise reduction of nBSS solution, and model order selection in nBSS problem, which are important components of the above three research topics are discussed and studied. Many experiments on both synthetic data and real microarray gene expression data are conducted to verify the ideas that formulate the related problems, to examine the algorithms for solving these problems, and to demonstrate their potential practical applications.

Chapter 2 addresses the first research topic: cluster modeling, visualization, and discovery on gene expression data. A detailed review on existing clustering methods and their major limitations for gene expression data analysis is given. To (at least partially) overcome these limitations, a novel clustering and visualization algorithm is proposed. The main steps of the algorithm that are directly applicable for the analysis task of gene clustering are given first. Then the algorithm is extended to work on sample clustering (by adding unsupervised informative gene selection as a data pre-processing step) and phenotype clustering (by incorporating a cluster visualization and decomposition scheme that explicitly utilizes the phenotype category information). Applications of the proposed clustering and visualization algorithm to gene clustering, sample clustering, and phenotype clustering tasks on multiple microarray gene expression datasets are performed and the obtained experimental results are presented to demonstrate the performance of the proposed clustering method and its utility in biomedical research.

Chapter 3 addresses the second research topic: clustering evaluation for guiding users in the selection of clustering algorithm for gene expression data analysis. We first review existing clustering evaluation schemes and summarize their major limitations. Then we propose a ground-truth based clustering evaluation scheme, i.e. comparing sample clustering outcome to definitive ground truth, defined by phenotype categories in the data. Multiple objective and quantitative performance measures are designed, justified, and formulated to assess clustering performance. Using the proposed clustering evaluation scheme and criteria, we compare the performance of our newly developed clustering algorithm with those of several benchmark clustering methods, to show the superior and stable performance of the proposed clustering algorithm.

In Chapter 4, we address the third research topic: gene expression dissection for identification of underlying biological processes. Existing nBSS solutions are viewed and their limitations for gene expression dissection are summarized. We propose a linear mixture model suitable for gene expression dissection, prove a series of theorems to show the identifiability of the noise-free model, and develop the algorithm to identify it. Based on the study of noise-free model, a comprehensive solution that is applicable to noisy practical tasks is further developed and tested on synthetic data. The performance of the proposed solution is compared with those of several benchmark nBSS solutions based on their abilities to separate numerically mixed gene expression data. Applications of the proposed solution to several microarray gene expression datasets are performed and the obtained experimental results are presented to demonstrate its ability of discovering critical yet hidden biological processes.

Chapter 5 summarizes the original contributions of the dissertation research, proposes several tasks/problems for future research, and presents conclusions for the conducted research work.

2 Cluster Modeling, Visualization, and Discovery on Gene Expression Data

2.1 Introduction

Due to limited existing biological knowledge at the genomic and molecular level, data clustering for constructing a high-level overview of the dataset has become a popular and effective method to extract interesting and meaningful information from microarray gene expression data. Data clustering can help to discover novel functional gene groups, gene regulation networks, phenotypes/sub-phenotypes, and developmental/morphological relationships among phenotypes [10, 17, 23-27]. Due to the complex and challenging nature of the task, various clustering algorithms have been proposed and applied for gene expression data analysis [13, 16, 25, 28]. We review these methods here by classifying them using different taxonomies [13, 16, 28]. With respect to mathematical modeling, clustering algorithms can be classified as model-based methods like mixture model fitting [29-31], or “nonparametric” methods such as the graph-theoretical methods [15, 32]. Regarding the clustering scheme, there are agglomerative methods, such as conventional Hierarchical Clustering (HC) [17], or partitional methods including Self-Organizing Maps (SOM) [9, 10] and K-Means Clustering (KMC) [33]. The assignment of data points to clusters can be achieved by either soft clustering methods like fuzzy clustering [34-36] and mixture model fitting [29-31], or hard clustering methods like HC and KMC. Some clustering methods such as HC perform the algorithm once and obtain the clustering outcome, while other methods like KMC may run the algorithm multiple times with random algorithm initialization to obtain an optimal clustering outcome measured by a specific criterion function. Additionally, stability analysis based consensus clustering produces the clustering outcome by finding the consensus among different clustering partitions obtained in the presence of data perturbation [37].

While there is a rich variety of existing clustering methods, unfortunately, most of them suffer from several major limitations when clustering gene expression data. The limitations are briefly summarized as follows.

(1) Initialization Sensitivity and Local Optimum. There are often multiple local optimums associated with the objective function of partitional clustering methods, such as KMC and mixture model fitting. Standard learning methods optimize the objective function and find the local optimum solution “nearest to’’ the model initialization. More sophisticated methods or initialization schemes, which seek to avoid poor local optimums, although exist, require significant computation and normally do not ensure convergence to the global optimum [38]. The quality of the local optimum solution may be decidedly poor, compared to that of the global optimum. This is especially true for genomic data sets, with high dimensionality and small sample size [39-41], if sample clustering is performed. On the other hand, methods such as conventional agglomerative HC perform a very simple, greedy optimization, which severely limits the amount of search they perform over the space of possible solutions, and thus limits the solution accuracy and optimality.

(2) Lack of Solution Reproducibility. Since clustering may help generate scientific hypotheses, it is extremely important that solutions be reproducible. This refers to the ability of a clustering algorithm to identify the same (or quite similar) structure under different model initializations, in the presence of small data perturbations or additive noise, as well as when analyzing independent data sets drawn from the same underlying distribution. KMC and mixture modeling, which are sensitive to algorithm initialization, do not yield reproducible solutions when random algorithm initialization is applied. On the other hand, agglomerative HC (e.g., the single linkage algorithm [40]) is highly sensitive to some particular data samples and to the noise in the data.

(3) Difficulty of Model Order Selection: Estimating the Number of Clusters. For hierarchical clustering, simple heuristics are typically applied to estimate the number of clusters, such as identifying the cluster number at which a large improvement in fitting accuracy occurs [40, 42, 43]. For mixture modeling, information-theoretic model selection criteria such as Minimum Description Length (MDL) are often applied [44, 45]. These criteria consist of a data fitting term (mismatch between data and model) and a model complexity term, and the optimum model is defined as the one with minimum criterion value. However, when applied to genomic data with high feature/gene dimensionality and small sample size, such criteria may grossly fail in estimating the cluster number due to inaccurate parameter estimation resulting from the “curse of dimensionality” or due to too many freely adjustable parameters [39, 46]. As one alternative solution, stability analysis has been applied to model order selection [47-49], where the stability of models with different model order is examined in the presence of data perturbation and the “correct” model order is defined as that of the most stable model.

(4) Challenge of Unsupervised Informative Gene Selection for Sample Clustering. Unsupervised informative gene selection for sample clustering is a critical yet difficult problem due to the existence of many irrelevant genes respective to the phenotypes/sub-phenotypes of interest [16, 28]. Without first possessing correct sample clusters, it is difficult to identify relevant genes, and without good gene selection to eliminate many noisy/irrelevant ones, it is very difficult to discern the true underlying sample cluster structure. Existing iterative algorithms wrapping gene selection around sample clustering were developed and tested for the two-cluster case [31, 50]. More research effort targeting unsupervised gene selection for the multi-cluster case is needed.

(5) Influence of Confounding Variables and Multiple Sources of Cluster Structure. A fundamental, flawed assumption in much clustering work is that there is a single source of clustering tendency in the data, e.g. “disease’’ vs. “non-disease’’. Actually, there may be other sources of clustering tendency, based either on measured or even unmeasured (latent) factors, e.g. gender, environment, measurement platform [51], or other biological processes that are peripheral or at any rate not of direct interest in the current study [52]. While there is some research in this area [51], more work is needed in removing or compensating for confounding influences in data clustering [25, 53].

(6) Insufficient Utilization of Prior Knowledge and Human Intelligence. The most fundamental limitation in clustering is the ill-posed nature of the problem. The number of clusters depends on the scale at which one views the data [38] and the data grouping clearly depends on the geometric or statistical assumption/definition of cluster [40, 49]. One potential way to break ill-posedness is to incorporate prior knowledge when such information or opportunity is available. Incorporating prior knowledge may also help to focus the clustering analysis on interesting data structure and remove the effect of confounding variables. Unfortunately, most clustering algorithms do not have mechanisms for exploiting prior information. Exceptions include semi-supervised gene clustering methods that utilize gene annotations to form gene clusters [30, 54, 55] and some other more general semi-supervised methods [56]. Besides auxiliary database information, the user's domain knowledge and the human gifts for pattern recognition and data visualization in low-dimensional spaces can also help to produce more interesting and meaningful clustering outcomes [57, 58]. For example, hierarchical data visualization schemes based on mixture models with human-data interaction were developed [59-61].

(7) Inflexibility and Non-adaptivity of Standard Methods. Most clustering algorithms have a standalone nature and rely on underlying statistical or geometric assumptions about clusters. When these assumptions are violated, the method may fail completely, and with no "backup plan'', i.e. no capability to modify the assumptions to seek an alternative solution. A recent strategy with some ability to mitigate this is the ensemble or consensus clustering [62, 63], wherein multiple algorithms are applied and their results then fused so as to maximize a clustering "consensus'' function. However, these methods, again being fully automated, cannot benefit from domain knowledge and human intelligence, which can guide algorithm choices in a flexible, adaptive manner, to match the underlying data characteristics and the domain-specific clustering structure of interest.

To address some of the existing methods' limitations outlined above and design a comprehensive and flexible tool applicable to cluster modeling, visualization, and discovery on gene expression data, we develop a hierarchical data exploration and clustering approach, called VIsual Statistical Data Analyzer (VISDA) [27, 64, 65]. VISDA performs progressive, divisive hierarchical clustering and visualization, supported by hierarchical mixture modeling, supervised/unsupervised informative gene selection, supervised/unsupervised data projection, and user/prior knowledge guidance through human-data interactions, to discover cluster structure within complex, high-dimensional gene expression data.

The data exploration process in VISDA starts from the top level where the whole dataset is viewed as one cluster, with clusters then hierarchically subdivided at successive levels until all salient structure in the data is revealed. Since a single 2-D data projection/visualization, even if it is nonlinear, may be insufficient for revealing all cluster structure in multimodal, high-dimensional data, the hierarchical visualization and clustering scheme of VISDA uses multiple local projection subspaces (one at each node of the hierarchy) and consequent subspace data modeling to reveal both global and local cluster structures. Consistent with the "divide and conquer" principle, local data projection and modeling can be done with relatively simple methods/models, while the complete hierarchy maintains overall flexibility and conveys considerable clustering information.

The inclusive VISDA framework incorporates the advantages of various complementary data clustering and visualization algorithms to visualize the obtained clusters, which not only produces a "transparent" clustering process that can enhance the user's understanding of the data structure but also provides an interface for incorporating human intelligence (e.g. user's discernment of sub-cluster separability and outliers) and domain knowledge to help improve clustering outcome and avoid finding non-meaningful or confounding cluster structure. Specifically, the interactive user participation guides the coarse-to-fine cluster identification via (1) the selection of a local visualization from a suite of data projections, each sensitive to a distinct type of data structure, for best revealing a cluster's substructure; (2) user-directed parameter initialization for the new sub-clusters that divide existing clusters; and (3) a user-guided model order selection, applied in conjunction with MDL, for deciding the number of sub-clusters in the low-dimensional local visualization space.

Based on its complementary building blocks and flexible functionality, VISDA is comprehensively suitable for multiple genomic data clustering tasks, including gene clustering, sample clustering, and phenotype clustering (wherein phenotype labels for samples are known), albeit with customized modifications for each of these tasks. The main clustering and visualization algorithm of VISDA is readily applicable for gene clustering where the attribute-to-object ratio is low, since there are usually thousands of genes and a relatively few number of samples. For sample clustering, VISDA requires a front-end dimensionality reduction via unsupervised informative gene selection by variation filtering and discrimination power analysis. Variation filtering removes genes that do not have a significant change over different biological conditions. These genes may be irrelevant genes that do not respond to the physiological event. Discrimination power analysis examines each gene's power for discriminating among potential sample clusters and removes those that contribute little to the cluster structure. For phenotype clustering, VISDA exploits the knowledge of phenotype labels in performing supervised informative gene selection, supervised data visualization, and statistical modeling that preserves the unity of samples from the same phenotype, to fulfill that in phenotype clustering known phenotypes, i.e. groups of samples with the same phenotype label, are taken as objects to be clustered. An important goal of phenotype clustering is to discover a Tree Of Phenotypes (TOP), i.e. a hierarchical tree structure with all phenotypes as leaves of the tree, which may reflect important biological relationships among the phenotypes.

Organization of this chapter is the following. In Section 2.2, we introduce the major steps of VISDA algorithm that directly describe the complete VISDA procedure for the task of gene clustering. In Section 2.3, we extend the algorithm to perform sample clustering by adding unsupervised informative gene selection as a data pre-processing step. In Section 2.4, we extend the algorithm for phenotype clustering by incorporating a cluster visualization and decomposition scheme that explicitly utilizes the phenotype category information. Section 2.5 shows a demo application of VISDA on sample clustering. Section 2.6 presents gene clustering results obtained by VISDA. Section 2.7 presents phenotype clustering results obtained by VISDA. Section 2.8 and Section 2.9 provide discussion and conclusion, respectively.

2.2 VISDA Algorithm

Let $\mathbf{t} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_N \mid \mathbf{t}_i \in \mathrm{R}^P, i = 1, 2, \ldots, N\}$ denote N P-dimensional data points of the dataset. Based on a hierarchical Standard Finite Normal Mixture (SFNM) model, VISDA performs top-down divisive clustering as outlined in Figure 2.1. Major blocks in the flowchart are introduced in the following sub-sections. Suppose that the hierarchical exploration has already proceeded to the lth level, i.e. K_l clusters have already been detected at level l and the posterior probability of data point $\mathbf{t}_i$ belonging to cluster k ($k = 1, \ldots, K_l$) is $z_{i,k}$.

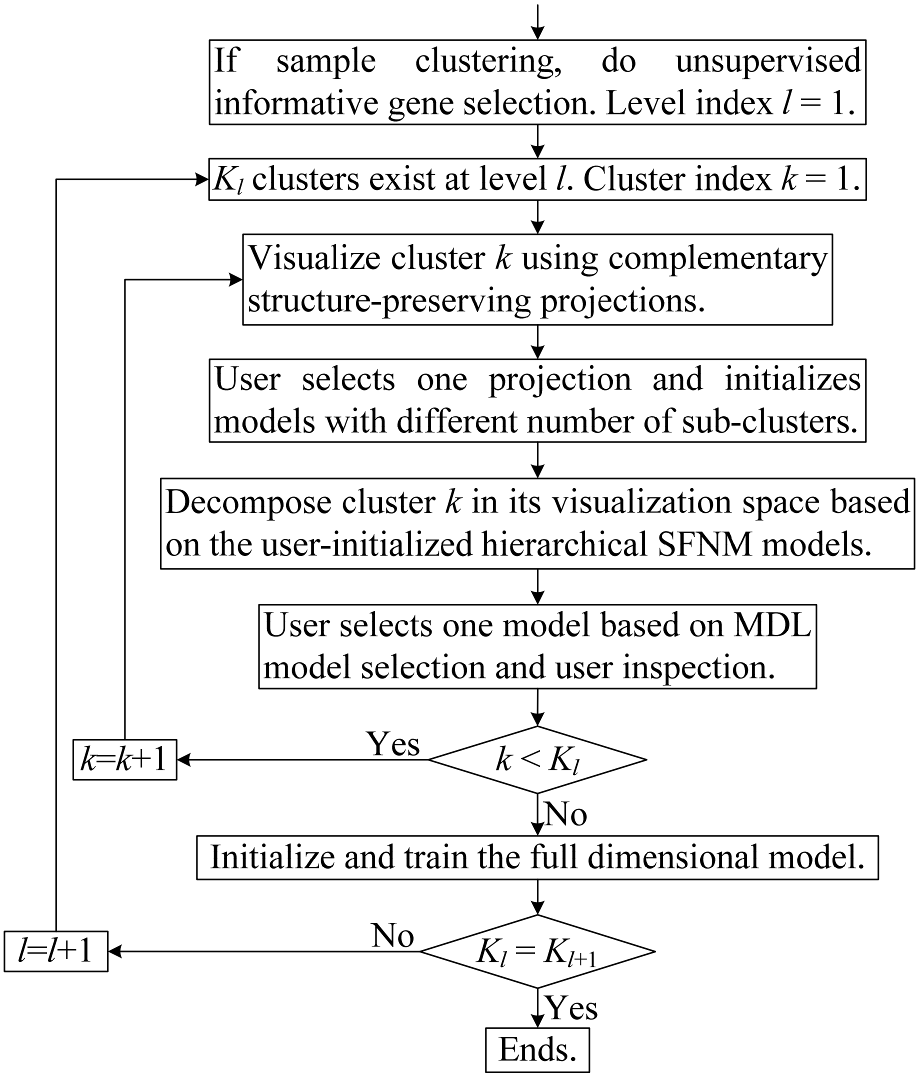

If sample clustering, do unsupervised informative gene selection. Level index $l = 1$.
K_l clusters exist at level l. Cluster index $k = 1$.
Visualize cluster k using complementary structure-preserving projections.
User selects one projection and initializes models with different number of sub-clusters.
Decompose cluster k in its visualization space based on the user-initialized hierarchical SFNM models.
User selects one model based on MDL model selection and user inspection.
$k < K_l$
Yes
$k=k+1$
No
Initialize and train the full dimensional model.
$K_l = K_{l+1}$
No
$l=l+1$
Yes
Ends.

Figure 2.1 VISDA flowchart.

2.2.1 Cluster Visualization by Complementary Structure-preserving Projections

For cluster k, VISDA projects the given cluster onto 2-D spaces by five linear projection methods preserving different data structures associated with distinct types of sub-cluster tendency.

(1) Principal Component Analysis (PCA) [40]. Sub-cluster structure is consistent with variation within the cluster. PCA performs an eigenvalue decomposition of the cluster's covariance matrix, which is calculated by

$$\boldsymbol{\Sigma}_{\mathbf{t},k} = \frac{\sum_{i=1}^{N} z_{i,k}\left(\mathbf{t}_i - \boldsymbol{\mu}_{\mathbf{t},k}\right)\left(\mathbf{t}_i - \boldsymbol{\mu}_{\mathbf{t},k}\right)^T}{\sum_{i=1}^{N} z_{i,k}}, \tag{2.1}$$

where $\boldsymbol{\mu}_{\mathbf{t},k} = \sum_{i=1}^{N} z_{i,k}\mathbf{t}_i \Big/ \sum_{i=1}^{N} z_{i,k}$ is the mean of cluster k, and the subscript '**t**' indicates that these parameters model the data in the original data space. The PCA projection uses the eigenvectors associated with the largest two eigenvalues as the projection directions. Measured by the second order statistic, the PCA projection preserves the largest variation within the cluster.

(2) Principal Component Analysis – Projection Pursuit Method (PCA–PPM) [66]. Although sub-cluster structure will surely impart variation within the cluster, the directions of largest variation may not optimally reveal sub-cluster structure [21]. Projection pursuit calculates the kurtosis of the projected data distribution on each of the eigenvectors obtained by PCA. If kurtosis is large, the projected data distribution presents a single sharp peak, which indicates no sub-cluster structure [21]. PCA–PPM selects the two eigenvectors whose associated kurtoses are smallest as projection directions. For any projection vector **w**, the kurtosis of the projected data distribution is calculated by

$$kurtosis = \frac{\dfrac{\sum_{i=1}^{N} z_{i,k}\lambda_i^4}{\sum_{i=1}^{N} z_{i,k}}}{\left(\dfrac{\sum_{i=1}^{N} z_{i,k}\lambda_i^2}{\sum_{i=1}^{N} z_{i,k}}\right)^2}, \tag{2.2}$$

where $\lambda_i = \mathbf{w}^T \times \left(\mathbf{t}_i - \boldsymbol{\mu}_{\mathbf{t},k}\right)$ is the image of $\mathbf{t}_i$ after projection.

(3) Locality Preserving Projection (LPP) [67]. Based on the Maximum A Posteriori probability (MAP) rule [40], data points that most likely belong to cluster k (evaluated by the data points' posterior probabilities of belonging to different clusters) are identified and used for calculating LPP. In LPP, the projection directions are obtained by minimizing a compactness cost function, which is a weighted summation of the pair-wise square distances between points in the projection space. The square distances between neighboring points are given large weights while the square distances between far apart points are given small weights. Thus the minimization emphasizes keeping the neighboring data points close in the projection space to preserve the local data structure. The minimization is achieved by the generalized eigenvalue decomposition [67]. The eigenvectors are orthogonalized by the Gram–Schmidt process [68] to form an affine projection matrix.

(4) HC–KMC–SFNM–DCA [27]. DCA refers to Discriminatory Component Analysis, a supervised mode projection (dimension reduction) method aiming to preserve as much as possible the discrimination/separability between known classes [66]. The main idea behind HC–KMC–SFNM–DCA is to use an unsupervised clustering method to obtain a partition of the data and then use DCA as a visual validation of partition separability. If a partition of the data is indeed consistent with the sub-cluster structure, the consequent DCA projection will show distinct sub-clusters. The first three steps in HC–KMC–SFNM–DCA sequentially execute HC, KMC and SFNM fitting, where the clustering result of the previous step is used to initialize the subsequent step. Based on the MAP rule, data points that most likely belong to cluster *k* are identified, and clustered using HC. Then, user chooses a distance threshold to cut the cluster into sub-clusters on the dendrogram produced by HC. Very small sub-clusters are merged with their closest, relatively big sub-cluster, because they are probably caused by noise and outliers in data. Thus the sub-clusters are initialized and the sub-cluster number is determined. After partitioning the data via the whole procedure of HC–KMC–SFNM, the obtained sub-clusters are taken as known classes and DCA is used to present the separability among them. DCA here maximizes weighted Fisher criterion [69], which is a modified version of Fisher criterion that is the trace of the multiplication of the inversed within-class scatter matrix and the between-class scatter matrix in the projection space [40]. Compared to Fisher criterion, weighted Fisher criterion weights the class pairs in the between-class scatter matrix, thus confines the influence of class pairs that are well-separated, and emphasizes the class pairs that are overlapped so as to improve the overall separation of classes [69]. Weighted Fisher criterion is calculated by

$$\text{Weighted Fisher Criterion} = \left(\mathbf{S}_{\mathrm{p,w,w}}\right)^{-1}\mathbf{S}_{\mathrm{p,w,b}} = \left(\mathbf{U}^{T}\mathbf{S}_{\mathbf{t},\mathrm{w,w}}\mathbf{U}\right)^{-1}\mathbf{U}^{T}\mathbf{S}_{\mathbf{t},\mathrm{w,b}}\mathbf{U}\,, \tag{2.3}$$

where **U** is the projection matrix from the original data space to the 2-D projection space, the subscript ‘p’ indicates that the parameters model the data in the projection space, the second subscript ‘w’ in $\mathbf{S}_{\text{p,w,w}}$, $\mathbf{S}_{\text{p,w,b}}$, $\mathbf{S}_{\mathbf{t},\text{w,w}}$, and $\mathbf{S}_{\mathbf{t},\text{w,b}}$ indicates that these scatter matrices are used in weighted Fisher criterion, $\mathbf{S}_{\text{p,w,w}}$ and $\mathbf{S}_{\text{p,w,b}}$ are the within-sub-cluster scatter matrix and between-sub-cluster scatter matrix in the projection space, respectively, and $\mathbf{S}_{\mathbf{t},\text{w,w}}$ and $\mathbf{S}_{\mathbf{t},\text{w,b}}$ are the within-sub-cluster scatter matrix and between-sub-cluster scatter matrix in the original data space, respectively. $\mathbf{S}_{\mathbf{t},\text{w,w}}$ and $\mathbf{S}_{\mathbf{t},\text{w,b}}$ are calculated by

$$\begin{aligned}
\mathbf{S}_{\mathbf{t},\text{w,w}} &= \sum_{\upsilon=1}^{K_{\text{sc}}} p_\upsilon \mathbf{\Sigma}_{\mathbf{t},\upsilon} \\
\mathbf{S}_{\mathbf{t},\text{w,b}} &= \sum_{\upsilon=1}^{K_{\text{sc}}-1} \sum_{\varphi=\upsilon+1}^{K_{\text{sc}}} p_\upsilon p_\varphi \omega\left(\Delta_{\upsilon\varphi}\right)\left(\boldsymbol{\mu}_{\mathbf{t},\upsilon} - \boldsymbol{\mu}_{\mathbf{t},\varphi}\right)\left(\boldsymbol{\mu}_{\mathbf{t},\upsilon} - \boldsymbol{\mu}_{\mathbf{t},\varphi}\right)^T \\
\omega\left(\Delta_{\upsilon\varphi}\right) &= \frac{1}{2\Delta_{\upsilon\varphi}^2}\operatorname{erf}\left(\frac{\Delta_{\upsilon\varphi}}{2\sqrt{2}}\right) \\
\Delta_{\upsilon\varphi} &= \sqrt{\left(\boldsymbol{\mu}_{\mathbf{t},\upsilon} - \boldsymbol{\mu}_{\mathbf{t},\varphi}\right)^T \mathbf{S}_{\mathbf{t},\text{w,w}}^{-1}\left(\boldsymbol{\mu}_{\mathbf{t},\upsilon} - \boldsymbol{\mu}_{\mathbf{t},\varphi}\right)}
\end{aligned}, \qquad (2.4)$$

where K_{sc} is the number of sub-clusters obtained by the HC-KMC-SFNM procedure, p_υ is the sample proportion of sub-cluster υ, $\mathbf{\Sigma}_{\mathbf{t},\upsilon}$ is the covariance matrix of sub-cluster υ in the original data space, $\boldsymbol{\mu}_{\mathbf{t},\upsilon}$ is the mean of sub-cluster υ in the original data space, and erf(•) is the Gaussian error function. Maximization of weighted Fisher criterion is achieved by eigenvalue decomposition of weighted Fisher scatter matrix, i.e. $(\mathbf{S}_{\mathbf{t},\text{w,w}})^{-1}\mathbf{S}_{\mathbf{t},\text{w,b}}$ [69]. The two eigenvectors associated with the largest two eigenvalues of weighted Fisher scatter matrix are selected to form the projection matrix. To achieve an affine projection, these two eigenvectors are further orthogonalized by the Gram–Schmidt process [68].

(5) Affinity Propagation Clustering – Discriminatory Component Analysis (APC–DCA) [40, 70]. Similar to HC–KMC–SFNM–DCA, APC–DCA follows the idea of using DCA to evaluate/confirm partitions learned in an unsupervised manner, but based on a different clustering procedure, namely APC. Again, only data points that most likely belong to cluster *k* (evaluated by their posterior probabilities of belonging to different clusters) are input for clustering. By viewing each data point as a node in a network, APC recursively transmits along edges of the network two kinds of real-valued messages, each of which takes into account a different kind of competition, until a good set of sub-cluster exemplars and corresponding sub-clusters emerge [70]. The "responsibility" message res($\mathbf{t}_\alpha$, $\mathbf{t}_\beta$) reflects the accumulated evidence for how well-suited candidate exemplar point $\mathbf{t}_\beta$ is to serve as the exemplar for data point $\mathbf{t}_\alpha$, taking into account other potential exemplars for $\mathbf{t}_\alpha$; and the "availability" message ava($\mathbf{t}_\alpha$, $\mathbf{t}_\beta$) reflects the accumulated evidence for how appropriate it would be for data point $\mathbf{t}_\alpha$ to choose candidate exemplar point $\mathbf{t}_\beta$ as its exemplar, taking into account the support from other points that $\mathbf{t}_\beta$ should be an exemplar [70]. The algorithm iteratively updates responsibilities and availabilities until convergence to search for maxima of the cost function that is the sum of similarities between data points and their sub-cluster exemplar. In each iteration, the algorithm first updates all responsibilities given the current availabilities according to

$$\operatorname{res}\left(\mathbf{t}_\alpha,\mathbf{t}_\beta\right)=\operatorname{sim}\left(\mathbf{t}_\alpha,\mathbf{t}_\beta\right)-\max_{\mathbf{t}_\gamma\text{ s.t. }\mathbf{t}_\gamma\in\Omega,\gamma\neq\beta}\left\{\operatorname{ava}\left(\mathbf{t}_\alpha,\mathbf{t}_\gamma\right)+\operatorname{sim}\left(\mathbf{t}_\alpha,\mathbf{t}_\gamma\right)\right\}, \tag{2.5}$$

where sim($\mathbf{t}_\alpha$, $\mathbf{t}_\beta$) is the similarity between $\mathbf{t}_\alpha$ and $\mathbf{t}_\beta$, which can be $-\|\mathbf{t}_\alpha - \mathbf{t}_\beta\|^2$, and Ω is the set including all data points involved in APC clustering. The update of responsibilities lets all candidate exemplars compete for the ownerships of data points. Then the algorithm updates all availabilities given the current responsibilities according to

$$\operatorname{ava}\left(\mathbf{t}_\alpha,\mathbf{t}_\beta\right)=\begin{cases}\min\left\{0,\operatorname{res}\left(\mathbf{t}_\beta,\mathbf{t}_\beta\right)+\sum\limits_{\mathbf{t}_\gamma\text{ s.t. }\mathbf{t}_\gamma\in\Omega,\gamma\notin\{\alpha,\beta\}}\max\left\{0,\operatorname{res}\left(\mathbf{t}_\gamma,\mathbf{t}_\beta\right)\right\}\right\}, & \text{for }\alpha\neq\beta\\ \sum\limits_{\mathbf{t}_\gamma\text{ s.t. }\mathbf{t}_\gamma\in\Omega,\gamma\neq\beta}\max\left\{0,\operatorname{res}\left(\mathbf{t}_\gamma,\mathbf{t}_\beta\right)\right\}, & \text{o.w.}\end{cases}. \tag{2.6}$$

The update of availabilities gathers evidence from data points as to whether each candidate exemplar would be a good exemplar. The third step in the iteration combines availabilities and responsibilities to determine the exemplars and corresponding sub-clusters according to the rule that for data point $\mathbf{t}_\alpha$, the data point $\mathbf{t}_\beta \in \Omega$ that maximizes ava($\mathbf{t}_\alpha$, $\mathbf{t}_\beta$) + res($\mathbf{t}_\alpha$, $\mathbf{t}_\beta$) is the exemplar of $\mathbf{t}_\alpha$. If the exemplar decisions do not change for a number of iterations, the algorithm terminates. In practical applications, certain damping strategies are taken to modify the update rules presented in Equation (2.5) and Equation (2.6) to avoid numerical oscillations [70]. It was shown that the affinity propagation method finds the best solution amongst those in a particularly large proximal region in the solution space [70, 71].

In each of the aforementioned five projection methods, after the projection matrix

$\mathbf{W}_k$ for cluster k is determined, the data projection is achieved by

$$\mathbf{y}_i = \mathbf{W}_k^T \left(\mathbf{t}_i - \boldsymbol{\mu}_{\mathbf{t},k} \right), \tag{2.7}$$

where $\mathbf{y}_i$ is the image of data point i in the projection space. Each of all the data points is displayed with an intensity proportional to the posterior probability $z_{i,k}$ (or we can set a threshold, and only points with a posterior probability bigger than this threshold are displayed). Available prior/domain information about the data is also provided to the user via additional user interface. For gene clustering, prior information can be gene annotations, such as gene ID and the functional category. For sample clustering, prior information can be array annotations, such as the experimental condition under which the array was generated.

Each of these five projection methods preserves different yet complementary data structure associated with a distinct type of sub-cluster tendency. PCA preserves directions with largest variation in the data. PCA-PPM moderates PCA to consider projection directions on which the projected data have flat distributions or distributions with thick tails. LPP preserves the neighborhood structure of the data. HC-KMC-SFNM-DCA and APC-DCA directly target presenting discrimination among sub-clusters via different unsupervised partition approaches. HC-KMC-SFNM partitioning is model-based and allows the user to determine the sub-cluster number, while APC partitioning is nonparametric and automatically determines the sub-cluster number. Because each projection method has its own, distinct theoretical or experimental assumption of data structure associated with sub-clusters, while whether the underlying sub-clusters of interest are consistent with these assumptions is data/application dependent, using all of them simultaneously increases the likelihood of revealing sub-clusters of interest.

After inspecting all five projections, user is asked to select one projection that best reveals the sub-cluster structure as the final visualization. Human interaction in choosing the best projection (and hence substructure) provides an interface to incorporate human discernment and domain knowledge in cluster discovery, which gives potential to avoid confounding, irrelevant, and uninteresting substructures. Nevertheless, the selection of a suitable/good projection is data/application dependent. Several guidelines based on human discernment and prior knowledge are as follows. (1) Select a projection in which the sub-clusters are well-separated and show clear sub-cluster structure. (2) Select a projection in which no sub-clusters are simply composed of several outliers. (3) Select a projection that does not oppose prior knowledge. For example, in gene clustering, if several genes have been well studied in previous researches and are known to be co-regulated and co-expressed under the particular experimental condition that produces the data, a projection that has these genes closely located is more preferred than a projection that has these genes located far apart. More details, discussion, and empirical understanding of these projections can be found in [64].

2.2.2 Cluster Decomposition Based on Hierarchical SFNM Model

We use a two-level hierarchical SFNM model to describe the relationship between the *l*th and the *l*+1th levels of VISDA's hierarchical exploration. The probability density function for a two-level hierarchical SFNM model is formulated as:

$$\begin{array}{ll} \multicolumn{2}{c}{\mathrm{f}\left(\mathbf{t}_i \middle| \boldsymbol{\theta}_{\mathbf{t}}, \boldsymbol{\pi}\right) = \sum_{k=1}^{K_l} \pi_k \sum_{j=1}^{K_{k,l+1}} \pi_{j|k}\, \mathrm{g}\left(\mathbf{t}_i \middle| \boldsymbol{\theta}_{\mathbf{t},(k,j)}\right)} \\ \sum_{k=1}^{K_l} \pi_k = 1 & \sum_{j=1}^{K_{k,l+1}} \pi_{j|k} = 1 \end{array}, \tag{2.8}$$

where $K_{k,l+1}$ sub-clusters exist at level *l*+1 for each cluster *k* at level *l*, π_k is the mixing proportion for cluster *k* at level *l*, $\pi_{j|k}$ is the mixing proportion for sub-cluster *j* within cluster *k*, g(•) is the Gaussian probability density function, and $\boldsymbol{\theta}_{\mathbf{t},(k,j)}$ are the associated parameters of sub-cluster *j*. When the cluster labels of the samples at level *l* are known, the conditional probability density function is formulated as

$$\mathrm{f}\left(\mathbf{t}_i \middle| \boldsymbol{\theta}_{\mathbf{t}}, \boldsymbol{\pi}, \boldsymbol{\xi}_i\right) = \prod_{k=1}^{K_l} \left(\sum_{j=1}^{K_{k,l+1}} \pi_{j|k}\, \mathrm{g}\left(\mathbf{t}_i \middle| \boldsymbol{\theta}_{\mathbf{t},(k,j)}\right) \right)^{\xi_{i,k}} \tag{2.9}$$

where $\xi_{i,k}$ is a binary cluster label indicator of sample *i*, i.e. one of $\{\xi_{i,1}, \ldots, \xi_{i,K_l}\}$ is 1 and the others are all zeros, and $\xi_{i,k} = 1$ means that cluster *k* generates $\mathbf{t}_i$. However, we actually only have partial, probabilistic information in the form of the posterior probability $z_{i,k}$ for cluster *k* having generated $\mathbf{t}_i$, instead of binary-valued $\xi_{i,k}$. We focus on cluster *k* and obtain the expectation of the conditional log-likelihood of cluster *k* as

$$\mathrm{L}(\mathbf{t} | \boldsymbol{\theta}_{\mathbf{t},k}, \boldsymbol{\pi}_k, \mathbf{z}_k) = \sum_{i=1}^{N} z_{i,k} \ln\left(\sum_{j=1}^{K_{k,l+1}} \pi_{j|k}\, \mathrm{g}\left(\mathbf{t}_i \middle| \boldsymbol{\theta}_{\mathbf{t},(k,j)}\right) \right). \tag{2.10}$$

According to the projection invariant property of normal distributions [72], i.e. a Gaussian distribution is still a Gaussian distribution after a linear projection, the projected data have an expectation of conditional log-likelihood given by

$$\mathrm{L}\left(\mathbf{y}\middle|\boldsymbol{\theta}_{\mathbf{y},k},\boldsymbol{\pi}_k,\mathbf{z}_k\right)=\sum_{i=1}^{N} z_{i,k}\ln\left(\sum_{j=1}^{K_{k,l+1}}\pi_{j|k}\,\mathrm{g}\left(\mathbf{y}_i\middle|\boldsymbol{\theta}_{\mathbf{y},(k,j)}\right)\right), \tag{2.11}$$

where $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N \mid \mathbf{y}_i \in \mathrm{R}^2, i = 1, 2, \ldots, N\}$ indicates all the projected data points in the visualization space, the subscript '**y**' indicates that these parameters model the data in the visualization space, and $\boldsymbol{\theta}_{\mathbf{y},(k,j)}$ are the associated parameters of sub-cluster j in the visualization space. Equation (2.11) is a weighted log-likelihood, which can be maximized or locally maximized by the Expectation Maximization (EM) algorithm [73]. The EM algorithm performs the E-step and M-step iteratively until convergence. The E-step and M-step for training the above hierarchical SFNM model are given by

$$\begin{aligned}
&\text{E Step:}\quad z_{i,(k,j)} = z_{i,k}\frac{\pi_{j|k}\,\mathrm{g}(\mathbf{y}_i|\boldsymbol{\mu}_{\mathbf{y},(k,j)},\boldsymbol{\Sigma}_{\mathbf{y},(k,j)})}{\sum_{\zeta=1}^{K_{k,l+1}}\pi_{\zeta|k}\,\mathrm{g}(\mathbf{y}_i|\boldsymbol{\mu}_{\mathbf{y},(k,\zeta)},\boldsymbol{\Sigma}_{\mathbf{y},(k,\zeta)})}\\
&\text{M Step:}\quad \pi_{j|k} = \frac{\sum_{i=1}^{N} z_{i,(k,j)}}{\sum_{i=1}^{N} z_{i,k}}\\
&\qquad\boldsymbol{\mu}_{\mathbf{y},(k,j)} = \frac{\sum_{i=1}^{N} z_{i,(k,j)}\mathbf{y}_i}{\sum_{i=1}^{N} z_{i,(k,j)}}\\
&\qquad\boldsymbol{\Sigma}_{\mathbf{y},(k,j)} = \frac{\sum_{i=1}^{N} z_{i,(k,j)}\left(\mathbf{y}_i-\boldsymbol{\mu}_{\mathbf{y},(k,j)}\right)\left(\mathbf{y}_i-\boldsymbol{\mu}_{\mathbf{y},(k,j)}\right)^T}{\sum_{i=1}^{N} z_{i,(k,j)}}
\end{aligned}\quad, \tag{2.12}$$

where $z_{i,(k,j)}$ is the posterior probability of data point $\mathbf{y}_i$ belonging to the jth sub-cluster in cluster k, $\boldsymbol{\mu}_{\mathbf{y},(k,j)}$ and $\boldsymbol{\Sigma}_{\mathbf{y},(k,j)}$ are the mean and covariance matrix of sub-cluster j in cluster k. This training process decomposes cluster k by keeping the data point's posterior probability of belonging to cluster k unchanged and adjusting its conditional posterior probabilities of belonging to the lower level sub-clusters.

To get an accurate and biologically meaningful initialization of the model parameters, which is an effective strategy to obtain a robust yet meaningful clustering result, VISDA exploits a user-guided initialization of sub-cluster means in the visualization space. The user pinpoints on the visualization screen where he/she thinks the sub-cluster centers should be, according to his/her discernment of the sub-cluster structure and domain knowledge. This initialization can potentially avoid learning uninteresting or biologically irrelevant substructures. For example, if a sub-cluster has several outliers, the user will most likely initialize the sub-cluster mean on the "main body" of the sub-cluster but not on the outliers.

Models with different numbers of sub-clusters are initialized by the user and trained by the EM algorithm. The obtained partitions of all the models are displayed to the user as a reference for model selection. The MDL criterion is also utilized as a theoretical validation for model order selection [44, 45]. When the data size is small, the classical MDL model order selection on Gaussian distributions has the tendency to select complex models in low-dimensional space [74]. Based on our experimental experience and reference to [74, 75], we use a modified formula to calculate the description length given by

$$-\mathrm{L}(\mathbf{y}|\boldsymbol{\theta}_{\mathrm{y},k}, \boldsymbol{\pi}_k, \mathbf{z}_k) + \frac{K_{a,k} \times N_k \times \ln\left(N_k\right)}{2\left(N_k - K_{a,k}\right)}, \tag{2.13}$$

where $K_{a,k} = 6K_{k,l+1} - 1$ is the number of freely adjustable parameters, $N_k = \sum_{i=1}^{N} z_{i,k}$ is the effective number of data points in cluster k, and $\mathrm{L}(\mathbf{y}|\boldsymbol{\theta}_{\mathrm{y},k}, \boldsymbol{\pi}_k, \mathbf{z}_k)$ is the log-likelihood given in Equation (2.11). This modified MDL formula not only eases the trend to overestimate the number of sub-clusters when data size is small, but also is asymptotically consistent with classical MDL formula, which is

$$-\mathrm{L}\left(\mathbf{y}|\boldsymbol{\theta}_{\mathrm{y},k}, \boldsymbol{\pi}_k, \mathbf{z}_k\right) + \frac{K_{a,k}}{2}\ln\left(N_k\right). \tag{2.14}$$

According to the MDL criterion, the optimum model is the one with minimum description length. To fully take advantage of human intelligence and domain knowledge, the user is also allowed to override the MDL model selection by specifying a sub-cluster number based on his/her inspection of all the partitions resulting from models with different number of sub-clusters. Again, this offers an opportunity to incorporate human intelligence and domain knowledge in the clustering process. For example, the user can refuse a model with sub-clusters formed by a few outliers.

2.2.3 Initialization and Training of Full Dimensional Model

Each sub-cluster in the chosen model corresponds to a new cluster at the l+1th level of the exploration hierarchy. The parameters of the sub-clusters in all the selected models of the lth level are transformed from the visualization spaces back to the original data space. This transform is achieved by

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{t},(k,j)} &= \mathbf{W}_k \boldsymbol{\mu}_{\mathbf{y},(k,j)} + \boldsymbol{\mu}_{\mathbf{t},k} \\ \boldsymbol{\Sigma}_{\mathbf{t},(k,j)} &= \mathbf{W}_k \boldsymbol{\Sigma}_{\mathbf{y},(k,j)} \mathbf{W}_k^T \end{aligned} , \tag{2.15}$$

where $\boldsymbol{\mu}_{\mathbf{t},(k,j)}$ and $\boldsymbol{\Sigma}_{\mathbf{t},(k,j)}$ are the mean and covariance matrix for the jth sub-cluster within cluster k in the original data space. Obviously, these transformed parameters may not accurately describe the full dimensional data distribution. From Equation (2.8), we can see that the two-level hierarchical SFNM model can be written in the form of a standard one-level SFNM model simply by putting π_k inside the second summation, giving a mixture with components indexed by (k, j) and mass $\pi_k\pi_{j|k}$. Let $\pi_{(k,j)}$ = $\pi_k\pi_{j|k}$. The probability density function of the one-level SFNM model in the original data space is

$$\mathrm{f}\left(\mathbf{t}_i \middle| \boldsymbol{\theta}_{\mathbf{t}}, \boldsymbol{\pi}\right) = \sum_{k=1}^{K_l} \sum_{j=1}^{K_{k,l+1}} \pi_{(k,j)} \,\mathrm{g}\left(\mathbf{t}_i \middle| \boldsymbol{\theta}_{\mathbf{t},(k,j)}\right) \quad \text{and} \quad \sum_{k=1}^{K_l} \sum_{j=1}^{K_{k,l+1}} \pi_{(k,j)} = 1 . \tag{2.16}$$

Thus we can use the transformed parameters as initialization for the SFNM model and then further refine the parameters using the EM algorithm, given by

$$
\begin{aligned}
&\text{E Step:} \quad z_{i,(k,j)} = \frac{\pi_{(k,j)}\,\mathrm{g}\left(\mathbf{t}_i \middle| \boldsymbol{\mu}_{\mathbf{t},(k,j)}, \boldsymbol{\Sigma}_{\mathbf{t},(k,j)}\right)}{\sum_{c=1}^{K_l}\sum_{b=1}^{K_{k,l+1}} \pi_{(c,b)}\,\mathrm{g}\left(\mathbf{t}_i \middle| \boldsymbol{\mu}_{\mathbf{t},(c,b)}, \boldsymbol{\Sigma}_{\mathbf{t},(c,b)}\right)} \\
&\text{M Step:} \quad \pi_{(k,j)} = \frac{\sum_{i=1}^{N} z_{i,(k,j)}}{N} \\
&\qquad \boldsymbol{\mu}_{\mathbf{t},(k,j)} = \frac{\sum_{i=1}^{N} z_{i,(k,j)}\mathbf{t}_i}{\sum_{i=1}^{N} z_{i,(k,j)}} \qquad . \\
&\qquad \boldsymbol{\Sigma}_{\mathbf{t},(k,j)} = \frac{\sum_{i=1}^{N} z_{i,(k,j)}\left(\mathbf{t}_i - \boldsymbol{\mu}_{\mathbf{t},(k,j)}\right)\left(\mathbf{t}_i - \boldsymbol{\mu}_{\mathbf{t},(k,j)}\right)^T}{\sum_{i=1}^{N} z_{i,(k,j)}}
\end{aligned}
\tag{2.17}
$$

When the training finishes, the l+1th level in the exploration hierarchy is generated. All sub-clusters at level l become clusters at level l+1 and each sample obtains its posterior probabilities of belonging to different clusters at level l+1. If no new cluster is detected at level l+1 compared to level l, or if the user believes all interesting cluster structure has been detected, the algorithm ends.

2.3 Algorithm Extension for Sample Clustering

The main clustering and visualization algorithm introduced above is directly applicable for gene clustering, which is a “data-sufficient” case due to the large ratio of gene number to sample number. Sample clustering is usually a “data-insufficient” case that suffers from the “curse of dimensionality”, because in sample clustering the number of data points to be clustered is much smaller than data dimensionality. Many of the genes are actually irrelevant respective to the phenotypes/sub-phenotypes of interest [16, 28, 53]. Thus we propose unsupervised informative gene selection as a preprocessing step before we use the basic VISDA algorithm to cluster samples. Non-informative genes can be divided into two categories. (1) Irrelevant genes, i.e. those which do not respond to the physiological event. These genes are normally constantly expressed across different experimental conditions or time snapshots. (2) Non-discriminative genes, i.e. ones that do not contribute to sample cluster structure.

Two variation criteria, the variance and the absolute difference between the minimum and maximum gene expression values across all the samples, can be used to identify and subsequently remove constantly expressed genes. For each criterion, a rank of all the genes is obtained, with genes of large variation ranked at the top.

To identify and remove non-discriminative genes, discrimination power analysis is applied. Discrimination power analysis measures each gene's individual ability both to elicit and to discriminate sample clusters/components. For each gene, the sample components are generated by fitting a 1-D SFNM model to the gene's expression values across all samples using the EM algorithm. In the SFNM model, each Gaussian distribution represents a sample component. After obtaining the fitted SFNM models of all the genes, we remove the genes whose SFNM model has only one component, because they do not support any cluster structure. For each of the remaining genes, based on its associated SFNM model, we can write the conditional probability density function of the components generating a gene expression value v, given by

$$\mathrm{f}\left(v\middle|\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\chi}\right)=\prod_{\kappa=1}^{K_{\mathrm{com}}}\left(\pi_{\kappa}\,\mathrm{g}\left(v\middle|\boldsymbol{\theta}_{\kappa}\right)\right)^{\chi_{\kappa}}, \tag{2.18}$$

where χ_κ is a binary component label indicator (i.e. one of $\{\chi_1, \ldots, \chi_{K_{\mathrm{com}}}\}$ is 1 and the others are all zeros, and $\chi_\kappa = 1$ means that the κth component generates the expression value v), K_{com} is the number of components, g(•) is the Gaussian probability density function, π_κ and $\boldsymbol{\theta}_\kappa$ are the mixing proportion and parameters associated with component κ, respectively. From the model we can derive a classification hypothesis H(v) about the component from which v is drawn. By the MAP rule, v is classified to the component that it most likely belongs to, evaluated by its posterior probabilities of belonging to different components, which can be calculated using the parameters K_{com}, π_κ and $\boldsymbol{\theta}_\kappa$ ($\kappa = 1, \ldots, K_{\mathrm{com}}$) that are learned in the process of fitting the SFNM model to the gene's expression values. The Bayes accuracy, i.e. the probability of correctly classifying v is

$$\text{Bayes Accuracy}=\sum_{\kappa=1}^{K_{\mathrm{com}}}\pi_{\kappa}\Pr\left(\mathrm{H}\left(v\right)=\kappa\middle|\chi_{\kappa}=1\right), \tag{2.19}$$

where $\Pr(\mathrm{H}(v) = \kappa \mid \chi_\kappa = 1)$ is the probability of classifying v to component κ using the MAP rule conditional on that v is generated by component κ. This accuracy is the best we can achieve when classifying samples drawn from the Gaussian mixture and is intuitively a reasonable measure of the discrimination power of the gene. For highly discriminative genes, we anticipate the accuracy is high. Thus, a rank of genes according to their discrimination power can be constructed.

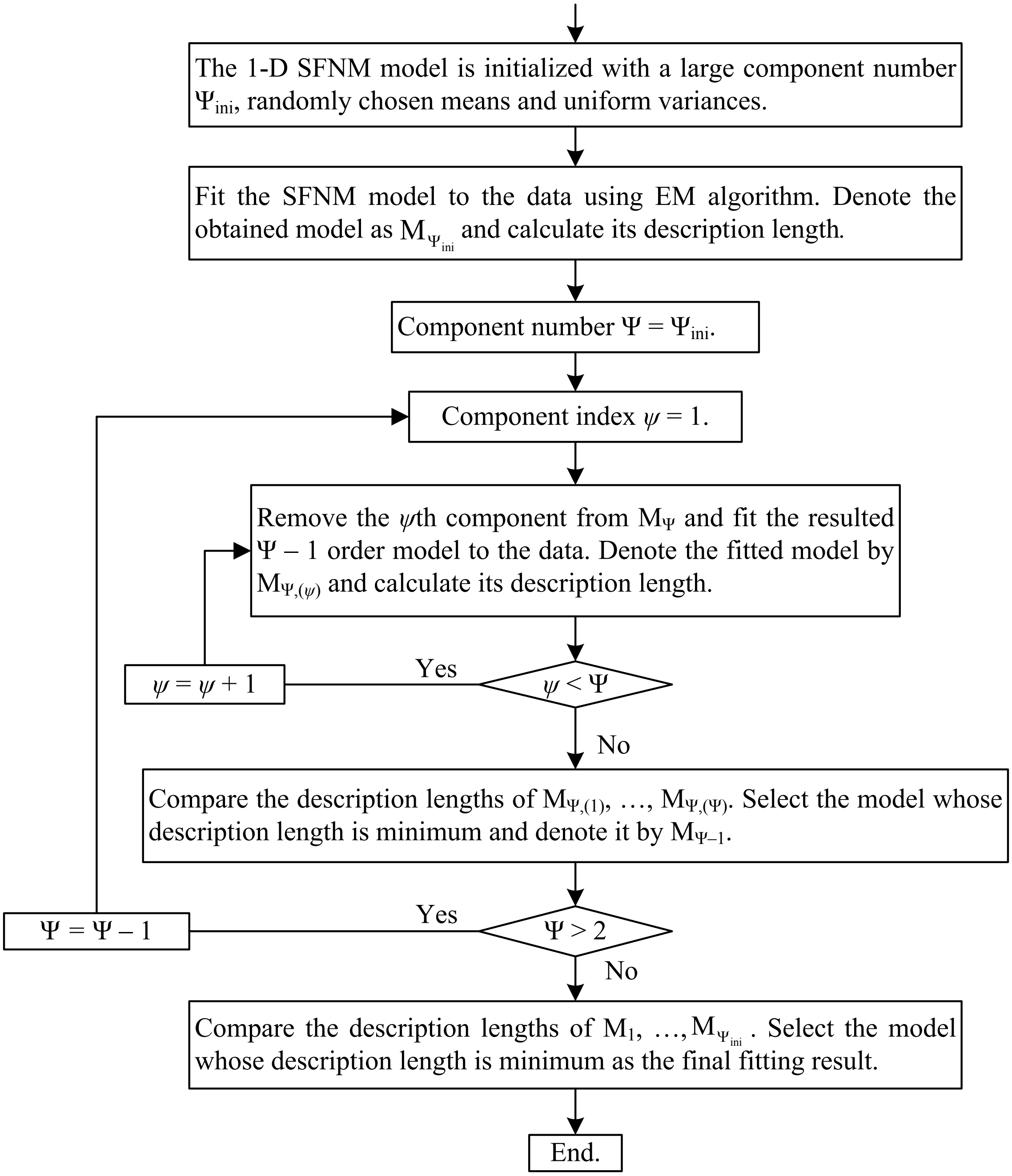

The 1-D SFNM model is initialized with a large component number Ψ_{ini}, randomly chosen means and uniform variances.
Fit the SFNM model to the data using EM algorithm. Denote the obtained model as $M_{\Psi_{ini}}$ and calculate its description length.
Component number $\Psi = \Psi_{ini}$.
Component index $\psi = 1$.
Remove the ψth component from M_{Ψ} and fit the resulted $\Psi - 1$ order model to the data. Denote the fitted model by $M_{\Psi,(\psi)}$ and calculate its description length.
Yes
$\psi = \psi + 1$
$\psi < \Psi$
No
Compare the description lengths of $M_{\Psi,(1)}$, …, $M_{\Psi,(\Psi)}$. Select the model whose description length is minimum and denote it by $M_{\Psi-1}$.
Yes
$\Psi = \Psi - 1$
$\Psi > 2$
No
Compare the description lengths of M_1, …, $M_{\Psi_{ini}}$. Select the model whose description length is minimum as the final fitting result.
End.

Figure 2.2 Flowchart of the iterative procedure for learning 1-D SFNM model.

Based on the variation ranks and the discrimination power rank, a list of genes with large variations and large discrimination power will be selected by taking the intersection of the top parts of the ranks.

In the discrimination power analysis, fitting the 1-D SFNM model to a gene's expression values follows the iterative procedure proposed in [76]. Figure 2.2 shows the iterative procedure in details. The 1-D SFNM model is initialized with a large component number (much bigger than the true component number), randomly chosen means and uniform variances. In each iteration, we (one-by-one) trial-delete each component and rerun the fitting algorithm. The component whose removal yields a model with the minimum description length will be permanently removed. Thus the component number decreases by one and a model of the current component number is obtained. This iterative process ends when only one component remains. The optimal component number and the corresponding SFNM model is determined by the MDL model selection criterion via comparing the description lengths of models with different number of components obtained in the sequence.

2.4 Algorithm Extension for Phenotype Clustering

As an extension of the basic VISDA clustering and visualization algorithm, phenotype clustering follows a similar hierarchical, interactive exploration process, shown in Figure 2.3. By exploiting the known phenotype-categorical information, modified data visualization and decomposition scheme is developed as indicated by the green blocks with dashed borders in Figure 2.3. Suppose that the exploration process has proceeded to the lth level with K_l phenotype clusters, each of which contains all the samples from one or multiple phenotypes. For phenotype cluster k (k = 1, …, K_l), if it contains only one phenotype, we do not need to decompose it; if it contains two phenotypes, we simply split the cluster into two sub-clusters, each containing the samples of one phenotype; if it contains more than two phenotypes, we perform the following steps to visualize and decompose it. Let Q_k and $N^{(q)}$ denote the number of phenotypes in cluster k and the number of samples from the qth phenotype in cluster k, respectively.

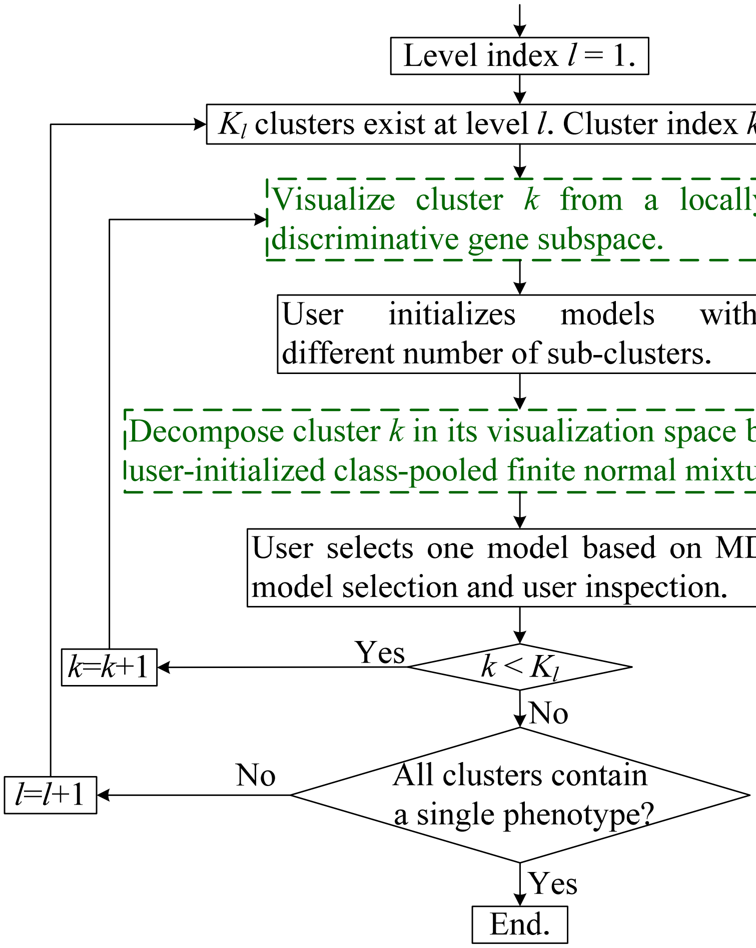

Level index l = 1.
K_l clusters exist at level l. Cluster index k = 1.
Visualize cluster k from a locally discriminative gene subspace.
User initializes models with different number of sub-clusters.
Decompose cluster k in its visualization space based on the user-initialized class-pooled finite normal mixture models.
User selects one model based on MDL model selection and user inspection.
$k < K_l$
Yes
k=k+1
No
All clusters contain a single phenotype?
No
l=l+1
Yes
End.

Figure 2.3 VISDA Flowchart including the algorithm extension for phenotype clustering. The green blocks with dashed borders indicate the algorithm extensions, i.e. the modified visualization scheme and decomposition scheme.

2.4.1 Cluster Visualization from Locally Discriminative Gene Subspace

Let $\{\mathbf{t}^{(1)}, \ldots, \mathbf{t}^{(Q_k)}\}$ denote the phenotypes in cluster k. $\mathbf{t}^{(q)} = \{\mathbf{t}_i^{(q)}, i = 1, 2, \ldots, N^{(q)}\}$ $(q = 1, \ldots, Q_k)$ is the set of samples in phenotype q. To achieve an effective visualization, we first use supervised discriminative gene selection to form a locally discriminative gene subspace respective to the phenotype categories in the cluster, and then project the samples from the discriminative gene subspace to a 2-D visualization space through DCA. The locally discriminative gene subspace contains the most discriminative genes, and the discrimination power of a gene is measured by

$$\frac{\sum_{q=1}^{Q_k-1}\sum_{m=q+1}^{Q_k} p_q p_m \left(\mu_q - \mu_m\right)^2}{\sum_{q=1}^{Q_k} p_q \sigma_q^2}, \qquad (2.20)$$

where $p_q = N^{(q)} \Big/ \sum_{q=1}^{Q_k} N^{(q)}$ is the sample proportion of phenotype q in cluster k, μ_q and σ_q are the mean and standard deviation of the gene's expression values in phenotype q. The number of genes in this gene subspace is $Q_k n_g$, where n_g is the number of selected genes per phenotype, an input parameter of the algorithm.

We use DCA to project the samples from the gene subspace onto a 2-D visualization space. Because an important outcome of phenotype clustering is the relationships among the phenotypes that are estimated directly based on the relative distances between samples of different phenotypes, to preserve the original and undistorted data structure, DCA here maximizes Fisher criterion that treats all the phenotype pairs equally. Fisher criterion is calculated by

$$\text{Fisher Criterion} = \left(\mathbf{S}_{\text{p,w}}\right)^{-1}\mathbf{S}_{\text{p,b}} = \left(\mathbf{V}^T\mathbf{S}_{\text{s,w}}\mathbf{V}\right)^{-1}\mathbf{V}^T\mathbf{S}_{\text{s,b}}\mathbf{V}, \qquad (2.21)$$

where $\mathbf{V}$ is the projection matrix from the gene subspace to the 2-D projection space, the subscript ‘p’ indicates that the parameters model the data in the projection space, $\mathbf{S}_{\text{p,w}}$ and $\mathbf{S}_{\text{p,b}}$ are the within-phenotype scatter matrix and between-phenotype scatter matrix in the projection space, respectively, the subscript ‘s’ indicates that the parameters model the data in the gene subspace, and $\mathbf{S}_{\text{s,w}}$ and $\mathbf{S}_{\text{s,b}}$ are the within-phenotype scatter matrix and between-phenotype scatter matrix in the gene subspace. $\mathbf{S}_{\text{s,w}}$ and $\mathbf{S}_{\text{s,b}}$ are calculated by

$$\begin{aligned}\mathbf{S}_{\text{s,w}} &= \sum_{q=1}^{Q_k} p_q \boldsymbol{\Sigma}_{\text{s},q} \\ \mathbf{S}_{\text{s,b}} &= \sum_{q=1}^{Q_k-1}\sum_{m=q+1}^{Q_k} p_q p_m \left(\boldsymbol{\mu}_{\text{s},q} - \boldsymbol{\mu}_{\text{s},m}\right)\left(\boldsymbol{\mu}_{\text{s},q} - \boldsymbol{\mu}_{\text{s},m}\right)^T\end{aligned}, \qquad (2.22)$$

where $\boldsymbol{\mu}_{\text{s},q}$ and $\boldsymbol{\Sigma}_{\text{s},q}$ are the mean and covariance matrix of phenotype q in the gene subspace, respectively. Maximization of Fisher criterion is achieved by eigenvalue decomposition of Fisher scatter matrix, i.e. $(\mathbf{S}_{\text{s,w}})^{-1}\mathbf{S}_{\text{s,b}}$ [40]. The two eigenvectors associated with the largest two eigenvalues of Fisher scatter matrix are selected to form the projection matrix. Because we want to keep the original data structure, these two eigenvectors are orthogonalized by the Gram–Schmidt process [68], to form an affine projection matrix. When the samples are projected onto the visualization space, prior information in the form of phenotype labels of samples are also provided to the user.

2.4.2 Cluster Decomposition Based on Class-pooled Finite Normal Mixture Model

Phenotype clustering differs from sample/gene clustering in that it assigns a cluster label to each phenotype in its entirety, not to each sample/gene. Due to this difference, we use a class-pooled finite normal mixture to model the projected samples in the visualization space. Let $\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(Q_k)}\}$ denote the projected phenotypes in cluster k, where $\mathbf{y}^{(q)}=\{\mathbf{y}_i^{(q)}, i = 1, 2, \ldots, N^{(q)}\}$ $(q = 1, \ldots, Q_k)$ is the set of samples from phenotype q. The probability density function for all projected samples from phenotype q is

$$\mathrm{f}\left(\mathbf{y}^{(q)}\middle|\boldsymbol{\theta}_{\mathbf{y},k},\boldsymbol{\pi}_k\right)=\sum_{j=1}^{K_{k,l+1}}\pi_{j|k}\prod_{i=1}^{N^{(q)}}\mathrm{g}\left(\mathbf{y}_i^{(q)}\middle|\boldsymbol{\theta}_{\mathbf{y},(k,j)}\right) \quad \text{and} \quad \sum_{j=1}^{K_{k,l+1}}\pi_{j|k}=1, \tag{2.23}$$

where cluster k at level l is decomposed into $K_{k,\,l+1}$ sub-clusters at level l+1, $\pi_{j|k}$ and $\boldsymbol{\theta}_{\mathbf{y},(k,j)}$ are the mixing proportion and parameters associated with sub-cluster j. The model parameters are learned by the EM algorithm given by

$$\text{E Step:}\quad z_{(q),(k,j)}=\frac{\pi_{j|k}\prod_{i=1}^{N^{(q)}}\mathrm{g}\left(\mathbf{y}_i^{(q)}\middle|\boldsymbol{\theta}_{\mathbf{y},(k,j)}\right)}{\sum_{\zeta=1}^{K_{k,l+1}}\pi_{\zeta|k}\prod_{i=1}^{N^{(q)}}\mathrm{g}\left(\mathbf{y}_i^{(q)}\middle|\boldsymbol{\theta}_{\mathbf{y},(k,\zeta)}\right)}$$

$$\begin{aligned}\text{M Step:}\quad \boldsymbol{\mu}_{\mathbf{y},(k,j)}&=\frac{\sum_{q=1}^{Q_k}\left(z_{(q),(k,j)}\sum_{i=1}^{N^{(q)}}\mathbf{y}_i^{(q)}\right)}{\sum_{q=1}^{Q_k}z_{(q),(k,j)}N^{(q)}}\\ \boldsymbol{\Sigma}_{\mathbf{y},(k,j)}&=\frac{\sum_{q=1}^{Q_k}z_{(q),(k,j)}\sum_{i=1}^{N^{(q)}}\left(\mathbf{y}_i^{(q)}-\boldsymbol{\mu}_{\mathbf{y},(k,j)}\right)\left(\mathbf{y}_i^{(q)}-\boldsymbol{\mu}_{\mathbf{y},(k,j)}\right)^T}{\sum_{q=1}^{Q_k}z_{(q),(k,j)}N^{(q)}}\\ \pi_{j|k}&=\frac{1}{Q_k}\sum_{q=1}^{Q_k}z_{(q),(k,j)}\end{aligned}, \tag{2.24}$$

where $z_{(q),(k,j)}$ is the posterior probability of phenotype q (all samples from phenotype q) belonging to sub-cluster j, and $\boldsymbol{\mu}_{\mathbf{y},(k,j)}$ and $\boldsymbol{\Sigma}_{\mathbf{y},(k,j)}$ are the mean and covariance matrix of sub-cluster j in the visualization space. Similar to sample/gene clustering, user is invited to initialize the sub-cluster centers by pinpointing them on the visualization screen according to his/her understanding about the data structure and domain knowledge. Models with different number of sub-clusters are initialized by the user, and trained by the EM algorithm. The resulting partitions are shown to the user for comparison. The MDL model selection criterion is also applied for theoretical validation. The description length is calculated by

$$-\mathrm{L}\left(\left\{\mathbf{y}^{(1)},\cdots,\mathbf{y}^{(Q_k)}\right\}\middle|\boldsymbol{\theta}_{\mathrm{y},k},\boldsymbol{\pi}_k\right)+\frac{K_{a,k}\times N_k\times\ln\left(N_k\right)}{2\left(N_k-K_{a,k}\right)}, \tag{2.25}$$

where $\mathrm{L}(\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(Q_k)}\}|\boldsymbol{\theta}_{\mathrm{y},k}, \boldsymbol{\pi}_k)$ is the log-likelihood, $K_{a,k}$ is the number of free adjustable parameters, and N_k is the number of samples in cluster k. They are given by

$$\begin{aligned}
&\mathrm{L}\left(\left\{\mathbf{y}^{(1)},\cdots,\mathbf{y}^{(Q_k)}\right\}\middle|\boldsymbol{\theta}_{\mathrm{y},k},\boldsymbol{\pi}_k\right)=\sum_{q=1}^{Q_k}\ln\left(\sum_{j=1}^{K_{k,l+1}}\pi_{j|k}\prod_{i=1}^{N^{(q)}}\mathrm{g}\left(\mathbf{y}_i^{(q)}\middle|\boldsymbol{\theta}_{\mathrm{y},(k,j)}\right)\right)\\
&K_{a,k}=6K_{k,l+1}-1\\
&N_k=\sum_{q=1}^{Q_k}N^{(q)}
\end{aligned}\quad. \tag{2.26}$$

The MDL model selection criterion selects the model with minimum description length as the optimum model. The user can override the MDL model selection by specifying the number of sub-clusters according to his/her justification and domain knowledge. Once the best model is selected, the phenotypes are assigned to sub-clusters using the MAP rule.

After visualizing and decomposing the clusters at level l, all the sub-clusters become clusters at level l+1. Thus the hierarchical exploration process proceeds to the l+1th level. If all the clusters at the l+1th level contain a single phenotype, the algorithm ends.

2.5 A Demo Application on Sample Clustering

To show how VISDA discovers data structure, we consider the UM microarray gene expression cancer dataset as an example and perform sample clustering [77]. This dataset consists of brain (73 samples), colon (60 samples), lung (91 samples), ovary (119 samples, 113 for ovarian cancer and 6 for uterine cancer), and pancreas (10 samples) cancer types. We removed the pancreas category due to its relatively small size. The total number of genes is 7069. We applied our unsupervised gene selection method to choose informative genes. To emphasize the genes with discrimination power, those which manifest true, underlying cluster structure, we used a more stringent requirement on a gene's discrimination power than on its variation. We took the intersection of the top 700 discriminative genes, the top 1600 genes ranked by variance, and the top 1600 genes from the absolute difference ranking. A subset of 107 genes was thus selected and used as the input gene space for sample clustering. Clustering of the 343 samples was performed in a purely unsupervised fashion, i.e. category labels and the number of categories were not used by any phase of the algorithm and were not known to the user during the clustering process. After clustering, we use colors to indicate different cancer categories, with the results shown in Figure 2.4.

Figure 2.4a shows the five different projections obtained at the top level. PCA gives roughly a three-cluster structure with dispersion of the left cluster and some overlap between the other two clusters. PCA-PPM shows a two-cluster structure with significant overlap. The HC–KMC–SFNM–DCA projection gives a well-separated two-cluster structure. LPP also produces a two-cluster structure, but not well-separated. APC–DCA gives roughly a three-cluster structure with dispersion of the right cluster and overlap between the other two clusters. For the sake of simplicity, we select data visualization based on human inspection of the separability among the clusters. Since the HC–KMC–SFNM–DCA projection presents the best separated clusters, we select it for the top level visualization and continue decomposing these clusters. Figure 2.4b shows the user's initialization and corresponding obtained partitions of models with different cluster number for the top level decomposition.

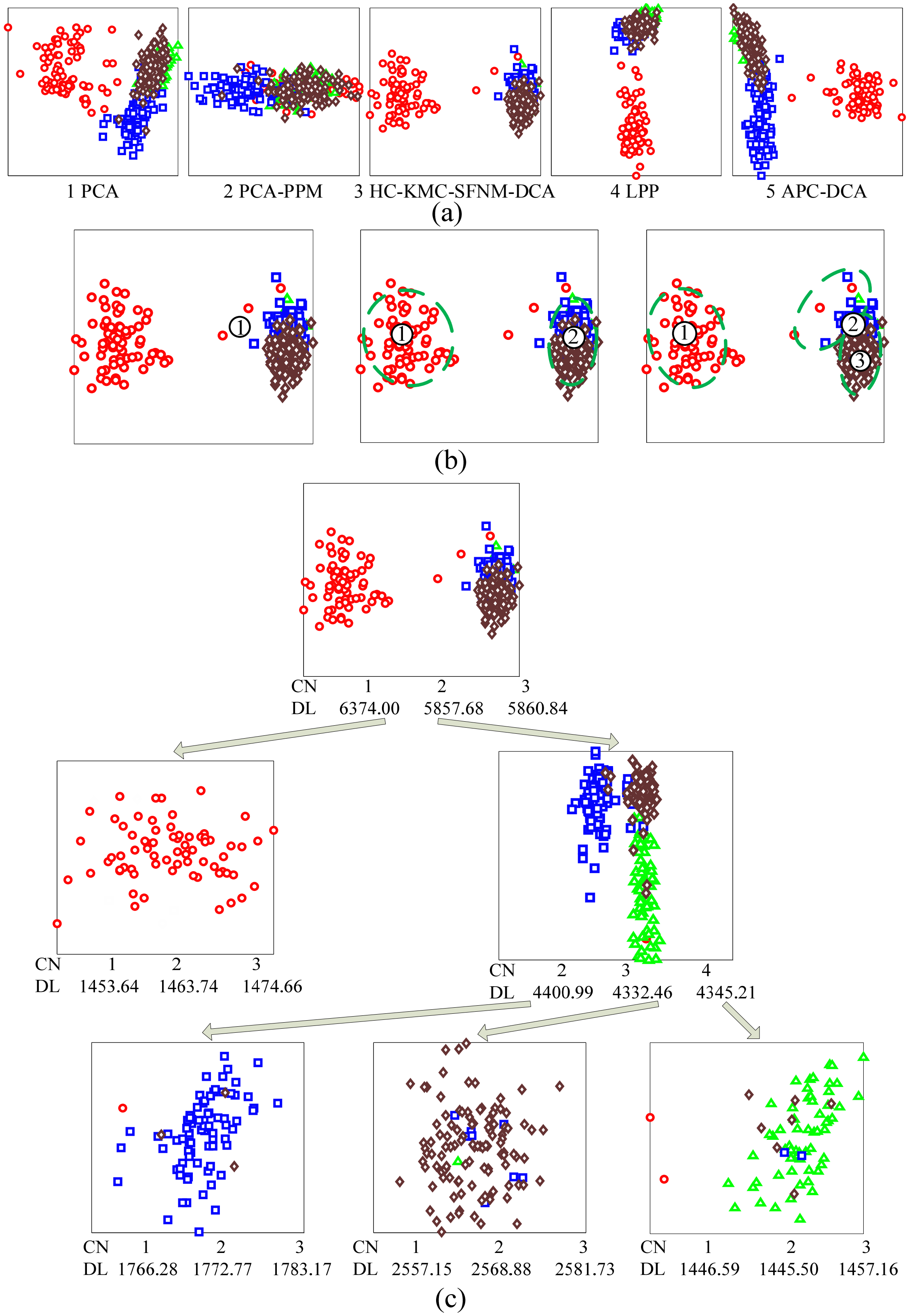


Figure 2.4 Illustration of VISDA on sample clustering. (a) The five different projections obtained at the top level. Red circles are brain cancer; green triangles are colon cancer; blue squares are lung cancer; and

brown diamonds are ovary cancer. (b) User's initialization of cluster means (indicated by the numbers in small circles) and the resulted clusters (indicated by the green dashed ellipses). The left, middle, and right figures are for the model of one cluster, two clusters, and three clusters, respectively. (c) The hierarchical data structure detected by VISDA. Sub-Cluster Number (CN) and corresponding Description Length (DL) are shown under the visualization.

Figure 2.4c shows the tree structure detected by VISDA. The clusters at the leaf nodes of the hierarchy form the final clustering result. For the colon cancer cluster (the third cluster at the third level of the hierarchy), the one sub-cluster model and the two sub-cluster model have a description length of 1446.59 and 1445.50, respectively, which are very close. By examining the two sub-cluster partition, we find that one of the sub-clusters essentially only contains the two left-most samples in the cluster, which are apparently outliers. Thus we choose not to decompose this cluster.

2.6 Identification of Gene Clusters on Muscular Dystrophy Data and Muscle Regeneration Data

2.6.1 Gene Clusters Obtained on Muscular Dystrophy Data

On a muscular dystrophy microarray gene expression dataset (Table 2.1 gives a brief description of the dataset) [26], we used VISDA to define gene clusters to discover functional gene groups and gene regulation networks. After performing the clustering, we superimposed existing knowledge of gene regulation and gene function from Ingenuity Pathway Analysis database (IPA) [78] to analyze some of the obtained gene clusters that showed interesting gene expression patterns. The IPA system provides the statistical significance of a gene cluster being associated with different genomic function categories or gene regulation networks, evaluated by *p*-values. *p*-value here is the probability that a randomly selected gene group can achieve the annotation abundance level of the gene cluster in analysis. So the smaller the *p*-value is, the more statistically significant the association between the gene cluster and the genomic function category (or gene regulation network) is. IPA calculated *p*-values based on the hypergeometric distribution via Fisher's exact test for 2×2 contingency table [78]. Suppose that we have a gene cluster containing *M* genes and *O* out of the *M* genes belong to a genomic function category (or a gene regulation network). Let C denote the total number of genes and B denote the total number of genes in the genomic function category (or gene regulation network). Apparently, $O \le M \le C$ and $O \le B \le C$. The p-value is calculated by

$$p\text{-value} = 1 - \sum_{o=0}^{O-1} \frac{\begin{pmatrix} B \\ o \end{pmatrix}\begin{pmatrix} C-B \\ M-o \end{pmatrix}}{\begin{pmatrix} C \\ M \end{pmatrix}}. \tag{2.27}$$

Table 2.1 A Brief Introduction of the Muscular Dystrophy Dataset.

Phenotype	Sample Number	Description
JDM	25	Juvenile dermatomyositis
FKRP	7	Fukutin related protein deficiency
DMD	10	Duchenne muscular dystrophy, dystrophin deficiency
BMD	5	Becker muscular dystrophy, hypomorphic for dystrophin
Dysferlin	10	Dysferlin deficiency, putative vesicle traffic defect
Calpain III	10	Calpain III deficiency
FSHD	14	Fascioscapulohumeral dystrophy
AQM	5	Acute quadriplegic myopathy
HSP	4	Spastin haploinsufficiency, microtubule traffic defect
Lamin A/C	4	Emery dreifuss muscular dystrophy, missense mutations
Emerin	4	Emery dreifuss muscular dystrophy, emerin deficient
ALS	9	Amyotrophic lateral sclerosis
NHM	18	Normal skeletal muscle

We found that one gene cluster that contained 122 genes was highly expressed in JDM but lowly expressed in all other phenotypes. JDM is a relatively severe childhood autoimmune disorder that is thought to be associated with viral infections that stimulate muscle destruction by inflammatory cells and ischemic processes in a small subset of the children with the virus. IPA showed that this gene cluster pointed to gene regulation networks involved in key inflammatory pathways. In the most significant network (with a negative log p-value of 77), shown in Figure 2.5, specific proteins that are known to be critical for initiating and perpetuating inflammation and subsequent cell death are seen as key focus genes. STAT1 is an important signaling molecule that responds to interferons and other cytokines. Both TNFSF10 and CASP7 influence cell death via apoptosis. Consistently, patients with JDM show extensive cell death and failure of regeneration in their muscle, leading to weakness. This network also points to drugs that would be expected to inhibit this process in JDM patients, which can be tested in a mouse model. IPA also shows that this gene cluster was significantly associated with organismal injury & abnormalities (negative log *p*-value > 15) and immune response (negative log *p*-value > 10) in terms of gene function category, which is consistent with our understanding about the disease mechanism.

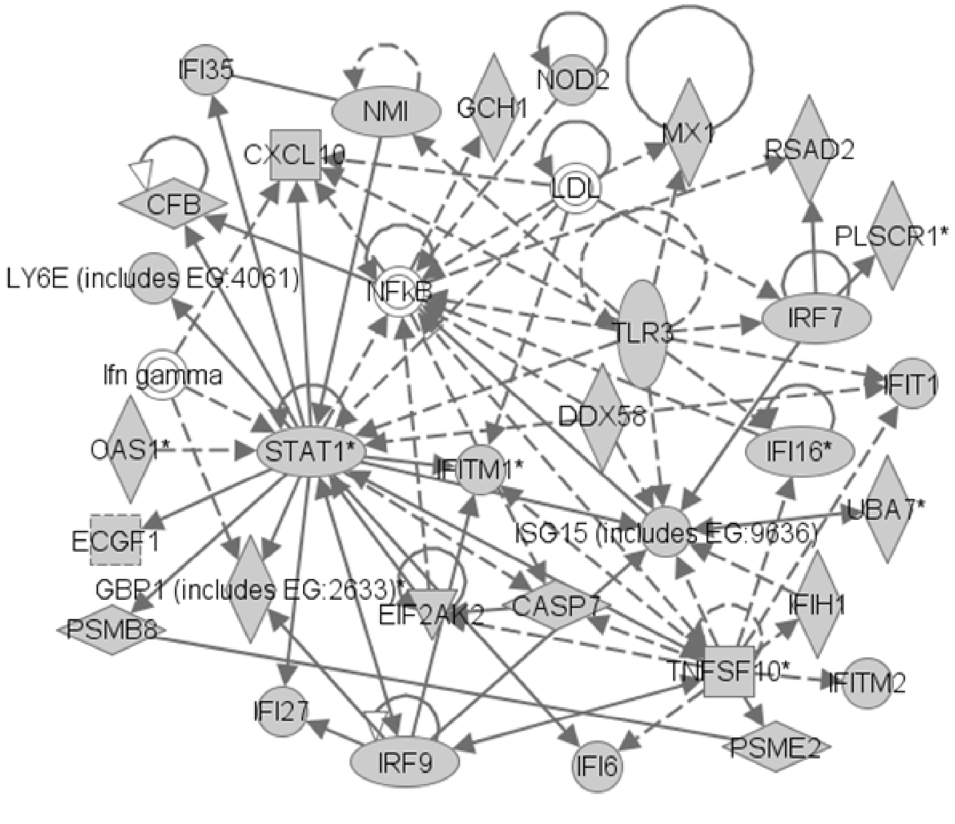


Figure 2.5 Top scoring gene regulation network indicated by the gene cluster. Grey colour indicates that the gene is in the detected gene cluster. Solid lines indicate direct interactions. Dashed lines indicate indirect interactions. This network was generated through the use of IPA (Ingenuity® Systems, www.ingenuity.com).

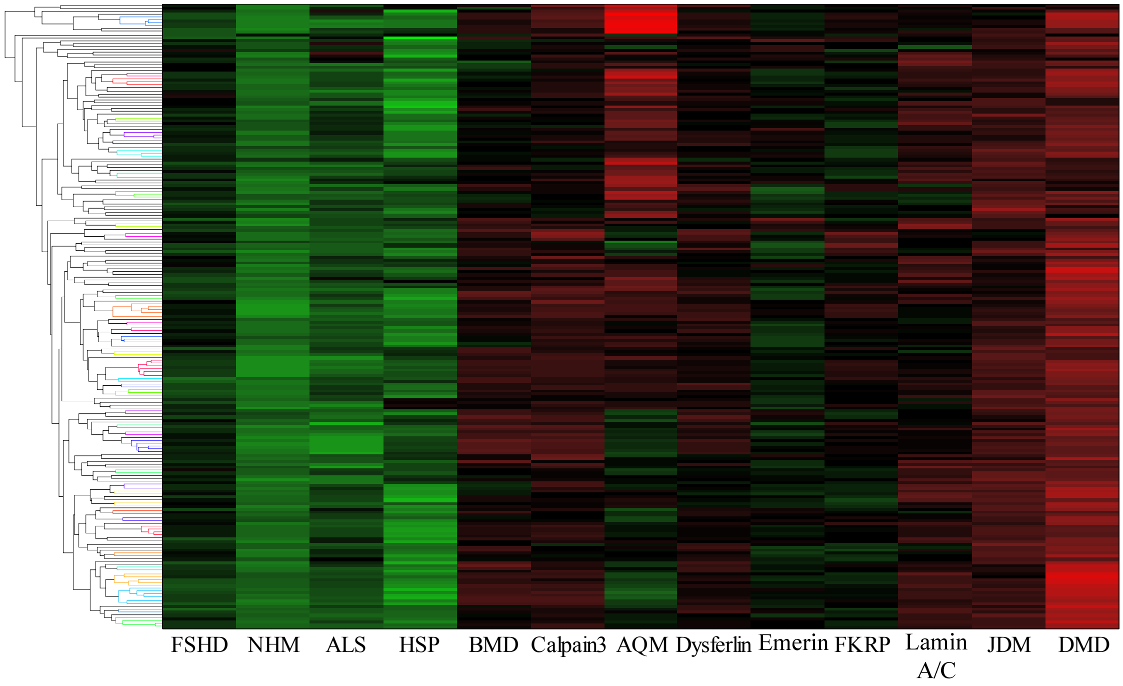


Figure 2.6 Heat map of the gene cluster that is highly expressed in DMD and JDM, but lowly expressed in FSHD, NHM and ALS. Average gene expression values of samples from the same phenotype were used to draw this figure. Each column is a phenotype. Each row is a gene. Genes are ordered according to the gene clustering result obtained by HC. Red color indicates up-regulation and green color indicates down-regulation.

Another gene cluster was highly expressed in two disorders (i.e. DMD and JDM) that show failed muscle regeneration and associated muscle wasting due to fibrosis, while lowly expressed in disorders/phenotypes (e.g. FSH, NHM, and ALS) showing little muscle degeneration/regeneration or failed muscle regeneration. Fibrosis refers to the development of excess connective tissue in an organ, such as muscle. We took the average of gene expression values of samples from the same phenotype and drew the heat map shown in Figure 2.6. Figure 2.7 shows the gene regulation network statistically significantly associated with the identified gene cluster (with a negative log *p*-value of 45). This network included TGF-β, a well studied cytokine associated with pathological fibrosis in many tissues, and also many of the known components of fibrotic tissue, such as collagen (i.e. COL6A1, COL6A2, and COL6A3) that is the main kind of proteins of connective tissue in human body.

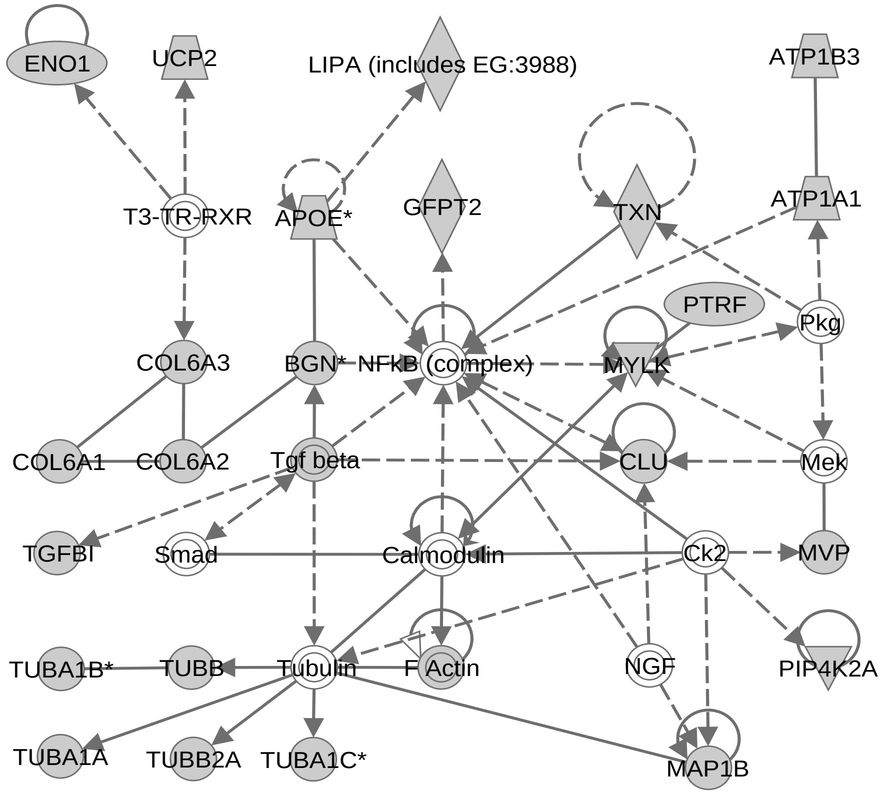


Figure 2.7 The gene regulation network statistically significantly associated with the identified gene cluster. Grey colour indicates that the gene is in the detected gene cluster. Solid lines indicate direct interactions. Dashed lines indicate indirect interactions. This network was generated through the use of IPA (Ingenuity® Systems, www.ingenuity.com).

2.6.2 Gene Clusters Obtained on Muscle Regeneration Data

We used VISDA to define gene clusters in a 27 time-point microarray gene expression dataset of muscle regeneration in vivo based on the mouse model [79]. This dataset measures the gene expression levels of mouse skeletal muscle tissue after the injection of cardiotoxin, which damages the muscle tissue and induces staged muscle regeneration. After pre-filtering by "present call", 7570 genes were believed to be significantly present and were thus input to VISDA for gene clustering. Two of the eighteen gene clusters detected by VISDA peaked at the 3rd day time point, which correlated with the expression pattern of MyoD, a prototypical member of myogenic regulatory factors that control the staged induction of genes important for interpretation of positional cues, proliferation, and differentiation of myogenic cells. These two gene clusters contained a subset of the in vitro MyoD down-stream targets identified in [80], which characterized the relevance of in vitro myogenesis to in vivo muscle regeneration.

2.7 Construction of TOPs on Muscular Dystrophy Data and Multi-category Cancer Data

2.7.1 TOP Obtained on Muscular Dystrophy Data

We used VISDA to cluster phenotypes in the muscular dystrophy dataset that includes 13 muscular dystrophy related phenotypes (Table 2.1 gives a brief introduction of the dataset) [26]. The TOP constructed by VISDA with n_g (the number of selected genes per phenotype) equal to 2 is shown in Figure 2.8. AQM, JDM, HSP, and ALS were first separated from the rest, which is consistent with each of them having an underlying disease mechanism much different from the other classes. Then, the tree showed two major branches. The left branch contained BMD, Calpain 3, DMD, Dysferlin, and FKRP, most of which are the "dystrophic myopathies", inborn single gene disorders causing degeneration/regeneration of muscle fibers. The right branch contained Lamin A/C, Emerin, FSHD, and NHM. The two nuclear envelope disorders, Lamin A/C and Emerin, form their own group, showing their close relationship reflected at mRNA profiles. FSHD disrupts chromatin attachment sites to the nuclear envelope, which supports its co-segregation with Lamin A/C and Emerin in the right branch.

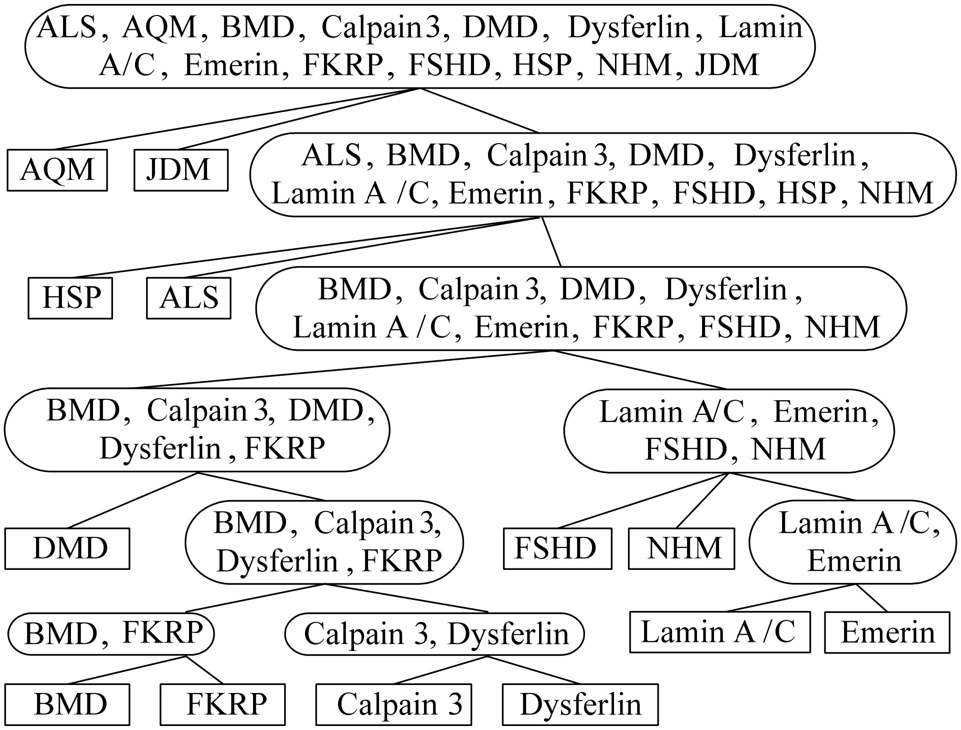

ALS, AQM, BMD, Calpain3, DMD, Dysferlin, Lamin A/C, Emerin, FKRP, FSHD, HSP, NHM, JDM
AQM
JDM
ALS, BMD, Calpain 3, DMD, Dysferlin, Lamin A /C, Emerin, FKRP, FSHD, HSP, NHM
HSP
ALS
BMD, Calpain 3, DMD, Dysferlin, Lamin A /C, Emerin, FKRP, FSHD, NHM
BMD, Calpain 3, DMD, Dysferlin, FKRP
Lamin A/C, Emerin, FSHD, NHM
DMD
BMD, Calpain 3, Dysferlin, FKRP
FSHD
NHM
Lamin A /C, Emerin
BMD, FKRP
Calpain 3, Dysferlin
Lamin A /C
Emerin
BMD
FKRP
Calpain 3
Dysferlin

Figure 2.8 TOP found by VISDA on the muscular dystrophy dataset. Rectangles contain individual phenotypes. Ellipses contain a group of phenotypes.

2.7.2 Leave-one-out Stability Analysis of TOPs on Cancer Data

On a microarray gene expression dataset that consists of 14 cancer classes and 190 samples (see Table 2.2) [11, 81], we applied leave-one-out stability analysis with n_g = 6. In each experimental trial of the leave-one-out stability analysis, one sample was left out and we constructed a TOP based on the remaining samples. Thus totally 190 TOPs were generated and we took the tree with the highest frequency of occurrence as the final solution, which best reflects the underlying stable structure of the data. As a validation, we compared the most frequent TOP to the known developmental/morphological relationships among the various cancer classes, which was published in [11].

Forty three different TOPs occurred in the leave-one-out stability analysis. The most frequent TOP occurred 121 times; the second most frequent TOP occurred 11 times; the third most frequent TOP occurred 7 times; most of the other TOPs only occurred once. The most frequent TOP has an occurrence frequency of 121/190 ≈ 63.68%. Considering that some TOPs have only minor differences compared to the most frequent

TOP, the underlying stable structure likely has an even higher occurrence frequency. We also applied VISDA on the whole dataset without leaving any sample out and obtained the same structure as the most frequent TOP. Figure 2.9a shows the known pathological cancer tree [11] and Figure 2.9b shows the obtained most frequent TOP. We can see that the most frequent TOP captured some pathological relationships reflected in mRNA profiles. The neoplasm of lymphoma and leukemia are hematolymphoid; appropriately, in the most frequent TOP, they were far away from the other cancer classes whose neoplasm is solid. CNS and mesothelioma were separated from epithelial tumors. The ovary cancer and the uterus cancer are mullerian tumors and closely located in the tree. Breast cancer, bladder cancer and pancreas cancer belong to the non-mullerian category and formed a tight subgroup.

Table 2.2 A Brief Introduction of the Multi-class Cancer Dataset.

Phenotype	Sample Number
Breast Cancer	11
Prostate Cancer	10
Lung Cancer	11
Colon Cancer	11
Lymphoma Cancer	22
Melanoma Cancer	10
Bladder Cancer	11
Uterus Cancer	10
Leukemia Cancer	30
Kidney Cancer	11
Pancreas Cancer	11
Ovary Cancer	11
Mesothelioma Cancer	11
Central Nervous System Cancer	20

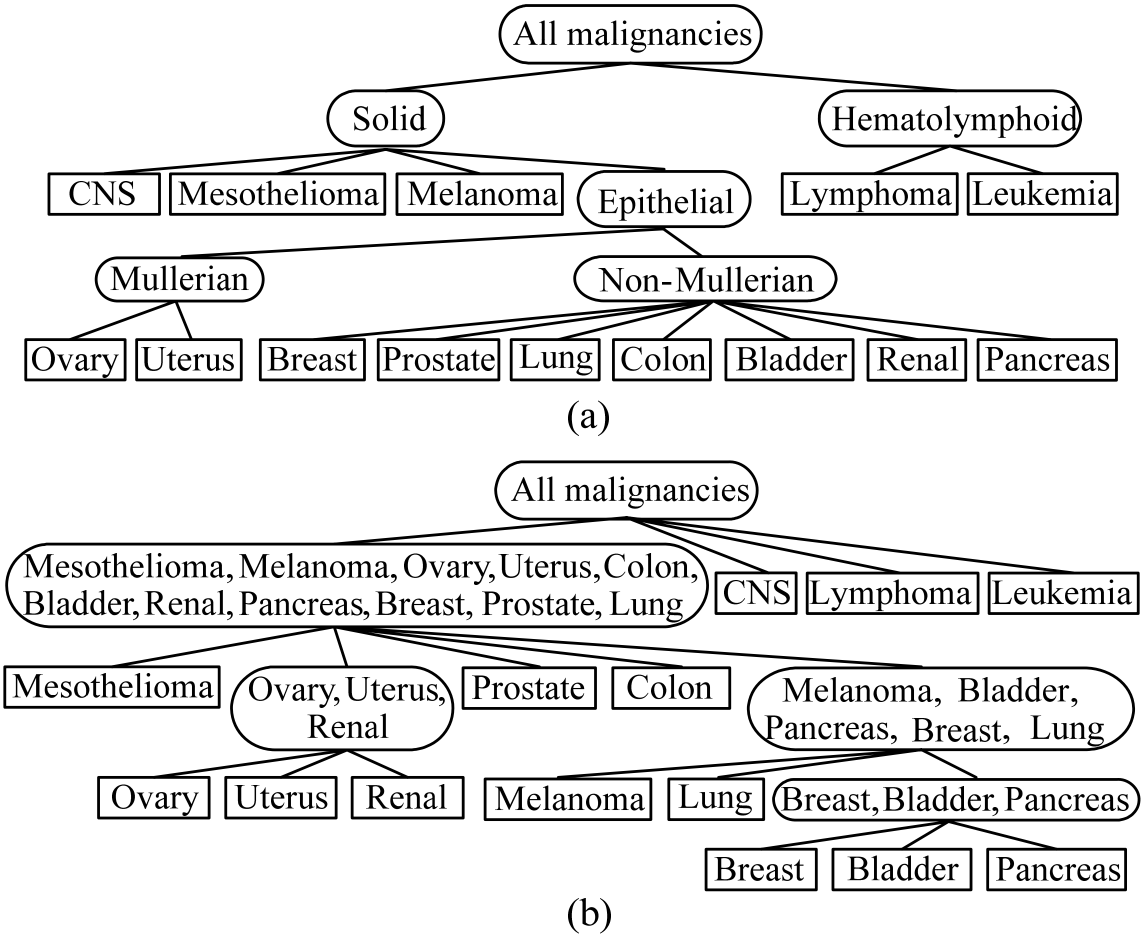


Figure 2.9 Comparison between the most frequent TOP and the pathological relationships among the cancer classes. (a) Published developmental/morphological relationships among the cancer classes. (b) The most frequent TOP constructed by VISDA. Rectangles contain one cancer type. Ellipses contain a group of cancer types.

2.8 Discussion

VISDA is proven to be an effective data clustering tool that incorporates human intelligence and domain knowledge. When applied by experienced users and domain experts, VISDA is more likely to generate accurate/meaningful clustering and visualization results. Since different human-data interaction may lead to different clustering outcomes, to achieve optimum performance, the user needs to acquire experience in using VISDA on various kinds of data, especially on the dataset of interest. Multiple trials applying VISDA are suggested when analyzing a new dataset. Notice that

VISDA only requires the user to have common sense about cluster distribution, cluster separability, and outlier.

Besides the two kinds of non-informative genes discussed in Section 2.3, "redundant" genes (genes that are highly correlated with other genes) provide only limited additional separability between sample clusters. However, this limited additional separability may in fact greatly improve the achievable partition accuracy [82]. Thus, we take removal of redundant genes as an optional step for sample clustering. If the dimensionality of the gene space after variation filtering and discrimination power filtering cannot be well handled by the clustering algorithm (i.e. if the samples-to-genes ratio is not sufficiently large), we suggest removing highly correlated genes. Here, we provide a simple scheme to remove redundant genes. In the gene list resulting from variation filtering and discrimination power analysis, keep the most discriminative gene and remove the genes that are highly correlated with it. Then keep the second most discriminative gene in the remaining list and remove the genes that are highly correlated with this second most discriminative gene. Keep performing this procedure until no further removal can be done. The correlation between genes can be measured by Pearson correlation coefficient or mutual information normalized by entropy. A threshold needs to be set to identify the highly correlated genes.

Various visualization techniques, such as dendrogram, heat maps, and projections, have been applied to present gene expression data structures and clustering outcomes [17, 37, 83, 84]. Many linear/nonlinear projection methods, such as PCA [40], random projection [48], variant of multi-dimensional scaling [83], and projection based on frequency domain analysis [84], have been used to visualize/analyze gene expression data. In VISDA, data are hierarchically visualized using multiple local data projections, one at each node of the hierarchy. Such a hierarchical visualization scheme allows each local data projection to be fulfilled by relatively simple method, i.e. linear projection, while the whole visualization hierarchy is capable to reveal both global and local cluster structures. Since every clustering and visualization method has its own underlying assumptions about the cluster structure of interest [13, 16, 28, 57-59], VISDA provides users with an extensible visualization capability by a projection suite that can incorporate novel, effective, complementary projection methods to increase the likelihood of revealing the data/application-dependent cluster structure of interest. Besides enhancing human understanding of the data structure, data visualization in VISDA has a further function of providing the basis for introducing human intelligence and domain knowledge to the clustering process through human-data interactions.

One point needs to be noted is that VISDA selects a data model in the locally-discriminative low-dimensional visualization space. Although visualization with dimension reduction may reveal only the main data structure and lose minor/local data structures within a cluster, these minor/local structures may become the main data structure captured at subsequent levels. VISDA discovers hierarchical relationships between clusters, which allows analyzing the data at different resolutions/scales. Larger clusters can be obtained by simply merging small clusters according to the hierarchy. The discovered hierarchical relationships among clusters may reveal important biological information, for example the developmental/morphological information among diseases revealed by TOPs. The TOP discovered by VISDA can also be used to construct a hierarchical classifier to solve the complex task of multiple-disease diagnosis by embedding a relatively simple classifier at each node of the TOP, which may obtain good classification performance [85].

Despite our successful applications of VISDA to real microarray gene expression data, there are remaining limitations of the reported method. For example, in sample clustering, dimension reduction via unsupervised informative gene selection is highly data-dependent and often achieves only limited success. This is a very challenging task due to no prior knowledge and potentially complex gene-gene interactions embedded within high-dimensional data. Furthermore, user-data interaction may bring certain subjectivity into the clustering process if not being properly orchestrated, and projection visualization may cause some unrecoverable information loss leading to only a suboptimum solution, although VISDA's hierarchical framework can partially alleviate this problem. Lastly, VISDA presently assumes each cluster follows a Gaussian distribution largely driven by mathematical convenience. However, small sample size problem can defeat this assumption and composite clusters at higher-levels of the hierarchy are not even theoretically normally distributed but are more generally mixture distributions.

2.9 Conclusion

We design and develop a clustering and visualization algorithm for discovering cluster structure in high-dimensional gene expression data. VISDA can discover and visualize gene clusters, sample clusters, phenotype clusters, and the hierarchical relationships between the detected clusters. The hierarchical visualization and clustering scheme of VISDA uses multiple local visualization subspaces (one at each node of the hierarchy) and consequent subspace data modeling to reveal both global and local cluster structures in a “divide and conquer” scenario. VISDA visualizes data by structure-preserving projections and provides an interface for human-data interaction, which facilitates incorporation of expert domain knowledge and human intelligence to help achieve accurate and meaningful data visualization and modeling. The extensible VISDA framework can incorporate various existing clustering and visualization algorithms to increase the likelihood of revealing data structure of interest. Applications to muscular dystrophy, muscle regeneration, and cancer data illustrated that VISDA produced biologically meaningful clustering results that can enhance users' understanding about the underlying biological mechanism and stimulate novel hypotheses for further research.

3 Ground Truth Based Clustering Evaluation and Comparison

3.1 Introduction

In Chapter 2, we have reviewed various data clustering algorithms applied for gene expression data analysis and summarized their major limitations. To overcome some of the limitations, we have developed a hierarchical data exploration and clustering algorithm VISDA for discovering complex cluster structure in high-dimensional gene expression data. However, an important problem associated with data clustering is performance evaluation, i.e. assessing how well the clustering algorithm performs. Given the variety of available clustering methods for gene expression data analysis, it is important to assess the performance and limitations of clustering algorithms to guide users in selecting a suitable one for the analysis task at hand. For our newly developed data clustering algorithm VISDA, we also need to evaluate its performance for gene expression data analysis and compare its performance to those of other existing clustering methods, to further validate and examine the principle of VISDA.

Efforts have been made to evaluate and compare the performance and applicability of various clustering algorithms for gene expression data analysis. As Handl et al. stated in [86], external measures and internal measures currently are the two main lines to validate clustering results. External assessment approaches use knowledge of the correct class labels in defining an objective criterion for evaluating the quality of a clustering solution. Gibbons and Roth used mutual information to examine the relevance between clustered genes and a filtered collection of GO classes [18, 87]. Gat-Viks et al. projected genes onto a line through linear combination of the biological attribute vectors (GO classes) and evaluated the quality of the gene clusters using an ANOVA test [88]. Datta and Datta used a biological homogeneity index (relevance between gene clusters and GO classes) and a biological stability index (stability of the gene clusters' biological relevance with one experimental condition missing) to compare clustering algorithms [89]. Loganantharaj et al. proposed to measure both within-cluster homogeneity and between-cluster separability of the gene clusters with respect to GO classes [90]. Thalamuthu et al. assessed gene clusters by calculating and pooling *p*-values (i.e. the probability that random clustering generates gene clusters with a certain annotation abundance level) of clustering solutions with different numbers of clusters [91].

When trusted class labels are not available, internal measures serve as alternatives. Yeung et al. compared the prediction power of several clustering methods using an adjusted Figure of Merit (FOM) when leaving one experimental condition out [92]. Shamir and Sharan used a FOM-based homogeneity index to evaluate the separation of obtained clusters [93]. Datta and Datta designed three FOM-based consistency measures to assess pair-wise co-assignment of genes, preservation of gene cluster centers, and gene cluster compactness, respectively [94]. A resampling based validity scheme was also proposed in [95].

We propose ground truth based performance evaluation on clustering algorithms using sample clustering [41]. Evaluation using sample clustering assesses clustering outcome with an objective and reliable ground truth, i.e. the known phenotype categories. There are several major differences between our evaluation scheme and previously reported works. First, our evaluation focuses on sample clustering rather than gene clustering. Sample clustering aims to confirm/refine known phenotypes or discover new phenotypes/sub-phenotypes [10]. Sample clustering normally has a much higher attribute-to-object ratio than gene clustering, even after front-end gene selection [31, 50, 66], and imposes a significant challenge to many existing clustering algorithms [81]. Second, instead of using internal measures (consistency) to evaluate the variance but not the bias of clustering outcome, our comparison uses external measures to evaluate both the bias and the stability of the obtained sample clusters respective to the biological categories [96]. Third, our evaluation of clustering algorithms is based on sample clustering against phenotype categories. It is thus more objective and reliable than most reported evaluations, which were based on gene clustering against gene annotations like GO classes. These gene annotations are prone to significant “false positive evidence” when used under biological contexts different from the specific biological processes that produced the annotations in the database. Furthermore, since most GO-like databases only provide partial gene annotations, the comparisons derived from such incomplete truth cannot be considered conclusive.

In this chapter, we design the ground truth based clustering evaluation scheme and use it to conduct a comprehensive performance comparison among VISDA and several existing benchmark clustering methods, i.e. HC [40], KMC [40], SOM [97], and SFNM fitting [40, 98]. Organization of this chapter is as follows. Section 3.2 briefly reviews the four clustering algorithms that will be compared to VISDA. Section 3.3, Section 3.4, and Section 3.5 introduce in details the methodology and experimental design of the evaluation and comparison procedures. Section 3.6 presents the evaluation and comparison results. Section 3.7 is devoted to discussion and conclusion. Some of the notations used in this chapter are the following. Let $\mathbf{t} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_N \mid \mathbf{t}_i \in \mathrm{R}^P, i = 1, \ldots, N\}$ denote N P-dimensional vector-point samples to be clustered. Suppose that the samples are grouped into K_c clusters.

3.2 Competing Clustering Algorithms

3.2.1 Hierarchical Clustering

As a bottom-up approach, agglomerative HC starts from singleton clusters, one for each data point in the sample set, and produces a nested sequence of clusters with the property that whenever two clusters merge, they remain together at any higher level. At each level of the hierarchy, the pair-wise distances between all the clusters are calculated, with the closest cluster pair merged. This procedure is repeated until the top level is reached, where the whole dataset exists as a single cluster. Table 3.1 shows the algorithm of agglomerative HC. The cluster merging process is usually presented using a dendrogram (see Figure 3.1), which can be cut according to different distance thresholds to produce partitions with different number of clusters.

Table 3.1 Algorithm of Agglomerative Hierarchical Clustering

Given	Multiple data points. Each data point forms a singleton cluster.
Step 1	Calculate the pair-wise distances between all clusters.
Step 2	Merge the two nearest clusters.

Step 3 If all data points form one cluster, the algorithm ends; otherwise go to step 1.

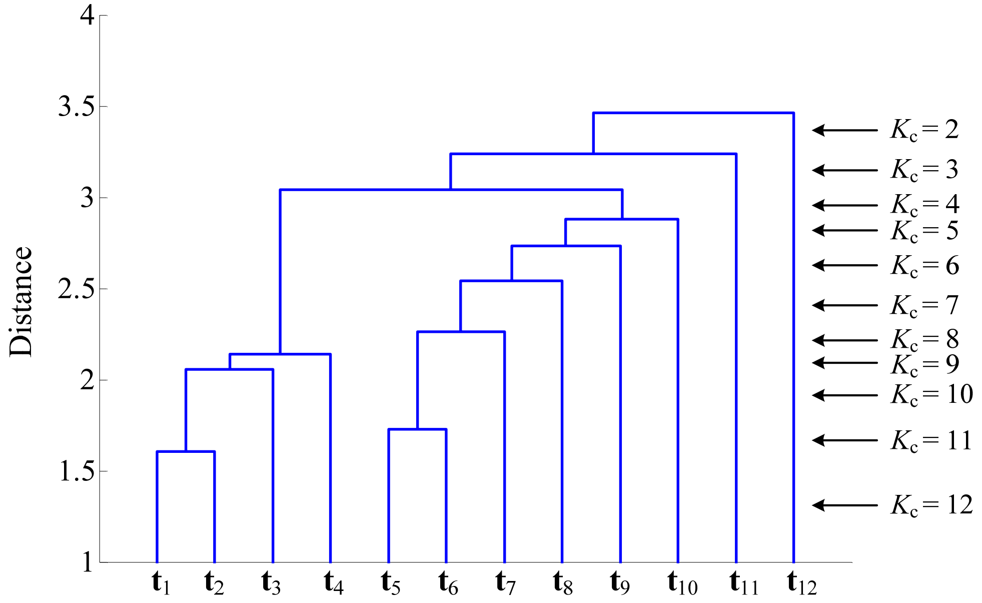


Figure 3.1 Dendrogram obtained by agglomerative HC. Twelve data points $\{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{12}\}$ are clustered using the average distance measure. Vertical coordinate shows the distance values, at which the clusters are merged. Arrows and the corresponding K_c numbers on the right side indicate the resulted number of clusters if we cut on the dendrogram according to different distance thresholds.

There are multiple distance measures available for calculating the distance between two clusters. Let C_1 and C_2 denote two clusters. Let N_{C_1} and N_{C_2} denote the number of data points in C_1 and C_2, respectively. The following equations show four different distance measures, i.e. minimum distance, maximum distance, average distance, and distance between cluster means, given by

$$\begin{aligned} d_{\text{min}}\left(C_1, C_2\right) &= \min_{\substack{\mathbf{t}_q \in C_1 \\ \mathbf{t}_m \in C_2}} \left\|\mathbf{t}_q - \mathbf{t}_m\right\| \\ d_{\text{max}}\left(C_1, C_2\right) &= \max_{\substack{\mathbf{t}_q \in C_1 \\ \mathbf{t}_m \in C_2}} \left\|\mathbf{t}_q - \mathbf{t}_m\right\| \\ d_{\text{ave}}\left(C_1, C_2\right) &= \frac{1}{N_{C_1} N_{C_2}} \sum_{\mathbf{t}_q \in C_1} \sum_{\mathbf{t}_m \in C_2} \left\|\mathbf{t}_q - \mathbf{t}_m\right\| \\ d_{\text{mean}}\left(C_1, C_2\right) &= \left\|\boldsymbol{\mu}_{C_1} - \boldsymbol{\mu}_{C_2}\right\| \end{aligned}, \tag{3.1}$$

where $\boldsymbol{\mu}_{C_1}$ and $\boldsymbol{\mu}_{C_2}$ are the mean of data points in C_1 and C_2, respectively. In the comparison experiment, we used a Matlab implementation of HC with the average distance measure.

3.2.2 K-means Clustering

Widely adopted as a top-down scheme, KMC seeks a partition that (locally) minimizes the Mean Squared Compactness (MSC), the average squared distance between the center of the cluster and its members. Table 3.2 shows the algorithm of KMC. KMC randomly selects K_c data points to initialize the cluster centers and its clustering outcome may be sensitive to the initialization [40]. So multiple runs of KMC with random initialization may generate different data partitions. A common strategy is to select the data partition with minimum MSC as the final clustering outcome. See [40] for a more detailed description of KMC methodology. We used a Matlab implementation of KMC in the experiment.

Table 3.2 Algorithm of K-means Clustering

Given	Multiple data points and the number of clusters K_c.
Step 1	Randomly select K_c data points to initialize the cluster centers.
Step 2	Assign each data point to its nearest cluster center measured by the Euclidean distance between the data point and the cluster center. Thus a partition of the data points is formed.
Step 3	For each cluster, update its cluster center with the mean of the data points assigned to it.
Step 4	If the current data partition is the same as the data partition obtained in the previous iteration, the algorithm ends; otherwise go to step 2.

3.2.3 Self-organizing Maps

SOM performs partitional clustering using a competitive learning scheme [97]. With its roots in neural computation, SOM maps the data from the high-dimensional data space to a low-dimensional output space, usually a 1-D or 2-D lattice. Each node (also called a neuron) of the lattice has a reference vector. In the sequential learning process, when a data point is input, the neuron whose reference vector is closest to the input data point is identified as the winning neuron. The reference vectors of all neurons are updated towards the input data point in proportion to a learning rate and to a neighborhood function of the spatial distance in the lattice between the winning neuron and the given neuron. The update formula for a neuron Γ with a reference vector **δ** is

$$\boldsymbol{\delta}(h+1)=\boldsymbol{\delta}(h)+\mathrm{f}_{\mathrm{N}}\left(\mathrm{d}_{\mathrm{L}}\left(\Gamma_{\mathrm{w}},\Gamma\right),h\right)\eta(h)\left(\mathbf{t}_i-\boldsymbol{\delta}(h)\right), \tag{3.2}$$

where h is the index of update, $\mathbf{t}_i$ is the input data point, Γ_{w} is the winning neuron, $\mathrm{d}_{\mathrm{L}}(\bullet, \bullet)$ returns the distance between Γ_{w} and Γ in the lattice, $\mathrm{f}_{\mathrm{N}}(\bullet, \bullet)$ is a neighborhood function, and $\eta(\bullet)$ is a learning rate.

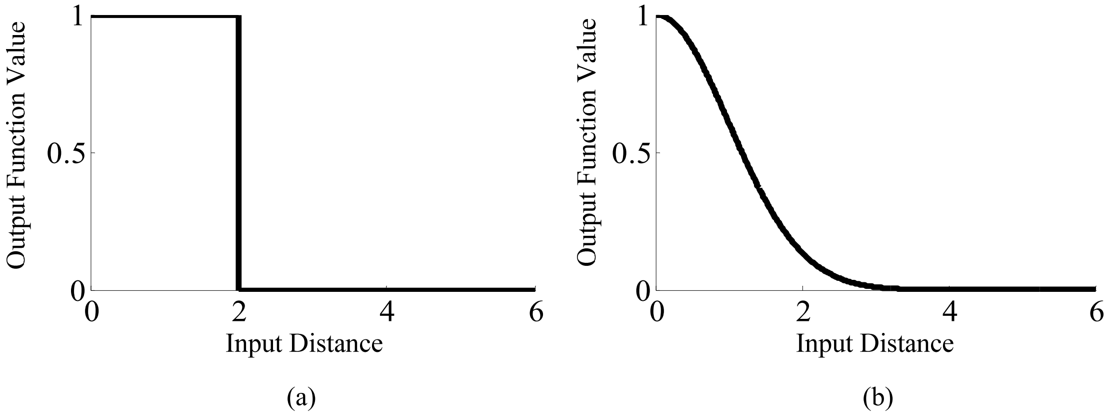


Figure 3.2 Illustration of two types of neighborhood function. (a) Constant window neighborhood function. (b) Gaussian neighborhood function.

To reach convergence, the learning rate $\eta(h)$ starts from a number smaller than 1 such as 0.9 or 0.5, and decreases linearly or exponentially to zero during the learning process as h increases. The neighborhood function $\mathrm{f}_{\mathrm{N}}(\bullet, \bullet)$ can takes a constant window form or a Gaussian form, as shown in Figure 3.2. Regardless of the functional form, the neighborhood function shrinks with time (as h increases). At the beginning when the neighborhood is broad, more neurons are updated and the self-organizing takes place on the global scale; when the neighborhood has shrunk to just a couple of neurons, the reference vectors are converging to local estimates [97]. In the sequential learning process, data points are repeatedly input into the algorithm with a random order. After the learning process converges and ends, each data point is assigned to its winning neuron whose reference vector is closest to it. Data points assigned to the same neuron form a cluster. We used the conventional, sequential SOM implemented by Matlab in the experiment.

3.2.4 SFNM Fitting

The SFNM fitting method uses EM algorithm to estimate an SFNM distribution for the data [40, 98]. Each Gaussian distribution in the SFNM model represents one cluster. An SFNM model can be described by the following probability density function

$$\mathrm{f}\left(\mathbf{t}_i \middle| \boldsymbol{\theta}, \boldsymbol{\pi}\right) = \sum_{j=1}^{K_\mathrm{c}} \pi_j \, \mathrm{g}\left(\mathbf{t}_i \middle| \boldsymbol{\theta}_j\right), \quad \text{and} \quad \sum_{j=1}^{K_\mathrm{c}} \pi_j = 1, \tag{3.3}$$

where g(•) is the Gaussian probability density function, and π_j and $\boldsymbol{\theta}_j$ are the mixing proportion and parameters associated with cluster j, respectively. To (locally) maximize the likelihood of the SFNM model, we use the EM algorithm to train the model. The EM algorithm performs the following two steps alternately until convergence:

$$\begin{aligned}
&\text{E Step:} \quad z_{ij} = \frac{\pi_j \, \mathrm{g}(\mathbf{t}_i | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K_\mathrm{c}} \pi_k \, \mathrm{g}(\mathbf{t}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \\
&\text{M Step:} \quad \pi_j = \frac{1}{N} \sum_{i=1}^{N} z_{ij} \\
&\qquad\qquad \boldsymbol{\mu}_j = \frac{\sum_{i=1}^{N} z_{ij} \mathbf{t}_i}{\sum_{i=1}^{N} z_{ij}} \qquad , \\
&\qquad\qquad \boldsymbol{\Sigma}_j = \frac{\sum_{i=1}^{N} z_{ij} \left(\mathbf{t}_i - \boldsymbol{\mu}_j\right)\left(\mathbf{t}_i - \boldsymbol{\mu}_j\right)^T}{\sum_{i=1}^{N} z_{ij}}
\end{aligned} \tag{3.4}$$

where z_{ij} is the posterior probability of data point i belonging to cluster j, $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ are the mean and covariance matrix of cluster j, respectively. Since the SFNM fitting method gives the posterior probabilities of each data point belonging to different clusters, it provides a soft partition of the data. We implemented SFNM fitting based on Equation (3.4) in our experiments. We randomly selected K_c data points to initialize the cluster means and the covariance matrices of the clusters were uniformly initialized by the covariance matrix of the whole dataset. Note that the trained SFNM model may be sensitive to its initialization. For a more detailed description of SFNM fitting method, please see [40].

3.3 Evaluation Design

Our evaluation focused on three fundamental characteristics of clustering solutions, namely, *functionality*, *accuracy*, and *reproducibility/stability*. Multiple cross-validation trials on multiple datasets are conducted to estimate the performance.

3.3.1 Functionality

Determining the number of clusters and the membership of data points is the major objective of data clustering. Although model order selection criteria have been proposed for use with HC, KMC, SOM, and SFNM fitting algorithms, there is no consensus about the proper model order selection criterion. Thus, we simply fixed the cluster number at the true number of classes for these methods in the evaluation and comparison experiments. VISDA provides an MDL based model order selection module assisted by human justification. We assessed this functionality by its ability to detect the correct cluster number (i.e. the ground truth class number) in cross-validation trials. Furthermore, both SFNM fitting and VISDA provide soft clustering with confidence values; while both HC and VISDA perform hierarchical clustering and show the hierarchical relationship among discovered clusters, which may contain biological meaningful information and allows cluster analysis at multiple resolutions, achieved by simply merging clusters according to the tree structure.

3.3.2 Accuracy

A natural measure of clustering accuracy is the percentage of correctly labeled samples, i.e. the partition accuracy. For soft clustering, soft memberships are transformed to hard memberships via the MAP rule [40] (i.e. each data point is assigned to the cluster that it most likely belongs to, evaluated by its posterior probabilities of belonging to different clusters), to calculate the partition accuracy.

3.3.3 Reproducibility/Stability

Since clustering may help drive scientific hypotheses, it is extremely important that clustering solutions be reproducible/stable. To test the stability of the clustering algorithms, we calculate the variation of the clustering outcomes obtained through N_f-fold cross-validation. In each of the multiple cross-validation trials, only $(N_f - 1)/N_f$ of the samples in each class are used to produce the clustering outcome. Stability of a clustering algorithm is reflected by the resulting standard deviation of partition accuracy. For VISDA, since it includes intensive human data interactions in the clustering process, we want to know how sensitive VISDA is to different users. So we ask multiple users to apply VISDA and examine the variation of partition accuracy and cluster number detection accuracy obtained by different users.

3.3.4 Additional Internal Measures

Besides MSC as an internal clustering validity measure, we can use Mean Log-Likelihood (MLL) for mixture model fitting or Mean Classification Log-Likelihood (MCLL) for the hard clustering result to measure the goodness of fit between the estimated probabilistic model and the soft or hard partitioned data in terms of average joint log-likelihood. Formulas for calculating MSC, MLL, and MCLL are presented in Section 3.4.

3.4 Quantitative Performance Measures

For assessing the model order selection functionality of VISDA, cluster number detection accuracy is calculated based on doubled N_f-fold cross-validation trials, where a detection trial is considered successful if the number of clusters detected by VISDA is equal to the ground truth class number. The formula for calculating cluster number detection accuracy is

$$\frac{\text{number of successful detection trials}}{2 \times N_{\text{f}}} \times 100\% \quad , \qquad (3.5)$$

Mean and standard deviation of the partition accuracy are calculated based on 20 successful detection cross-validation trials in which the cluster number was correctly detected. A prerequisite for calculating the partition accuracy is the correct association between the discovered clusters and ground truth classes. To assure the global optimality of the association, all permuted matches between the detected clusters and the ground truth classes are evaluated. For this purpose, after correctly detecting the cluster number, we calculate the consistency between the permuted cluster labels and the ground truth labels over all data points and choose the association whose consistency is maximum among all permuted matches, given by

$$\mathrm{PA}_l = \max_{\phi \in \Phi_{K_\mathrm{c}}} \frac{1}{N_l} \sum_{i=1}^{N_l} 1\left\{\phi\left(\Theta_l\left(\mathbf{t}_i\right)\right), \Theta^*\left(\mathbf{t}_i\right)\right\} \quad (3.6)$$

,

where PA$_l$ is the partition accuracy in the lth cross-validation trial, Φ_{K_c} is the set of permutations of cluster indices $\{1, 2, \ldots, K_\mathrm{c}\}$, N_l is the number of samples used in the lth trial, $\Theta_l(\mathbf{t}_i)$ is the clustering label of data point $\mathbf{t}_i$ in the lth trial, $\Theta^*(\mathbf{t}_i)$ is the ground true label of $\mathbf{t}_i$, and $1\{\bullet, \bullet\}$ is the indicator function, which returns 1 if the two input arguments are equal and returns 0 if not. Using the Hungarian method, the complexity of the search is $O(N_l+K_\mathrm{c}^3)$ [99].

MSC of hard clustering in the lth cross-validation trial is calculated by

$$\mathrm{MSC}_l = \frac{1}{N_l} \sum_{j=1}^{K_\mathrm{c}} \sum_{i=1}^{N_{lj}} \left\| \mathbf{t}_i - \hat{\boldsymbol{\mu}}_{lj} \right\|^2 , \quad (3.7)$$

where N_{lj} is the number of samples in the jth obtained cluster in the lth cross-validation trial, and $\hat{\boldsymbol{\mu}}_{lj}$ is the mean of cluster j in trial l. MCLL of hard clustering is calculated by

$$\mathrm{MCLL}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \ln\left(\mathrm{g}\left(\mathbf{t}_i \middle| \hat{\boldsymbol{\mu}}_{l,\mathbf{t}_i}, \hat{\boldsymbol{\Sigma}}_{l,\mathbf{t}_i}\right)\right), \quad (3.8)$$

where $\hat{\boldsymbol{\mu}}_{l,\mathbf{t}_i}$ and $\hat{\boldsymbol{\Sigma}}_{l,\mathbf{t}_i}$ are the mean vector and covariance matrix of the cluster that $\mathbf{t}_i$ belongs to in trial l, and g(•) is the Gaussian probability density function. The MLL for soft clustering with an SFNM model (i.e. SFNM fitting method and VISDA) in the lth cross-validation trial is calculated by

$$\mathrm{MLL}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \ln \sum_{j=1}^{K_\mathrm{c}} \pi_{lj}\, \mathrm{g}\left(\mathbf{t}_i \middle| \hat{\boldsymbol{\mu}}_{lj,\mathrm{SFNM}}, \hat{\boldsymbol{\Sigma}}_{lj,\mathrm{SFNM}}\right), \quad (3.9)$$

where $\pi_{lj} = \sum_{i=1}^{N_l} z_{lij} \Big/ N_l$ (z_{lij} is the posterior probability of sample i belonging to cluster j in trial l) is the sample proportion of cluster j in the lth trial, $\hat{\boldsymbol{\mu}}_{lj,\mathrm{SFNM}} = \sum_{i=1}^{N_l} z_{lij}\mathbf{t}_i \Big/ \sum_{i=1}^{N_l} z_{lij}$ is the mean of cluster j in trial l, and $\hat{\boldsymbol{\Sigma}}_{lj,\mathrm{SFNM}}$ is the covariance matrix of cluster j in trial l calculated by $\hat{\boldsymbol{\Sigma}}_{lj,\mathrm{SFNM}} = \sum_{i=1}^{N_l} z_{lij}\left(\mathbf{t}_i - \hat{\boldsymbol{\mu}}_{lj,\mathrm{SFNM}}\right)\left(\mathbf{t}_i - \hat{\boldsymbol{\mu}}_{lj,\mathrm{SFNM}}\right)^T \Big/ \sum_{i=1}^{N_l} z_{lij}$.

3.5 Datasets and Additional Experimental Details

We chose a total of seven real microarray gene expression datasets and one synthetic dataset for this ground truth based comparative study, summarized in Table 3.3. The synthetic dataset consists of 4 normally-distributed clusters each with 100 samples in 3-D space. The cluster means are set to be $[2\ 2\ 0]^T$, $[2\ \text{-}2\ 0]^T$, $[\text{-}2\ 0\ \text{-}2]^T$, and $[\text{-}2\ 0\ 2]^T$. The cluster covariance matrices are set to be [1 0 0; 0 1 0; 0 0 3], [1 0 0; 0 1 0; 0 0 3], [1 0 0; 0 4 0; 0 0 1], and [1 0 0; 0 4 0; 0 0 1]. To assure a meaningful and well-grounded comparison, we emphasized the quality and suitability of these test datasets. For example, the datasets cannot be too “simple” (if the clusters are well-separated, all methods perform equally well) or too “complex” (no method will then perform reasonably well). Specifically, each cluster must be reasonably well-defined (for example, not a composite cluster) and contain sufficient data points.

Since there are a significant number of irrelevant genes (genes that do not respond to the physiological event, nor discriminate between phenotypes) in gene expression data [16, 81], we performed supervised front-end gene selection to select discriminative genes to ensure the basic functional condition for a meaningful comparison of the clustering algorithms, using the method introduced in [100, 101]. Supervised front-end gene selection using the knowledge of known classes is considered both logical and necessary, since the known classes are the “default” clusters whose distinctive multimodal distributions should be well-preserved in the genes used for clustering. The preprocessed datasets cover both the data-sufficient case that has a low attribute-to-object ratio and resembles typical gene clustering and the data-insufficient case that has a high attribute-to-object ratio and resembles typical sample clustering [64].

On all but one of the datasets, the sample number of the smallest phenotype category is bigger than or equal to 10, and the cross-validation fold number N_f is set at 10. On the dataset whose smallest phenotype category has 9 samples, we set N_f at 9. For the clustering algorithms that do not have a model order selection function, we set the ground truth class number as the input cluster number. For example, the dendrogram of HC was cut according to a threshold that produced a partition with a resulted cluster number equal to the ground truth class number, and the cluster centers in KMC, SOM, and SFNM algorithms were initialized by K_c randomly chosen samples. We used the best outcome from multiple runs of these randomly initialized clustering algorithms, evaluated using the aforementioned criteria. The KMC was chosen based on MSC. SOM was separately chosen based on both MSC and MCLL. SFNM fitting used MLL as the optimality criterion. Specifically, for each of the 20 successful detection cross-validation trials based on which the performance measures were calculated, the clustering procedure was performed 100 times, each with a different random algorithm initialization, for selecting the best outcome. For SOM, two different neighborhood functions were used 50 times in each cross-validation trial.

Table 3.3 Microarray Gene Expression Datasets Used in the Experiment

Dataset Name	Brief Description	Biological Category (Number of Samples in the Category)	Number of Categories / Selected Genes	Source
SRBCTs	Small round blue cell tumours	Ewing sarcoma (29), burkitt lymphoma (11), neuroblastoma (18), and rhabdomyosarcoma (25)	4/60	[14]
Multiclass Cancer	Multiple human tumour types	Prostate cancer (10), breast cancer (12), kidney cancer (10), and lung cancer (17)	4/7	[102]
Lung Cancer	Lung cancer sub-types and normal tissues	Adenocarcinoma (16), normal lung (17), squamous cell lung carcinoma (21), and pulmonary carcinoids (20)	4/13	[103]
UM Cancer	Classification of multiple human cancer types	Brain cancer (73), colon cancer (60), lung cancer (91), ovary cancer (119, including 6 uterine cancer samples)	4/8	[77]

Ovarian Cancer	Ovarian cancer sub-types and clear cell	Ovarian serous (29), ovarian mucinous (10), ovarian endometrioid (36), and clear ovarian cell (9)	4/25	[11, 104]
MMM-Cancer 1	Human cancer data from multi-platforms and multi-sites	Breast cancer (22), central-nervous meduloblastoma (57), lung-squamous cell carcinoma (20), and prostate cancer (39)	4/15	[105]
MMM-Cancer 2	Human cancer data from multi-platforms and multi-sites	Central-nervous glioma (10), lung-adenocarcinoma (58), lung-squamous cell carcinoma (21), lymphoma-large B cell (11), and prostate cancer (41)	5/20	[105]

3.6 Evaluation and Comparison Results

The experimental results are summarized in Table 3.4 ~ 3.8. Cluster number detection accuracy of VISDA is given in Table 3.4. The mean and standard deviation of obtained partition accuracy are given in Table 3.5. Table 3.6 gives the evaluation results based on internal measures, i.e. MSC and MLL. Table 3.7 and Table 3.8 present the evaluation results of VISDA's sensitivity to different users.

3.6.1 Cluster Number Detection Accuracy

VISDA achieves an average cluster number detection accuracy of 97% over all the datasets. This result indicates the effectiveness of the model order selection module of VISDA that exploits and combines the hierarchical SFNM model, the structure-preserving 2-D projections, the MDL model order selection in projection space, and human-computer interactions for visualization selection, model initialization, and cluster validation.

Table 3.4 Cluster Number Detection Accuracy of VISDA Obtained through Cross-validation Experiments

	Synthetic Dataset	SRBCTs	Multiclass Cancer	Lung Cancer	UM Cancer	Ovarian Cancer	MMM Cancer (1)	MMM Cancer (2)	Average
Detection Accuracy	100%	95%	100%	100%	100%	94.44%	90%	100%	97%

3.6.2 Partition Accuracy

Table 3.5 shows the mean and standard deviation of the partition accuracy obtained through cross-validation trails. VISDA gives the highest average partition accuracy -- 86.29% over all the datasets. Optimum SOM selected by MCLL ranks second with an average partition accuracy of 79.39%. On the synthetic dataset, both VISDA and SFNM fitting achieve the best average partition accuracy of 94.89%. On SRBCTs dataset, the average partition accuracies of optimum SOM selected by MCLL (94.32%) and VISDA (94.23%) are comparable. Optimum KMC and SOM selected by MSC show similar performance on all datasets.

On the synthetic data and the majority of the real microarray datasets, HC gives much lower partition accuracy as compared to all other competing methods. HC is very sensitive to outliers/noise and often produces very small or singleton clusters. On the relatively easy case of the synthetic data, KMC, SOM, VISDA, and SFNM fitting achieve almost equally good partition accuracy, with slightly better performance achieved by using soft clustering. On the two most difficult cases, the ovarian cancer and MMM-Cancer 2 datasets, HC achieves comparable partition accuracy to those of optimum KMC and SOM selected by MSC, while VISDA consistently outperforms all other methods. Interestingly, we have found that the optimum SOM selected by MCLL generally gives higher partition accuracy than that of optimum SOM selected by MSC. As a more complex model, SFNM fitting method performs well on the datasets with sufficient samples, such as the synthetic dataset and UM cancer dataset. However, when the sample size becomes relatively small and the attribute-to-object ratio becomes high, its performance significantly degrades, probably because of over-fitting, local optima, or inaccurate parameter estimation due to the curse of dimensionality.

Table 3.5 Mean/Standard-deviation of Partition Accuracy Obtained in Cross-validation Experiments

	VISDA	HC	KMC	SOM(MSC)	SOM(MCLL)	SFNM Fitting
Synthetic Data	**94.89%** /0.67%	52.11% /10.37%	92.14% /0.51%	92.14% /0.51%	92.18% /**0.49%**	**94.89%** /0.64%

SRBCTs	94.23% /3.01%	46.96% /11.71%	81.52% /5.68%	81.66% /5.65%	**94.32%** /4.98%	36.74% **/2.66%**
Multiclass Cancer	**94.66%** /2.08%	66.22% **/1.72%**	92.28% /11.49%	92.28% /11.49%	94.46% /8.74%	62.33% /10.97%
Lung Cancer	**79.00%** /7.43%	57.43% **/2.17%**	68.57% /6.73%	68.49% /6.31%	71.78% /4.75%	51.05% /7.99%
UM Cancer	**94.66%** /0.85%	64.14% /5.39%	84.84% **/0.49%**	84.84% **/0.49%**	82.20% /7.89%	93.59% /0.88%
Ovarian Cancer	**65.39%** /9.98%	59.83% /4.92%	55.47% /2.53%	55.40% /2.38%	55.07% **/2.21%**	43.14% /6.24%
MMM-Cancer 1	**89.36%** /3.06%	67.89% /1.74%	81.83% **/0.87%**	81.83% **/0.87%**	80.65% /4.16%	79.00% /4.44%
MMM-Cancer 2	**78.12%** /5.03%	56.50% **/2.20%**	55.08% /3.10%	55.55% /3.09%	64.46% /4.58%	55.05% /6.78%
Average	**86.29%** /4.01%	58.89% /5.03%	76.47% /3.92%	76.52% **/3.85%**	79.39% /4.73%	64.47% /5.07%

The highest mean and the smallest standard deviation of partition accuracy obtained on each dataset are in bold font.

From the standard deviation of partition accuracy, we can see that optimum SOM selected by MSC has the most stable partition accuracy, followed by optimum KMC selected by MSC and VISDA. These three methods generate clusters with more stable biological relevance than the other methods.

3.6.3 Additional Internal Measures

MSC and MLL are two popular internal measures that we also examined in the experiment (see Table 3.6). Since these two additional measures do not have a direct relation to the ground truth, although being easily adopted, the conclusions drawn from their values could be misleading and should be used with caution. For example, optimum KMC and optimum SOM selected by MSC consistently achieve the smallest MSC, (somewhat unexpectedly) even smaller than the MSC calculated based on the ground truth partition, i.e. the phenotype categories. Based on their corresponding imperfect partition accuracies, this result indicates that solely minimizing MSC does not constitute an unbiased clustering approach. A similar situation was observed for the MLL criterion with additional issues of inaccurate estimation of the second order statistics and local optima caused by both the curse of dimensionality and covariance matrix singularity. VISDA generally has smaller MLL values than the SFNM fitting method, while VISDA has better partition accuracy.

Table 3.6 Mean of MSC/ MLL Obtained through Cross-validation Experiments

	VISDA	HC	KMC	SOM (MSC)	SOM (MCLL)	SFNM fitting	Ground truth
Synthetic data	5.68e+0 **/-6.19e+0**	9.52e+0	**5.52e+0**	**5.52e+0**	**5.52e+0**	5.68e+0 **/-6.19e+0**	5.78e+0
SRBCTs	5.46e+1 **/-5.99e-1**	7.41e+1	**4.76e+1**	**4.76e+1**	5.12e+1	1.09e+2 /-8.33e+1	5.22e+1
Multiclass cancer	1.71e+5 /-4.21e+1	1.65e+5	**1.56e+5**	**1.56e+5**	1.58e+5	5.89e+5 **/-3.88e+1**	1.60e+5
Lung cancer	5.02e+5 /-7.15e+1	5.49e+5	**4.32e+5**	**4.32e+5**	4.33e+5	1.53e+6 **/-6.70e+1**	5.40e+5
UM cancer	9.07e+7 /-6.53e+1	8.99e+7	**5.42e+7**	**5.42e+7**	5.68e+7	9.25e+7 **/-6.52e+1**	8.75e+7
Ovarian cancer	5.10e+7 /-1.84e+2	5.87e+7	**4.72e+7**	**4.72e+7**	4.73e+7	8.74e+7 **/-1.64e+2**	5.70e+7
MMM-cancer 1	4.58e+6 /-8.98e+1	3.95e+6	**3.01e+6**	**3.01e+6**	3.32e+6	6.26e+6 **/-8.90e+1**	4.72e+6
MMM-cancer 2	2.42e+8 /-1.53e+2	1.41e+8	**1.18e+8**	**1.18e+8**	1.29e+8	3.16e+8 **/-1.50e+2**	2.66e+8

Best mean of MSC obtained on each dataset is indicated by bold font. Mean of MLL is shown only for VISDA and SFNM fitting. Better mean of MLL of VISDA and SFNM fitting method is also indicated by bold font. “Ground truth” indicates the mean of MSC calculated based on the phenotype categories.

3.6.4 Evaluation on VISDA’s Sensitivity to Users

For VISDA, since it includes intensive human-data interaction, we also want to test its sensitivity to users. We asked ten different users including both experienced and relatively inexperienced clustering analysts to apply VISDA on four of the eight datasets used in the performance comparison experiment, i.e. synthetic dataset, UM cancer dataset, multiclass cancer dataset, and MMM-Cancer 1 dataset. The mean and standard deviation of partition accuracy, and cluster number detection accuracy obtained by the ten users through cross-validation experiments on the datasets are shown in Table 3.7. Based on the average performance measurements obtained by different users shown at the bottom row of Table 3.7, we can calculate the mean and standard deviation of these average performance measurements respective to different users (see Table 3.8).

Table 3.7 Mean/Standard-deviation of Partition Accuracy (Cluster Number Detection Accuracy) of VISDA Applied by Different Users in the Cross-validation Experiments

User ID Dataset	1	2	3	4	5	6	7	8	9	10
Synthetic Data	94.89% /0.67% (100%)	94.87% /0.63% (100%)	94.74% /0.78% (100%)	94.82% /0.65% (100%)	94.68% /0.71% (100%)	94.83% /0.63% (100%)	94.76% /0.65% (100%)	94.72% /0.68% (90%)	94.79% /0.64% (95%)	94.94% /0.54% (75%)
UM Cancer	94.66% /0.85% (100%)	94.99% /0.86% (95%)	94.91% /0.95% (100%)	94.88% /1.09% (95%)	94.80% /1.16% (100%)	93.47% /1.13% (90%)	94.14% /1.36% (95%)	93.25% /3.68% (90%)	91.5% /8.65% (95%)	93.84% /1.09% (75%)
Multiclass Cancer	94.66% /2.08% (100%)	93.76% /1.63% (85%)	94.22% /1.37% (100%)	94.79% /2.1% (100%)	95.35% /2.26% (100%)	93.87% /1.08% (100%)	93.76% /1.8% (100%)	93.43% /1.93% (100%)	92.63% /5.4% (95%)	94.1% /1.87% (100%)
MMM-Cancer 1	89.36% /3.06% (90%)	90.70% /3.00% (80%)	79.66% /7.21% (85%)	86.28% /8.74% (80%)	89.69% /6.29% (90%)	91.55% /1.34% (100%)	89.25% /5.09% (100%)	88.4% /6.42% (95%)	87.78% /5.8% (90%)	88.79% /5.18% (100%)
Average	93.39% /1.67% (97.5%)	93.58% /1.53% (90%)	90.88% /2.58% (96.25%)	92.69% /3.15% (93.75%)	93.63% /2.61% (97.5%)	93.43% /1.05% (97.5%)	92.98% /2.23% (98.75%)	92.45% /3.18% (93.75%)	91.68% /5.12% (93.75%)	92.92% /2.17% (87.5%)

We can see that the standard deviation of the average of mean partition accuracy obtained by the ten different VISDA users is 0.89%, which is considered small. The standard deviation of the average standard deviation of partition accuracy obtained by the ten different users is 1.14%, which is also small. The standard deviation of the average cluster number detection accuracy obtained by the ten different users is 3.63%, which is small considering that one single unsuccessful detection trial can result in a decrease of cluster number detection accuracy of 5%. Thus, we can see that VISDA's sensitivity to different users on these four representative testing datasets is low. Since the ten users include both experienced and relatively inexperienced data clustering analysts, this result indicates that VISDA does not have high or special requirements on users' skills. Common sense about cluster structure and data visualization will be enough for the users to apply VISDA.

Table 3.8 Mean and Standard Deviation of Average Performance Measurements Obtained by Different Users.

	Average of Mean Partition Accuracy	Average of Standard Deviation of Partition Accuracy	Average of Cluster Number Detection Accuracy
Mean	92.76%	2.53%	94.63%
Standard Deviation	0.89%	1.14%	3.63%

3.7 Discussion and Conclusion

We designed a ground-truth based clustering evaluation scheme, based on which we conducted a comparative study of five clustering methods on gene expression data. The five clustering methods, i.e. VISDA, HC, KMC, SOM, and SFNM fitting, were compared on seven carefully-chosen real microarray gene expression datasets and one synthetic dataset with definitive ground truth. Multiple objective and quantitative performance measures were designed, justified, and formulated to assess the clustering accuracy and stability. Since VISDA incorporates intensive human interactions in the clustering process, we also examined the sensitivity of its clustering performance to different users.

Our experimental results showed that VISDA achieved greater clustering accuracy on most of the datasets than other methods. VISDA was also a stable performer among the competing methods. Its hierarchical exploration process with model order selection in low-dimensional locally-discriminative visualization spaces also provided an effective model order selection scheme for high-dimensional data. Common sense about cluster structure and data visualization will be enough for the users to apply VISDA, which is indicated by the small variations of partition accuracies and cluster number detection accuracies obtained by ten different VISDA users that include both experienced and relatively inexperienced data clustering analysts.

SOM optimized by the MCLL criterion produced the second best clustering accuracy overall. KMC and SOM optimized by the MSC criterion generally produced more stable clustering solutions than the other methods. The SFNM fitting method achieved good clustering accuracy in data-sufficient cases, but not in data-insufficient cases. The experiments also showed that for gene expression data, solely minimizing mean squared compactness of the clusters or solely maximizing mixture model likelihood may not yield biologically plausible results.

Several important points remain to be discussed. First, our comparative study focused on sample clustering [10, 66, 81], rather than gene clustering [9, 29]. Sample clustering in biomedical research often aims to either confirm/refine the known disease categories [81] or discover novel disease types/subtypes [10]. The expected number of "local" clusters of interest is often moderate [10, 14], e.g. 3~5 clusters as presented in our testing datasets. Compared to gene clustering, sample clustering faces a much higher attribute-to-object ratio and consequently a more severe "curse of dimensionality". While most existing comparison studies have been devoted to gene clustering, we believe that it is equally important to assess the competence of the competing clustering methods on sample clustering with high attribute-to-object ratios. In our comparison, even after front-end gene selection, some datasets still have much higher attribute-to-object ratios than typical gene clustering. Furthermore, if the competing methods are applied to gene clustering, the comparison of the methods is expected to be similar to what was seen on the synthetic dataset, where the attribute-to-object ratio is low.

Second, although VISDA and SFNM fitting methods both utilize a normal mixture model and performed similarly in data-sufficient cases, VISDA outperformed SFNM fitting in the data-insufficient cases. A critical difference between these two methods is that, unlike SFNM fitting, VISDA does not apply a randomly initialized fitting process but performs maximum likelihood fitting guided/constrained by the human operator's understanding of the data structure. Additionally, the hierarchical data model and exploration process of VISDA apply the idea of "divide and conquer" to find both global and local data structure.

Third, we selected representative clustering algorithms from various algorithm categories to conduct the comparison. Some of the selected algorithms may have more sophisticated variants; however, a more complex algorithm does not necessarily lead to stable clustering outcomes, as we observed in the experiments.

4 Convex Analysis of Mixtures for Gene Expression Dissection

4.1 Introduction

Many (dynamic) biological phenomena, such as muscle regeneration, are usually a joint effect of multiple underlying activated biological processes. A biological process is a collection of molecular events specifically pertinent to the functioning of integrated living units: cells, tissues, organs, and organisms [18]. Each biological process is characterized by a group of genes that are key players for the biological process to fulfill its specific biological functions. Dissection of the observed gene expression data into gene expression patterns that correspond to underlying biological processes, identification of the key genes of the biological processes, and estimation of the activity levels of the biological processes under different biological conditions (or at successive time points), will help understand the complex mechanisms of biological system.

The relationship between the observed gene expression data and the underlying active biological processes can be described by a linear mixture model **X** = **AS** [19, 20]. **X** is a known data matrix containing observed mixtures (i.e. observed gene expression profiles in the context of gene expression dissection) and each row of **X** represents a mixture. **A** is an unknown mixing matrix whose entries indicate the activity levels of the biological processes. **S** is an unknown non-negative data matrix containing underlying sources (i.e. gene expression patterns of underlying activated biological processes in the context of gene expression dissection) and each row of **S** represents a source. The number of mixtures is the number of gene expression profiles; the number of sources is the number of biological processes; and the data length is the number of genes. Identifying the expression patterns of underlying biological processes and their mixing coefficients, based on only the observed mixtures, is a typical nBSS problem. Notice that in this chapter, to accommodate the tradition of BSS research community, we use the term “sample” to refer to a mixture instance or a source instance, which is actually a gene’s expression values in the mixtures or sources. Thus, a mixture sample contains the observed expression values of a gene in different profiles and a source sample contains the unknown expression values of the gene in different biological processes.

4.1.1 Existing nBSS Methods and Their Major Limitations

A widely applied and intensively studied BSS solution is Independent Component Analysis (ICA) that can identify mutually independent sources [106], although the source independence assumption may hardly be true in many real-world applications. Many efforts aim to empirically incorporate intrinsic non-negativity of sources into the ICA-based principles for solving nBSS problems and have made some reasonable progress [107]. By imposing source non-negativity and that for each source and any $\delta > 0$, the probability that the source value is smaller than δ is nonzero [108], non-negative Independent Component Analysis (nICA) [108] can recover statistically uncorrelated non-negative sources. Stochastic Non-negative Independent Component Analysis (SNICA) does not require the sources to be independent or uncorrelated, and identifies least dependent components by minimizing the mutual information between recovered sources through a Monte Carlo search with the source non-negativity as a hard constraint [107]. Despite the progress in developing various nBSS techniques, accurate separation of dependent sources remains a challenging task [109].

Some nBSS methods aim to minimize the model fitting error, such as the square fitting error, of the matrix factorization form. Alternating least squares, a method that has been used in areas including analytical chemistry, alternatively updates the estimates for the mixing matrix and the sources by solving a series of least square problems [110, 111]. Non-negative Matrix Factorization (NMF) is a benchmark method used to decompose a non-negative mixture data matrix into two non-negative matrices [112], one as the estimate for sources and the other as the estimate for mixing matrix. However, it is well known that the solution of NMF is not unique, which means that NMF does not guarantee identifying the sources and may converge to a poor local optimum. Some algorithm modifications and additional constraints on sources or mixing matrix have been proposed to improve the performance of NMF. For example, Sparse Non-negative Matrix Factorization (SNMF) puts a source sparseness measure term in the objective function to be optimized, so that the recovered sources tend to be sparse [113].

Another branch of nBSS solutions is based on the well-grounded assumption, i.e. for each source there exist samples that are exclusively dominated by the source and these dominated, pure-source samples are called Well-Grounded Points (WGPs) of the source. To clearly illustrate the concept of well-groundedness, we give the following realization of **X**, **A**, and **S**,

$$\mathbf{X}=\begin{bmatrix}0.05 & 0.15 & 0.3 & 0.5\\ 0.1 & 0.25 & 0.15 & 0.5\\ 0.35 & 0.05 & 0.1 & 0.5\\ 0.15 & 0.15 & 0.2 & 0.5\end{bmatrix}=\mathbf{AS}=\begin{bmatrix}0.1 & 0.3 & 0.6\\ 0.2 & 0.5 & 0.3\\ 0.7 & 0.1 & 0.2\\ 0.3 & 0.3 & 0.4\end{bmatrix}\begin{bmatrix}0.5 & 0 & 0 & 0.5\\ 0 & 0.5 & 0 & 0.5\\ 0 & 0 & 0.5 & 0.5\end{bmatrix}. \quad (4.1)$$

The first mixture sample (i.e. the first column vector of **X**) is a WGP of the first source (i.e. the first row of **S**), because its source sample $[0.5\ 0\ 0]^T$ takes a non-zero value only in the first source. The second and third mixture samples are the WGP of the second and third sources, respectively. Due to the existence of these three WGPs, the three sources in Equation (4.1) are well-grounded. The well-grounded assumption is a good approximation of high contrast in signals and applicable to many practical problems, such as hyperspectral imaging [114, 115] and biomedical imaging [22, 116].

It is obvious that under the well-grounded assumption any mixing matrix column vector that contains the activity levels of a source in different mixtures is a linear scaling of the source's WGPs. So the mixing matrix can be recovered with scale ambiguity by identifying the WGPs. An intuitive, geometric way to identify WGPs is to first perspectively project the mixture samples onto a hyperplane, on which WGPs become vertices of the convex hull formed by the projected mixture samples [117]. Since perspective projection is simply a positive linear scaling of the samples and samples with a common vector direction are scaled to have the same vector norm [118], WGPs of a source overlap after projection and form one vertex of the convex hull. Then, conventional convex hull algorithms, e.g. QHull algorithm [119, 120], can be used to detect all the vertices without knowledge of the source number that is also the convex hull vertex number [117]. The limitation of this geometric approach is that convex hull algorithms usually have very high computational complexity when applied on high-dimensional data [121], which impedes its wide usage. If the source number is known, some other strategies can be used to identify WGPs. A typical strategy is to maximize simplex volume, which is adopted by N-FINDR [114], a popular method used in hyperspectral imaging. The first step of N-FINDR is similar to that of the geometric approach. The mixture samples are perspectively projected onto a hyperplane whose dimensionality is one less than the source number, so that the projected samples form a simplex with the WGPs being the vertices. Then N-FINDR identifies WGPs by seeking a sample set whose set size is equal to the source number, and the simplex formed by the sample set has the maximum volume [114].

However, there exist limitations associated with the application of existing nBSS methods to gene expression dissection. First, there is no biological knowledge available for assuring that the biological processes are uncorrelated or independent. Actually, some genes may participate simultaneously in multiple biological processes, which makes the expression patterns of these biological processes correlated. Second, methods such as NMF and its variants (e.g. SNMF) have drawbacks like non-unique solutions and convergence to poor local optimums, and the gene expression patterns of biological processes are not necessary to be overall sparse. Third, the existing non-negative well-grounded source separation methods, such as N-FINDR and the geometric approach, either need the knowledge of source number or have very high computational complexity when applied to high-dimensional data.

Two other problems in nBSS research that have not been well/fully solved by existing methods are noise reduction and source number detection. Many existing nBSS methods use linear dimension reduction, such as PCA, to reduce noise impact [21, 115]. However, the dimension number after reduction needs to be “smartly” selected according to some prior knowledge or good estimation for the noise level and the source number; otherwise, severe dimension reduction may cause the loss of valuable information and the transferring of a determined (i.e. equal number of sources and mixtures) or over-determined (i.e. more mixtures than sources) case into an under-determined case (i.e. more sources than mixtures) that is usually unidentifiable. Importantly, another limitation of noise reduction by linear projections is that linear projections only eliminate the noise existing in the removed dimensions and leave the noise existing in remaining dimensions untouched.

Existing nBSS methods usually determine the source number using information theoretical criteria like MDL [122, 123] or hypothesis tests based on eigenvalues of the covariance and correlation matrices of the mixture samples [124, 125]. However, information theoretical criteria require fitting the data with probabilistic models, which may be difficult when the data have a complex distribution, for example, gene expression data that can hardly be modeled by simple parametric probabilistic models [5, 126]; and hypothesis tests often require preset thresholds that may be subjective, and they cannot be applied to the under-determined case, which means that in practical use they cannot tell whether a separation task is under-determined and thus unidentifiable.

2.3.2 Our Approaches for Gene Expression Dissection

Employing a realistic understanding and contemplation of the biological processes, we design a linear mixture model suitable for gene expression dissection. Since each biological process has its distinct genomic functions, some process-specific genes are expected to be highly expressed in it while insignificantly or even not expressed in all other biological processes. These specific genes are the key players for the biological process to fulfill its genomic functions and thus are called Process Specific Genes (PSGs). Note that PSGs only constitute a small portion of all the genes, while a majority of the genes provide basic cellular structure and supporting functions so that the PSGs can function properly. Genes responsible for basic cellular structure and functions are commonly required and similarly expressed in all biological processes. Thus they are called Process Common Genes (PCGs). Strictly speaking, there may exist some intermediate genes between PSGs and PCGs. Note that the existence of many PCGs makes the sources in our model statistically dependent, correlated, and overall non-sparse. To illustrate the concepts of PSGs and PCGs, Figure 4.1 shows the scatter plot of three gene expression profiles including an ovarian serous adenocarcinoma profile, an ovarian mucinous adenocarcinoma profile, and an ovarian endometrioid adenocarcinoma profile [104]. These three gene expression profiles are from three ovarian cancer subtypes involving different gene regulation mechanisms and thus represent three different cell-type level biological processes. From Figure 4.1, we can clearly see the existence of PSGs and PCGs, as well as some intermediate genes between PSGs and PCGs.

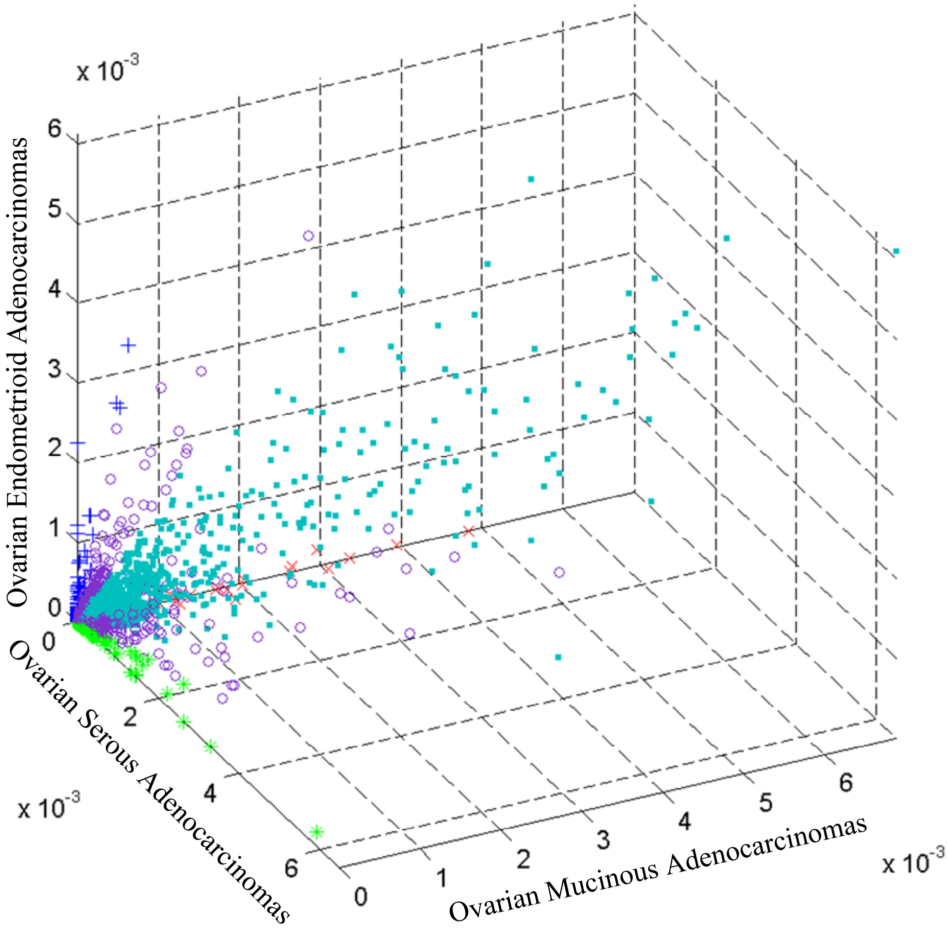


Figure 4.1 Scatter plot illustration of PSGs and PCGs. Green star markers indicate PSGs of ovarian serous adenocarcinoma. Red cross makers indicate PSGs of ovarian mucinous adenocarcinoma. Blue plus signs indicate PSGs of ovarian endometrioid adenocarcinoma. PCGs are indicated by cyan square markers. The purple circles indicate the intermediate genes between PCGs and PSGs.

Source well-groundedness is a realistic assumption based on the existence of PSGs, which are approximately WGPs. We mix the three ovarian cancer profiles using a mixing matrix of [0.7 0.15 0.15; 0.15 0.7 0.15; 0.15 0.15 0.7], and draw the scatter plot of the mixtures in Figure 4.2. We can see that the PSGs are very close to the mixing matrix column vectors indicated by grey lines in Figure 4.2. Thus the mixing matrix column vectors can be well estimated by identifying PSGs that are approximately WGPs. Based on this observation we can develop an nBSS method that utilizes geometric concepts, convex analysis and optimization techniques to identify WGPs for recovering the mixing matrix, and then recovers sources by inverse or pseudo-inverse of the mixing matrix. Since convex analysis and optimization will play an important role in the proposed method, we name the solution Convex Analysis of Mixtures (CAM) [127]. To overcome the limitations of existing nBSS solutions for identifying well-grounded sources, CAM detects WGPs without knowledge of the source number and has lower computational complexity for analyzing high-dimensional data, where the algorithm's computational complexity usually becomes a serious concern.

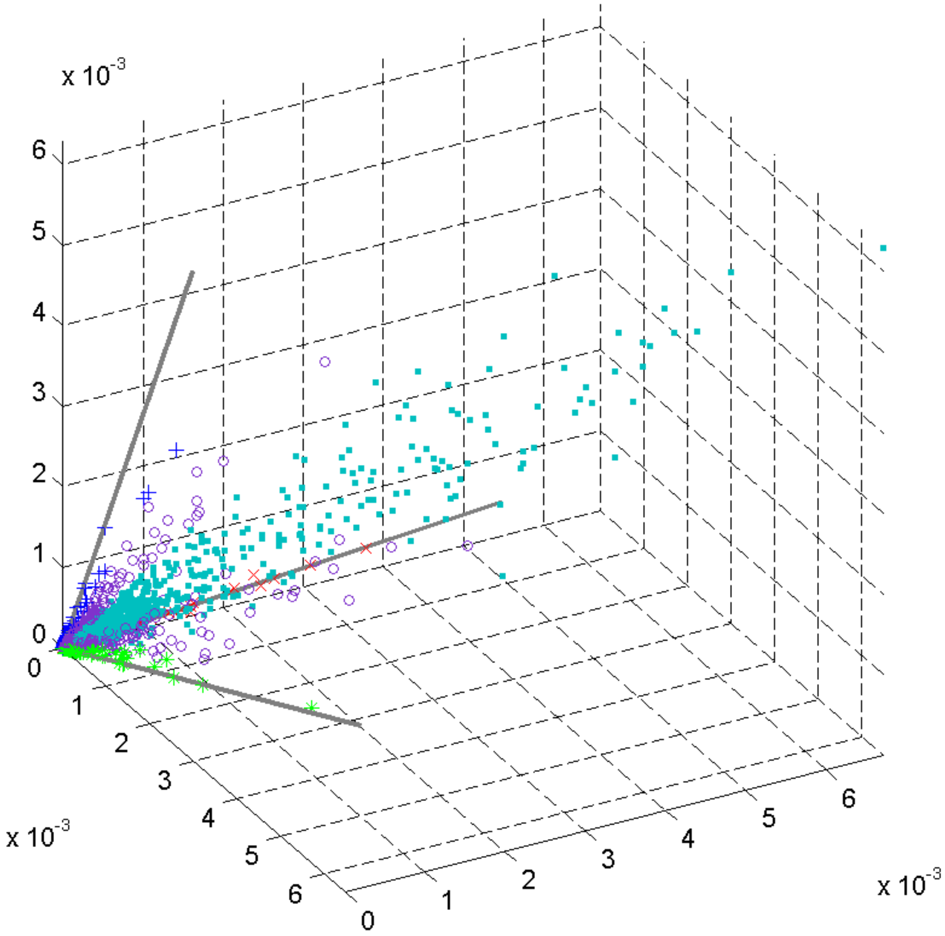


Figure 4.2 Scatter plot of the three mixtures generated by mixing the three gene expression profiles. Green star markers indicate PSGs of ovarian serous adenocarcinoma. Red cross markers indicate PSGs of ovarian mucinous adenocarcinoma. Blue plus signs indicate PSGs of ovarian endometrioid adenocarcinoma.

PCGs are indicated by cyan square markers. The purple circles indicate the intermediate genes between PCGs and PSGs. The grey lines indicate the mixing matrix column vectors.

To better suppress noise in data, CAM utilizes a sector-based clustering on mixture samples in the scatter plot. Sector-based clustering groups the mixture samples into sectors, so that samples within a sector have similar vector directions and samples in different sectors have much more different vector directions. Figure 4.3 is a 2-D illustration of sector-based clustering. Due to the scale ambiguity, vector direction rather than vector norm of the samples is critical for the recovery of mixing matrix. Sector central rays that are sector exemplars indicating the direction of sectors, can be taken as noise-reduced data, based on which the mixing matrix can be estimated. Compared to noise reduction schemes based on linear projections, the sector-based clustering does not reduce the dimensionality of data, while aims to average out noise in almost all dimensions. Keeping all dimensions of the data is preferable when the source number is unknown or cannot be well estimated, because it avoids the possibility of unnecessarily transferring a determined or over-determined problem into an under-determined one that is usually unidentifiable.

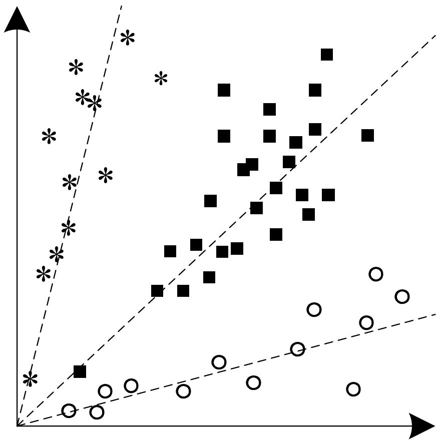


Figure 4.3 2-D illustration of data sectors and sector central rays. Three data sectors are indicated by different markers. The dashed lines indicate the sector central rays.

CAM determines the source number (i.e. model order) through stability analysis. Stability analysis for model order selection has been used in data clustering to determine the cluster number [47, 49]. We apply it to blind source separation by designing a stability analysis scheme and a normalized model instability index that are suitable for the non-negative well-grounded source separation problem. The stability of the separation solution in the presence of data perturbation varies with the number of sources being inferred. For instance, inferring too many sources allows the unnecessary model flexibility to fit random data perturbation and noise, and the solution becomes unstable. Inferring too few sources might lead to unstable solutions since the lack of degree of freedom forces the algorithm to ambiguously mix sources that should be kept separate. Following these considerations, the "correct" source number is defined as that of the separation solution with maximum stability. Selecting a stable model is also consistent with the principle that scientific discovery should be reproducible. Stability analysis based model order selection does not require fitting the data using probabilistic models or any preset subjective threshold, and it is uniformly applicable to the determined, over-determined, and under-determined cases, which also enables CAM to tell whether a source separation task is under-determined in practical use.

Organization of this chapter is as follows. In Section 4.2, we define the notations used in this chapter and review several geometry and convex analysis concepts. In Section 4.3, the noise-free CAM model is introduced and its identifiability is proven, with the identification solution developed. In Section 4.4, the CAM framework applicable to noisy practical problems is presented. In Section 4.5, we evaluate the performance of CAM based on the accuracy of separating numerically mixed gene expression data and compare its performance to those of several benchmark nBSS methods. In Section 4.6 and Section 4.7, we apply CAM on real gene expression data of muscle regeneration and muscular dystrophy to identify underlying active biological processes. Section 4.8 and Section 4.9 are devoted to discussion and conclusion, respectively.

4.2 Geometry and Convex Analysis Concepts

For the ease of mathematical description, we define the following notations.

R, R^P, and $R^{P \times Q}$ Set of real numbers, P-dimensional vectors, and P×Q matrices.

R_+, R_+^P, and $\mathrm{R}_+^{P\times Q}$ Set of non-negative real numbers, P-dimensional vectors, and $P\times Q$ matrices.

$\mathbf{1}_P$ and $\mathbf{1}_{P\times Q}$ All one P-dimensional vector and $P\times Q$ matrix.

$\mathbf{0}_P$ and $\mathbf{0}_{P\times Q}$ All zero P-dimensional vector and $P\times Q$ matrix.

$\mathbf{I}_P$ $P\times P$ identity matrix.

$\|\bullet\|$ Euclidean norm of a vector.

$\lfloor\bullet\rfloor$ The largest integer less than or equal to the input number.

$\mathbf{e}_i$ Vector with the ith entry equal to 1 and all other entries equal to 0.

^{T} Superscript indicating the transpose of a vector or matrix.

$_{(n_1, \cdots, n_Q)}$ For a matrix, this subscript indicates the matrix obtained by removing the n_1th, ···, and n_Qth columns of the original matrix; for a vector, this subscript indicates the vector obtained by removing the n_1th, ···, and n_Qth entries of the original vector.

$\{\bullet\}$ Set. If the input argument is a matrix, it denotes the vector set formed by the column vectors of the matrix.

$\angle(\bullet, \bullet)$ If the input arguments are two vectors of equal size, this function indicates the angle between the two vectors; if the input arguments are two matrices of equal size, it denotes the minimum average angle between column vectors of the two matrices. For example, $\mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_Q] \in \mathrm{R}^{P\times Q}$ and $\mathbf{W} = [\mathbf{w}_1 \cdots \mathbf{w}_Q] \in \mathrm{R}^{P\times Q}$, $\angle(\mathbf{U}, \mathbf{W})$ is calculated by

$$\angle(\mathbf{U},\mathbf{W}) = \min_{\phi\in\Phi_Q} \frac{1}{Q} \sum_{q=1}^{Q} \angle\left(\mathbf{u}_q, \mathbf{w}_{\phi(q)}\right), \tag{4.2}$$

where Φ_Q is the set including all permutations of $\{1, \cdots, Q\}$. Since we do not know the association between column vectors of $\mathbf{U}$ and column vectors of $\mathbf{W}$, we need to search through all possible associations to find the optimal one. Using the Hungarian method, the complexity of this search is $O\left(Q^3\right)$ [99].

In this chapter, bold capital letters indicate matrices. Bold lowercase letters indicate vectors. If without specification, vectors in this chapter refer to non-zero vectors, i.e. vectors with a non-zero Euclidean norm. If a variable appears on both sides of an equation, the one on the right side of the equation indicates the variable value before executing the equation, while the one on the left side of the equation indicates the updated variable value after executing the equation. For example, $\boldsymbol{\gamma} = [1\ 1\ 1]^T$ and $\boldsymbol{\upsilon} = [2\ 2\ 2]^T$, and we execute the equation of $\boldsymbol{\gamma} = \boldsymbol{\gamma} + \boldsymbol{\upsilon}$. Then $\boldsymbol{\gamma}$ on the right side of the equation is equal to $[1\ 1\ 1]^T$, and $\boldsymbol{\gamma}$ on the left side of the equation is equal to $[3\ 3\ 3]^T$.

The following geometry and convex analysis concepts will be used for developing CAM methodology.

Definition 4.1 Finite Cone. A finite cone generated by the Q column vectors of matrix $\mathbf{B}$ ($\mathbf{B} = [\mathbf{b}_1 \cdots \mathbf{b}_Q]$) is

$$\text{cone}\left(\{\mathbf{B}\}\right) = \left\{\mathbf{y} \middle| \mathbf{y} = \mathbf{B}\boldsymbol{\alpha},\ \boldsymbol{\alpha} \in \mathrm{R}_+^Q\right\}. \tag{4.3}$$

Definition 4.2 Edge of a finite cone. For cone($\{\mathbf{B}\}$) ($\mathbf{B} = [\mathbf{b}_1 \cdots \mathbf{b}_Q]$), a vector $\mathbf{v}$ is an edge of cone($\{\mathbf{B}\}$), if $\mathbf{v} \in$ cone($\{\mathbf{B}\}$) (i.e. $\mathbf{v} = \mathbf{B}\boldsymbol{\alpha}$, $\boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_Q]^T \in \mathrm{R}_+^Q$) and $\mathbf{v}$ can only be a trivial non-negative combination of $\{\mathbf{B}\}$. A trivial non-negative combination means that any $\mathbf{b}_q$ ($q \in \{1, \cdots, Q\}$) with $\alpha_q > 0$ is a positive scaling of $\mathbf{v}$.

Definition 4.3 Projection Image obtained by projecting a vector point onto a finite cone. For cone($\{\mathbf{B}\}$) ($\mathbf{B} \in \mathrm{R}^{P\times Q}$) and a vector $\mathbf{v}$, the projection image of $\mathbf{v}$ onto cone($\{\mathbf{B}\}$) is

$$\mathbf{v}' = \underset{\mathbf{c}\in\text{cone}(\{\mathbf{B}\})}{\arg\min} \left\|\mathbf{v} - \mathbf{c}\right\|. \tag{4.4}$$

Obviously, if $\mathbf{v} \in$ cone($\{\mathbf{B}\}$), then $\mathbf{v}' = \mathbf{v}$ and $\angle(\mathbf{v}, \mathbf{v}') = 0$; if $\mathbf{v} \notin$ cone($\{\mathbf{B}\}$), then $\mathbf{v}' \neq \mathbf{v}$ and $\angle(\mathbf{v}, \mathbf{v}') > 0$**.** The optimization problem in Equation (4.4) is a second order cone programming problem that can be solved by existing algorithms with a worst-case running time of $O\left(Q^2P\right)$ [128, 129].

Figure 4.4 gives an illustration of the three geometry and convex analysis concepts reviewed above. By the definition of finite cone, all vectors belonging to the cone are confined within the cone.

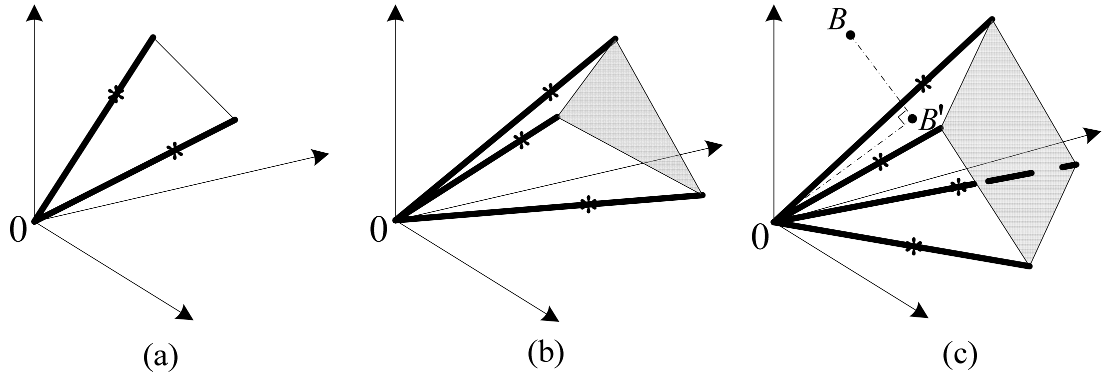


Figure 4.4 Illustration of convex analysis concepts. (a), (b), and (c) present a cone generated by two edges, three edges, and four edges, respectively, in 3-D space. Lines with an arrow are the coordinates. Bold lines are edges. Fine line and fine lines with a grey interior indicate the cross-section of the cone. The cross-section of the cone in (a), (b), and (c) is a line segment, triangle, and quadrangle, respectively. The star markers are points on the edges. In (c), *B* is a point outside of the cone. Its projection image on the cone is *B*'.

4.3 Noise-free CAM Model and Its Identifiability

4.3.1 Assumptions of CAM Model

The mathematical form of the linear mixture model is **X** = **AS**, where $\mathbf{X} \in \mathrm{R}^{M\times N}$ is the observed data matrix containing *M* observed mixtures and *N* samples (i.e. data instances), $\mathbf{A} \in \mathrm{R}^{M\times K}$ is the unknown mixing matrix, and $\mathbf{S} \in \mathrm{R}_{+}^{K\times N}$ is the unknown non-negative data matrix containing *K* sources. The vector form of the linear mixture model is

$$\mathbf{x}[n] = \mathbf{A}\mathbf{s}[n], \quad n = 1, \cdots, N \tag{4.5}$$

where $\mathbf{x}[n] = [x_1[n] \cdots x_M[n]]^T$ is the *n*th mixture sample, $\mathbf{s}[n] = [s_1[n] \cdots s_K[n]]^T$ is the *n*th source sample, and $\mathbf{A} = [\mathbf{a}_1 \cdots \mathbf{a}_K]$. Note that in the context of gene expression dissection column vectors of **X** and **S**, i.e. **x**[*n*] and **s**[*n*], contain the expression values of a gene, but we call **x**[*n*] or **s**[*n*] a mixture sample or a source sample, solely to accommodate the traditions of BSS research community, since $\mathbf{x}[n]$ and $\mathbf{s}[n]$ are data instances. We assume that any vector in $\{\mathbf{A}\}$ is not a positive scaling of other vectors in $\{\mathbf{A}\}$, otherwise the model is degenerated. The CAM model is based on the following assumptions.

(A1) $\mathbf{S} \in \mathrm{R}_{+}^{K\times N}$ and rank($\mathbf{S}$) = K.

(A2) All sources are well-grounded, i.e. $\forall\ i \in \{1, \cdots, K\}$, there exists at least one (unknown) sample index n_i such that $\mathbf{s}[n_i]$ is a positive scaling of $\mathbf{e}_i$. $\mathbf{x}[n_i]$ is the WGP of source i and obviously it is a positive scaling of $\mathbf{a}_i$.

(A3) $\mathbf{A}$ has a full column rank, i.e. rank($\mathbf{A}$) = K.

(A4) Any vector in $\{\mathbf{A}\}$ is not a non-negative or non-positive linear combination of other vectors in $\{\mathbf{A}\}$, which can be formulated as

$$\forall i \in \{1, \cdots, K\}, \quad \mathbf{a}_i \notin \text{cone}\left(\left\{\mathbf{A}_{(i)}\right\}\right) \quad \text{and} \quad \mathbf{a}_i \notin \text{cone}\left(\left\{-\mathbf{A}_{(i)}\right\}\right) = -\text{cone}\left(\left\{\mathbf{A}_{(i)}\right\}\right). \quad (4.6)$$

Let us discuss the practicability of these assumptions. (A1) is naturally satisfied in many applications, such as gene expression dissection. (A2) serves as a good approximation to the high contrast in sources. In the context of gene expression dissection, WGPs represent PSGs that are highly expressed in one biological process but insignificantly or even not expressed in all other biological processes. (A3) is a standard assumption in nBSS problems. (A4) will later be shown critical for the identifiability of mixing matrix. It shall be noted that (A3) is a sufficient but not necessary condition for (A4) to hold.

4.3.2 Model Identifiability

The identifiability of the model is shown by the following theorems. Lemma 4.1 is about the critical role of (A4).

Lemma 4.1 $\{\mathbf{A}\}$ are the unique edges of cone($\{\mathbf{A}\}$) up to a positive scaling, if and only if (A4) is satisfied.

The proof of Lemma 4.1 can be found in Appendix A. Here, we use Figure 4.4 and Figure 4.5 to illustrate the main idea of Lemma 4.1. In Figure 4.4a, b, and c, let $\{\mathbf{A}\}$

be formed by the bold lines in the figure. Apparently, (*A*4) is satisfied. Consequently, we see that in these three figures {**A**} indeed constitute the unique edges of cone({**A**}), i.e. there are no other edges of cone({**A**}) besides {**A**}. In Figure 4.5, we use $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \mathbf{a}_3] \in R^{2\times 3}$ to illustrate what will happen if (*A*4) is not satisfied. Figure 4.5a is the case that $\mathbf{a}_3 \in$ cone({$\mathbf{A}_{(3)}$}). In this case, cone({$\mathbf{A}_{(3)}$}) = cone({**A**}) and $\mathbf{a}_3$ is not an edge of cone({**A**}). Figure 4.5b is the case that $\mathbf{a}_3 \in -$cone({$\mathbf{A}_{(3)}$}). cone({**A**}) occupies the whole 2-D space and there is no real lateral edge of cone({**A**}). Summarizing Figure 4.5a and b, we can see that if (*A*4) is not satisfied, at least one of {**A**} is not an edge of cone({**A**}), which means that (*A*4) being satisfied is necessary for {**A**} to be the edges of cone({**A**}).

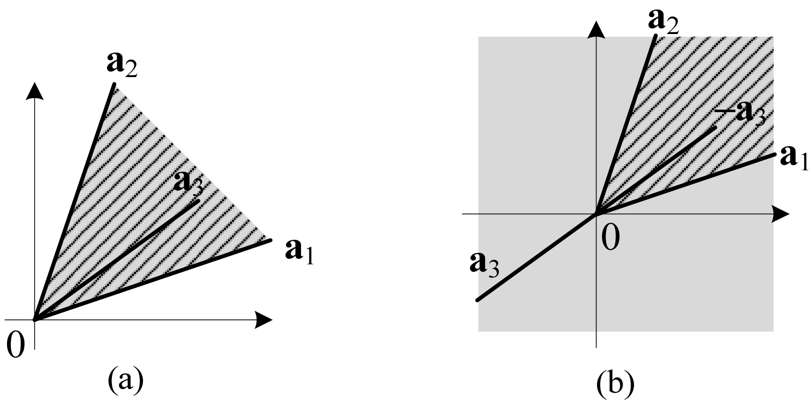


Figure 4.5 Illustration of cases that (*A*4) is not satisfied. (a) The case that $\mathbf{a}_3 \in$ cone({$\mathbf{a}_1$, $\mathbf{a}_2$}). (b) The case that $\mathbf{a}_3 \in -$cone({$\mathbf{a}_1$, $\mathbf{a}_2$}). In (a) and (b), the grey area is cone({$\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$}) and the shadow area is cone({$\mathbf{a}_1$, $\mathbf{a}_2$}).

Since only **X** contains the observed data, based on which **A** will be estimated, we need to study the relationship between **X** and **A**, which leads to the following lemma.

Lemma 4.2 For a linear mixture model **X** = **AS** that satisfies (*A*1) and (*A*2), we have cone({**X**}) = cone({**A**}).

The proof of Lemma 4.2 is given in Appendix B. (*A*2) guarantees the existence of WGPs. The following corollary explicitly states that cone({**X**}) is actually determined only by WGPs.

Corollary 4.1 For a linear mixture model **X** = **AS** that satisfies (*A*1) and (*A*2), removal of non-WGPs $\mathbf{x}[n_1]$, ···, $\mathbf{x}[n_Q]$ from {**X**} will not affect the cone generated by the mixture samples, i.e. cone($\{\mathbf{X}_{(n_1, \cdots, n_Q)}\}$) = cone({**X**}).

Proof: It can be seen that if $\mathbf{x}[n_1]$, ···, $\mathbf{x}[n_Q]$ are non-WGPs, the linear mixture model $\mathbf{X}_{(n_1, \cdots, n_Q)} = \mathbf{AS}_{(n_1, \cdots, n_Q)}$ also satisfies (*A*1) and (*A*2). Thus, according to Lemma 4.2, cone($\{\mathbf{X}_{(n_1, \cdots, n_Q)}\}$) = cone({**A**}) = cone({**X**}).

In Figure 4.4, let {**A**} be formed by the bold line edges. WGPs are the samples on the edges and indicated by the star markers. We can see that WGPs are the most outside, extreme mixture samples in terms of vector direction, and all mixture samples should be confined within cone({**A**}), which is also cone({**X**}). From Lemma 4.1 and Lemma 4.2, we can have the following theorem.

Theorem 4.1 For a linear mixture model **X** = **AS** that satisfies (*A*1), (*A*2), and (*A*4), {**A**} are the unique edges of cone({**X**}) up to a positive scaling.

Proof: Theorem 4.1 is a direct inference from Lemma 4.1 and Lemma 4.2.

Theorem 4.1 indicates that if (*A*1), (*A*2), and (*A*4) hold, **A** can be recovered by identifying edges of cone({**X**}) and that the solution of **A** is unique up to a permutation and a positive scaling of the column vectors. Since whether a vector is an edge or not only depends on its direction, for the simplicity of discussion, we assume that all mixture samples in {**X**} have different directions, i.e. any mixture sample in {**X**} is not a positive scaling of another mixture sample in {**X**}. The uniqueness of vector direction can be easily guaranteed by keeping only one of the mixture samples whose directions are the same and removing the redundant ones. Such a removal will not affect the detection of edges. The following theorem shows how the edges of cone({**X**}) can be identified.

Theorem 4.2 A linear mixture model **X** = **AS** satisfies (*A*1) and (*A*4). Each mixture sample in {**X**} has a unique vector direction. Denote the projection image of $\mathbf{x}[n]$ ($n \in$

$\{1, \dots, N\}$) onto cone($\{\mathbf{X}_{(n)}\}$) by $\mathbf{x}'[n]$. $\mathbf{x}[n]$ is an edge of cone($\{\mathbf{X}\}$), if and only if $\angle(\mathbf{x}[n], \mathbf{x}'[n]) > 0$.

The proof of Theorem 4.2 is given in Appendix C. Theorem 4.2 suggests a test to assert whether a mixture sample is an edge of cone($\{\mathbf{X}\}$) or not. This test needs to be performed one-by-one for each mixture sample to find all the edges. Based on Theorem 4.2, we develop an edge detection algorithm given in Table 4.1. According to Corollary 4.1, when (A1) and (A2) hold, if non-WGPs are removed from $\{\mathbf{X}\}$, the cone generated by the remaining mixture samples does not change. So in the one-by-one edge detection process non-WGPs can be removed once they are identified, while the edge detection result will not be affected by the removal. Note that the edge detection algorithm will detect all of the K edges without knowing K a priori, which means that the source number K is automatically detected in the edge detection process. Also note that Theorem 4.2 only requires (A4) not (A3), which gives the potential for the edge detection algorithm to work in under-determined cases. The worst-case computational complexity of the edge detection algorithm is $O\left(N^3 M\right)$. As we have reviewed in Sub-section 4.1.1, some geometric nBSS methods detect the WGPs using the perspective projection and conventional convex hull algorithms whose worst-case computational complexity is $O(N^{\lfloor (M+1)/2 \rfloor})$ [121]. Clearly, on relatively high-dimensional data (e.g. $M \geq 7$), where the algorithm complexity becomes critical for the application, the proposed edge detection algorithm is much faster than conventional convex hull algorithms.

Table 4.1 Edge Detection Algorithm

Given	Remove redundant mixture samples from $\{\mathbf{X}\}$, so that all remaining mixture samples in $\{\mathbf{X}\}$ have different vector directions. (A1), (A2), and (A4) hold. Index $n = 1$. Threshold τ_e is set to a very small positive number (e.g. 0.001) to tolerate the optimization imprecision that occurs in calculating projecting images.
Step 1	Calculate the projection image $\mathbf{x}'[n]$ resulted from projecting $\mathbf{x}[n]$ onto cone($\{\mathbf{X}_{(n)}\}$).
Step 2	If $\angle(\mathbf{x}[n], \mathbf{x}'[n]) > \tau_e$, $n = n + 1$; otherwise remove $\mathbf{x}[n]$ from $\{\mathbf{X}\}$.

Step 3	If n is bigger than the current size of $\{\mathbf{X}\}$, the algorithm ends and the resulting $\{\mathbf{X}\}$ only contains the K edges; otherwise, go to step 1.

Now we can summarize the condition for identifiability of **A** via the following remark.

Remark 4.1 If ($A1$), ($A2$), and ($A4$) hold, **A** can be identified up to a permutation and a positive scaling of column vectors through the proposed edge detection algorithm.

After study the identifiability of **A**, we can discuss the identifiability of both **A** and **S** through the following theorem.

Theorem 4.3 For a linear mixture model $\mathbf{X} = \mathbf{AS}$ that satisfies, ($A1$), ($A2$), and ($A3$), **A** and **S** are identifiable up to a permutation and a positive scaling. **A** can be identified through the proposed edge detection algorithm, and **S** can be identified by

$$\mathbf{S} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{X}\,. \tag{4.7}$$

The proof of Theorem 4.3 is given in Appendix D. An interesting question is whether the solution for **S** is unique when ($A3$) is not satisfied. ($A3$) being violated means that $\exists\ \boldsymbol{\beta} \in \mathrm{R}^K$, $\boldsymbol{\beta} \neq \mathbf{0}_K$ and $\mathbf{A}\boldsymbol{\beta} = \mathbf{0}_M$. Let $|\beta|_{\max} = \max\{|\beta_1|, \ldots, |\beta_K|\}$, and apparently $|\beta|_{\max} > 0$. Suppose that there exists a source sample $\mathbf{s}[n]$ whose entries are all bigger than 0 and $\mathbf{x}[n] = \mathbf{As}[n] \neq \mathbf{0}_M$. $\mathbf{s}[n]$ is not a linear scaling of $\boldsymbol{\beta}$, otherwise $\mathbf{x}[n] = \mathbf{As}[n] = \mathbf{0}_M$. Let $s[n]_{\min} = \min\{s_1[n], \ldots, s_K[n]\}$, and apparently $s[n]_{\min} > 0$. $\mathbf{x}[n] = \mathbf{As}[n] = \mathbf{As}[n] + (s[n]_{\min}/|\beta|_{\max})\mathbf{A}\boldsymbol{\beta} = \mathbf{A}(\mathbf{s}[n] + s[n]_{\min}\boldsymbol{\beta}/|\beta|_{\max})$, where $\mathbf{s}[n] + s[n]_{\min}\boldsymbol{\beta}/|\beta|_{\max} \in \mathrm{R}_+^K$ and $\mathbf{s}[n] + s[n]_{\min}\boldsymbol{\beta}/|\beta|_{\max}$ is not a linear scaling of $\mathbf{s}[n]$ due to that $\mathbf{s}[n]$ is not a linear scaling of $\boldsymbol{\beta}$. This shows that the nth source sample can be two different non-negative vectors and one vector is not a linear scaling of the other. Therefore, the solution for **S** is not unique, as long as at least one mixture sample is a mixture of all sources, which usually occurs in data.

Remark 4.2 When (A1), (A2), and (A4) hold, but (A3) does not hold, usually we can only identify **A** up to ambiguity of permutation and positive scaling, but cannot identify **S**.

In many applications, the relative proportions of sources in the mixtures are of interest. For example, in dissection of gene expressions for identification of underlying biological processes, these relative proportions actually indicate the relative activity levels of the biological processes in different biological conditions or at successive time points. To make the entries of **A** relative proportions, each row of **A** needs to be normalized to have a unit summation, i.e. $\mathbf{A1}_K = \mathbf{1}_M$, which can be ensured by scaling all sources and mixtures to have a unit summation, i.e. $\mathbf{X1}_N = \mathbf{1}_M$ and $\mathbf{S1}_N = \mathbf{1}_K$ [22], because if $\mathbf{X1}_N = \mathbf{1}_M$ and $\mathbf{S1}_N = \mathbf{1}_K$, then $\mathbf{1}_M = \mathbf{X1}_N = \mathbf{AS1}_N = \mathbf{A1}_K$. So even if the original **A** does not satisfy $\mathbf{A1}_K = \mathbf{1}_M$, we can scale the mixtures and sources to meet this requirement through the following way proposed in [22]

$$\begin{gathered}\mathbf{A} = \left(\operatorname{diag}\left(\mathbf{X1}_N\right)\right)^{-1}\mathbf{A}\operatorname{diag}\left(\mathbf{S1}_N\right)\\ \mathbf{S} = \left(\operatorname{diag}\left(\mathbf{S1}_N\right)\right)^{-1}\mathbf{S}\end{gathered} \tag{4.8}$$

and

$$\mathbf{X} = \left(\operatorname{diag}\left(\mathbf{X1}_N\right)\right)^{-1}\mathbf{X}\,, \tag{4.9}$$

where diag(•) returns a diagonal matrix whose diagonal entries come from the input vector, **X**, **S**, and **A** on the right side of the equations indicate the mixtures, sources, and mixing matrix before scaling, respectively, and **X**, **S**, and **A** on the left side of the equations indicate the mixtures, sources, and mixing matrix after scaling, respectively. Equation (4.8) gives a way, through which we can scale the identified **S** and **A** to remove the scale ambiguity of separation solution, and Equation (4.9) indicates a data preprocessing step that we need to perform before identifying the mixing matrix and sources. In the context of gene expression dissection, requiring all sources to have an equal summation is reasonable, because PCGs, the majority of the genes, have similar expression values in all biological processes, as we have discussed before.

4.3.3 Estimation of S for Gene Expression Dissection When (*A*3) Does Not Hold

The gene expression patterns of biological processes have unique characteristics that a majority of the genes are PCGs that are similarly expressed in all biological processes and that a small portion of the genes are PSGs that are exclusively highly expressed in only one biological process. Based on these characteristics, we may still be able to estimate **S** with certain accuracy, when (*A*3) does not hold. Suppose that (*A*1), (*A*2), and (*A*4) hold, but (*A*3) does not hold, and that we have scaled **X** to satisfy $\mathbf{X1}_N = \mathbf{1}_M$ and already identified **A** with positive scaling and permutation ambiguity using the edge detection algorithm proposed in Table 4.1. Table 4.2 shows an iterative algorithm to estimate **S** and scale both **A** and **S**. The following explains each step of the algorithm in details.

Table 4.2 Iterative Algorithm for Estimating **S** When (*A*3) Does Not Hold

Given	**X** and **A**. $\mathbf{X1}_N = \mathbf{1}_M$. Each **A** column vector is normalized to have a unit norm. $\tau_{\mathbf{A}}$ is a small positive number, e.g. 0.01, used as a threshold for the judgment of algorithm convergence.
Step 1	Estimate **S** by maximizing all genes' tendency to be a typical PCG whose source vector has all equal entries.
Step 2	Based on the **S** estimated in step 1, identify the PSGs.
Step 3	For each identified PSG, update its source vector by maximizing its tendency to be dominated by only one source.
Step 4.	If $\|\|\mathbf{A1}_K - \mathbf{1}_M\|\|$ is smaller than $\tau_{\mathbf{A}}$, the algorithm converges and stops; otherwise scale **A** using Equation (4.15), and then go to step 1.

Step 1 Because the number of PCGs is much large than that of PSGs, we first maximize all genes' trend to be a typical PCG whose source vector has all entries equal. For the *n*th gene, maximization of the gene's trend to be a typical PCG can be mathematically formulated as

$$\begin{aligned} \mathbf{s}[n] &= \arg\min_{\xi} \left\| \frac{\xi}{\mathbf{1}_K^T \xi} - \frac{\mathbf{1}_K}{K} \right\|. \\ \text{s.t.} \quad \mathbf{x}[n] &= \mathbf{A}\xi, \quad \xi \in \mathrm{R}_+^K \end{aligned} \tag{4.10}$$

This optimization problem is equivalent to the following second order cone problem

$$\begin{aligned} &\min_{\boldsymbol{\omega}} \left\| \begin{bmatrix} \mathbf{0}_{K\times(K+1)} & \mathbf{I}_K \end{bmatrix} \boldsymbol{\omega} \right\| \\ &\text{s.t.} \quad \begin{bmatrix} \mathbf{A} & -\mathbf{x}[n] & \mathbf{0}_{M\times K} \\ \mathbf{1}_K^T & 0 & \mathbf{0}_K^T \\ \mathbf{I}_K & \mathbf{0}_K & -\mathbf{I}_K \end{bmatrix} \boldsymbol{\omega} = \begin{bmatrix} \mathbf{0}_M \\ 1 \\ \mathbf{1}_K / K \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathbf{I}_{K+1} & \mathbf{0}_{(K+1)\times K} \end{bmatrix} \boldsymbol{\omega} \in \mathrm{R}_+^{K+1} \end{aligned}, \tag{4.11}$$

which can be solved by existing algorithm and software [128]. By solving this second order cone problem, we obtain the optimal $\boldsymbol{\omega}$, denoted by $\boldsymbol{\omega}^*$. Then $\mathbf{s}[n]$ can be calculated according to its relationship with $\boldsymbol{\omega}^*$, which is $\boldsymbol{\omega}^* = \begin{bmatrix} \frac{\mathbf{s}[n]^T}{\mathbf{1}^T\mathbf{s}[n]} & \frac{1}{\mathbf{1}^T\mathbf{s}[n]} & \frac{\mathbf{s}[n]^T}{\mathbf{1}^T\mathbf{s}[n]} - \frac{\mathbf{1}_K^T}{K} \end{bmatrix}^T$.

Step 2 Based on the **S** estimated in step 1, identify the PSGs. PSGs are a small group of genes whose source vectors have the least trend to have all entries equal. Based on this fact, we identify PSGs through the following scheme. For each gene n, we calculate $\angle(\mathbf{s}[n], \mathbf{1}_K)$, where $\mathbf{s}[n]$ is obtained in step 1. Then we cluster the genes based on the obtained angles, which actually is a 1-D clustering task that can be performed by many existing clustering methods. In our implementation, we fit a 1-D SFNM model to the 1-D angle data using the EM algorithm [73]. The model fitting process follows the iterative procedure proposed in [76] and has been introduced in detail in the fifth paragraph in Section 2.3 and Figure 2.2, so we do not re-elaborate it here. Once the SFNM model is trained, cluster memberships can be determined by the MAP principle [40], i.e. each gene is assigned to the cluster that it most likely belongs to evaluated by its posterior probabilities of belonging to different clusters. The gene cluster with the biggest average angle value is considered to be formed by PSGs. We also set a parameter N_{minPSG}, which is the minimum number of PSGs per source (i.e. biological process). If the number of identified PSGs is smaller than KN_{minPSG}, we include one more gene cluster whose average angle value is the biggest among the rest of the gene clusters, until the total PSG number is not smaller than KN_{minPSG}.

Step 3 For each identified PSG, update its source vector by maximizing its trend to be dominated by only one source. Let n be the index of a PSG, we need to perform the following optimization

$$\begin{aligned} &\mathbf{s}[n] = \arg\max_{\boldsymbol{\varphi}} \left(\frac{\max(\varphi_1, \ldots, \varphi_K)}{\mathbf{1}_K^T \boldsymbol{\varphi}} \right) \\ &\text{s.t.} \quad \mathbf{x}[n] = \mathbf{A}\boldsymbol{\varphi}, \quad \boldsymbol{\varphi} = \begin{bmatrix} \varphi_1 & \cdots & \varphi_K \end{bmatrix}^T \in \mathrm{R}_+^K \end{aligned} . \tag{4.12}$$

The above maximization problem is equivalent to take the maximum value of K individual linear-fractional programming problems, given by

$$\max \left(\begin{array}{c} \max\limits_{\boldsymbol{\varphi}} \dfrac{\varphi_1}{\mathbf{1}_K^T \boldsymbol{\varphi}} \\ \text{s.t.} \quad \mathbf{x}[n] = \mathbf{A}\boldsymbol{\varphi}, \quad \boldsymbol{\varphi} \in \mathrm{R}_+^K \end{array}, \ \cdots, \begin{array}{c} \max\limits_{\boldsymbol{\varphi}} \dfrac{\varphi_K}{\mathbf{1}_K^T \boldsymbol{\varphi}} \\ \text{s.t.} \quad \mathbf{x}[n] = \mathbf{A}\boldsymbol{\varphi}, \quad \boldsymbol{\varphi} \in \mathrm{R}_+^K \end{array} \right). \tag{4.13}$$

Each linear-fractional programming problem in Equation (4.13) can be converted to a linear programming problem that can be well solved by existing algorithm and software [128]. Take the ith ($i \in \{1, \ldots, K\}$) linear-fractional programming problem as an example, it can be converted to

$$\begin{aligned} &\max_{\boldsymbol{\lambda}} \begin{bmatrix} \mathbf{e}_i^T & 0 \end{bmatrix} \boldsymbol{\lambda} \\ &\text{s.t.} \quad \begin{bmatrix} \mathbf{A} & -\mathbf{x}[n] \\ \mathbf{1}_K^T & 0 \end{bmatrix} \boldsymbol{\lambda} = \begin{bmatrix} \mathbf{0}_M \\ 1 \end{bmatrix}, \quad \boldsymbol{\lambda} \in \mathrm{R}_+^{K+1} \end{aligned} , \tag{4.14}$$

where the relation between $\boldsymbol{\lambda}$ and $\boldsymbol{\varphi}$ is $\boldsymbol{\lambda} = \begin{bmatrix} \dfrac{\boldsymbol{\varphi}^T}{\mathbf{1}_K^T \boldsymbol{\varphi}} & \dfrac{1}{\mathbf{1}_K^T \boldsymbol{\varphi}} \end{bmatrix}^T$.

Step 4 If $\|\mathbf{A1} - \mathbf{1}\|$ is smaller than $\tau_{\mathbf{A}}$, the algorithm stops; otherwise, scale $\mathbf{A}$ using the following equation

$$\mathbf{A} = \mathbf{A}\left(\eta\left(\mathrm{diag}\left(\mathbf{S1}_N\right) - \mathbf{I}_K\right) + \mathbf{I}_K\right) \tag{4.15}$$

where $\mathbf{S}$ is obtained through step 1, 2, and 3, and η is a learning rate in the range of (0, 1) introduced for the purpose of conducting stepwise updates to avoid numerical oscillation in the learning process, and after executing Equation (4.15) the algorithm goes to step 1. Equation (4.15) scales $\mathbf{A}$ column vectors according to the summations of sources, i.e. if the summation of a source is bigger than 1, the norm of the corresponding $\mathbf{A}$ column vector is amplified, and if the summation of a source is smaller than 1, the norm of the corresponding **A** column vector is reduced.

4.3.4 Validation of CAM Principle

To validate the principle of CAM presented in Sub-sections 4.3.1, 4.3.2, and 4.3.3, we evaluate the performance of CAM in a noise-free enviroment based on its ability to separate numerically mixed microarray gene expression data. The performance evaluation is conducted in three cases, i.e. determined case (4 mixtures vs. 4 sources), over-determined case (6 mixtures vs. 4 sources), and under-determined case (3 mixtures vs. 4 sources). In the determined and over-determined cases, (*A*3) is satisfied. In the under-determined case, (*A*3) is not satisfied, but (*A*4) is satisfied. We calculate the recovery accuracies of CAM, and compare them to those of five existing nBSS methods, namely, nICA, SNICA, NMF, SNMF, and N-FINDR, focusing on the determined and over-determined cases. In the under-determined case, we evaluate CAM's performance.

To evaluate how accurate the estimates of sources and mixing matrix are, we propose two performance measures. These performance measures are designed to be scale free, because some nBSS methods involved in the comparison do not take care of the signal scale ambiguity. The performance measures are also designed to be applicable in all of the determined, over-determined, and under-determined cases. A prerequisite of evaluating the recovery accuracies of sources and mixing matrix is the knowledge of the association between true sources and recovered sources, which is also the association between column vectors in mixing matrix **A** and column vectors in the mixing matrix estimate denoted by $\hat{\mathbf{A}}$. This association is determined by calculating $\angle(\mathbf{A}, \hat{\mathbf{A}})$ using the mechanism introduced in Equation (4.2). Let ϕ^* denote the obtained optimal association. The first performance measure is the accuracy of recovering mixing matrix calculated by

$$\text{Accuracy of Recovering Mixing Matrix} = 1 - \frac{1}{\pi}\angle\left(\mathbf{A}, \hat{\mathbf{A}}\right), \qquad (4.16)$$

which is a scalar in the range of [0, 1], and the bigger the accuracy is the better the mixing matrix is recovered. Let $\mathbf{S} = [\mathbf{s}_1 \ldots \mathbf{s}_K]^T$, where $\mathbf{s}_i$ ($i \in \{1, \ldots, K\}$) denotes the *i*th true source. Similarly $\hat{\mathbf{S}} = \left[\hat{\mathbf{s}}_1 \quad \ldots \quad \hat{\mathbf{s}}_K\right]^T$, where $\hat{\mathbf{s}}_i$ ($i \in \{1, \ldots, K\}$) denotes the *i*th recovered source. The accuracy of recovering sources is the average correlation coefficient between the true sources and their corresponding recovered sources, calculated by

$$\text{Accuracy of Recovering Sources} = \frac{1}{K}\sum_{i=1}^{K}\rho\left(\mathbf{s}_i, \hat{\mathbf{s}}_{\phi^*(i)}\right), \tag{4.17}$$

where $\rho(\bullet, \bullet)$ calculates the correlation coefficient between two vectors.

From an ovarian cancer microarray gene expression dataset [104], we select four gene expression profiles from four different ovarian cancer subtypes as ground truth sources. The array IDs of the gene expression profiles are CHTN-OS-102, CHTN-OM-029, CHTN-OE-060, and CHTN-OC-045, where OS refers to ovarian serous adenocarcinomas, OM refers to ovarian mucinous adenocarcinomas, OE refers to ovarian endometrioid adenocarcinomas, and OC refers to ovarian clear cell adenocarcinomas. Each profile contains the expression values of 7069 genes and is scaled to have a unit summation. As expected, each source profile contains phenotype up-regulated (highly expressed) genes that are not or insignificantly expressed in all other source profiles. We examine the data and ensure that each profile has its own WGPs.

The average pair-wise correlation coefficient of the four source profiles is 0.8311, due to the fact that a majority of the genes are PCGs that provide the basic/common cellular structure/function required for most biological conditions and thus are similarly expressed in all source profiles. It shall be recognized that the distinct patterns that set apart different sources are of particular interest and shall be correctly and accurately recovered using nBSS methods. Thus, we proposed a third performance measure that focuses on the accuracy of recovering the distinct patterns of sources for the comparison study. We select 800 source-specifically dominated samples to form the distinct pattern of each source, where the degree of dominance for source i on sample n is calculated by $s_i[n]/(\mathbf{1}_K^T\mathbf{s}[n])$ ($i \in \{1, \ldots, K\}$ and $n \in \{1, \ldots, N\}$). The accuracy of recovering distinct patterns of sources is the average correlation coefficient between the recovered and true distinct patterns of sources, calculated by Equation (4.17) while only over the union of dominated samples of each source.

As aforementioned, the mixture samples will be generated by numerically mixing the source gene expression profiles, where the mixing matrix must be non-negative to assure the applicability of NMF and SNMF, which can only work on non-negative mixture data. In each of the determined, over-determined, and under-determined cases, 100 simulation datasets are generated using 100 different mixing matrices and the same sources profiles. All entries of the mixing matrices are randomly drawn from a uniform distribution in the range of [0 1], and the mixing matrices are scaled to have a unit row sum via $\mathbf{A} = (\mathrm{diag}(\mathbf{A1}_k))^{-1}\mathbf{A}$. In addition, we require that the condition number of mixing matrix should not be bigger than 4 to satisfy assumption ($A3$) for the determined and over-determined cases, and that for $i \in \{1, 2, 3, 4\}$, $\angle(\mathbf{a}_i, \mathbf{a}_i^{'})$ must be bigger than $\pi/7$ to satisfy assumption ($A4$) for the under-determined case, where $\mathbf{a}_i^{'}$ is the projection image of $\mathbf{a}_i$ on $\mathrm{cone}\{\mathbf{A}_{(i)}\}$.

For CAM, in the iterative algorithm for estimating **S** when ($A3$) is not satisfied by the estimated mixing matrix, the parameter N_{minPSG} is set at 100 and the learning rate η is set at 0.5. Among the competing methods, SNMF requires an input parameter that is the weight of the source sparseness measure term in its objective function. We set this parameter at 1. Thus its source sparseness measure and model fitting error measure are equally weighted in the resulted objective function. In the over-determined case, nICA, SNICA, and N-FINDR require dimension reduction from 6 dimensions to 4 dimensions, which is fulfilled by the principal component analysis methods [109, 116, 130].

Table 4.3 Performance Comparison in Noise-free Determined Case

	Average Accuracy of Recovering Mixing Matrix	Average Accuracy of Recovering Sources	Average Accuracy of Recovering Distinct Patterns of Sources
CAM	0.9983	0.9999	0.9998
nICA	0.8555	0.7726	0.8374
SNICA	0.9918	0.9991	0.9962
NMF	0.9569	0.9937	0.9754
SNMF	0.9512	0.9932	0.9730
N-FINDR	1	1	1

We take the average of performance measure over the 100 simulation datasets and show the average value in Table 4.3, Table 4.4, and Table 4.5, for the determined, over-determined, and under-determined cases, respectively. We can see that in the determined and over-determined cases N-FINDR's performance is always perfect. CAM's performance is consistently almost perfect in the determined and over-determined cases. Its recovery accuracies are not exactly 1, only due to the numerical precision of the algorithm. nICA performs worst among the methods due to that its assumption about source uncorrelatedness is not satisfied by the data. The experimental results demonstrate the validity of CAM's principle. Although CAM does not achieve perfect recovery accuracy of 1, it can be used in the under-determined case and without knowledge of source number, which are its significant advantages over N-FINDR. From Table 4.5, we can see that CAM's recovery accuracies are also quite good even in the under-dertermined case.

Table 4.4 Performance Comparison in Noise-free Over-determined Case

	Average Accuracy of Recovering Mixing Matrix	Average Accuracy of Recovering Sources	Average Accuracy of Recovering Distinct Patterns of Sources
CAM	0.9979	0.9999	0.9998
nICA	0.8616	0.7726	0.8374
SNICA	0.9546	0.9944	0.9769
NMF	0.9658	0.9957	0.9840
SNMF	0.9537	0.9936	0.9753
N-FINDR	1	1	1

Table 4.5 Performance Evaluation of CAM in Noise-free Under-determined Case

	Average Accuracy of Recovering Mixing Matrix	Average Accuracy of Recovering Sources	Average Accuracy of Recovering Distinct Patterns of Sources
CAM	0.9995	0.9838	0.9495

4.4 CAM Framework for Real Applications

The identifiability and computational solution of the noise-free CAM model have been studied in the previous section. In real applications, tasks become more complex due to the existence of noise. In this section, we introduce the extended CAM framework applicable for real data analysis. The extended CAM framework includes several major parts, i.e. data preprocessing, noise reduction by sector-based clustering, minimization of model fitting error, recovery of sources, and stability analysis based model order selection.

4.4.1 Data Preprocessing

As we have discussed, one preprocessing step on data is to scale each observed mixture to have a unit summation using Equation (4.9). Another critical step in data preprocessing is to reduce the noise in data. The CAM model with additive noise can be formulated as

$$\mathbf{x}[n] = \mathbf{A}\mathbf{s}[n] + \boldsymbol{\varepsilon}[n], \quad n = 1, \cdots, N \tag{4.18}$$

where $\boldsymbol{\varepsilon}[n]$ is the additive noise on sample n, and is independent of $\mathbf{s}[n]$. We assume that $\boldsymbol{\varepsilon}[n]$ is independently identically distributed (i.i.d.) and that $\boldsymbol{\varepsilon}[n]$ has a mean of $\mathbf{0}_M$ and a covariance matrix of $\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}$. We define Signal to Noise Ratio (SNR) of data as

$$\mathrm{SNR} = \frac{\frac{1}{N}\sum_{n=1}^{N}\left\|\mathbf{A}\mathbf{s}[n]\right\|^2}{\mathrm{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}\right)}, \tag{4.19}$$

where tr(•) is the trace of a matrix. We also define the signal to noise ratio of the *n*th sample as

$$\mathrm{SNR}[n] = \frac{\left\|\mathbf{A}\mathbf{s}[n]\right\|^2}{\left\|\boldsymbol{\varepsilon}[n]\right\|^2}. \tag{4.20}$$

Assume that signal power is much stronger than noise power, i.e. $\|\mathbf{A}\mathbf{s}[n]\| >> \|\boldsymbol{\varepsilon}[n]\|$. Then we ignore the impact of $\boldsymbol{\varepsilon}[n]$ on $\|\mathbf{x}[n]\|$ (i.e. let $\|\mathbf{x}[n]\| \approx \|\mathbf{A}\mathbf{s}[n]\|$) and calculate the expectation of SNR[n] respective to $\boldsymbol{\varepsilon}[n]$ by

$$
\begin{aligned}
\mathrm{E}_{\boldsymbol{\varepsilon}[n]}\left[\mathrm{SNR}[n]\right] &= \mathrm{E}_{\boldsymbol{\varepsilon}[n]}\left[\frac{\left\|\mathbf{As}[n]\right\|^2}{\left\|\boldsymbol{\varepsilon}[n]\right\|^2}\right] \\
&= \left\|\mathbf{As}[n]\right\|^2 \mathrm{E}_{\boldsymbol{\varepsilon}[n]}\left[\frac{1}{\left\|\boldsymbol{\varepsilon}[n]\right\|^2}\right] \\
&\approx \left\|\mathbf{x}[n]\right\|^2 \mathrm{E}_{\boldsymbol{\varepsilon}[n]}\left[\frac{1}{\left\|\boldsymbol{\varepsilon}[n]\right\|^2}\right]
\end{aligned}
\tag{4.21}
$$

where $\mathrm{E}_{\boldsymbol{\varepsilon}[n]}[1/\|\boldsymbol{\varepsilon}[n]\|^2]$ is a constant because $\boldsymbol{\varepsilon}[n]$ is i.i.d.. Equation (4.21) shows that the expectation of SNR[n] is approximately a monotonically increasing function of $\|\mathbf{x}[n]\|$, which indicates that large-norm mixture samples potentially have a higher SNR value than small-norm mixture samples. Therefore, a natural thought for reducing the noise effect is that we exclude some small-norm mixture samples when estimating the mixing matrix. On the other hand, in real applications, sometimes the dataset may contain some very large-norm mixture samples that are outliers. Including these large-norm outliers in estimating the model may also reduce the recovery accuracy, so we can also exclude some large-norm mixture samples in estimating the mixing matrix. Removing some large-norm mixture samples is an optional data preprocessing step and is necessary only when working on datasets that indeed contain large-norm outliers, which can be seen from the histogram that counts the mixture samples with different vector norms or some low-dimensional (e.g. 2-D or 3-D) scatter plots.

Here, we generate a toy dataset to illustrate the effect of removing small-norm samples. The toy dataset contain 1600 3-D mixture samples generated by mixing 3 sources using a mixing matrix of [-0.1 0.5 0.6; 0.6 -0.1 0.5; 0.5 0.6 -0.1]. We purposely choose a mixing matrix containing negative values to show that the CAM model only requires the sources to be non-negative, while the mixing matrix and the observed mixture data can contain negative values. A half of the source samples (i.e. 800 source samples) are drawn from a 3-D exponential distribution with independent variables to ensure that some samples are approximately WGPs. The mean of the exponential distribution is $[1\ 1\ 1]^T$. The other half of the source samples are absolute values of 3-D vectors drawn from a Gaussian distribution with a mean of $[0\ 0\ 0]^T$ and a covariance matrix of [1 0.9 0.9; 0.9 1 0.9; 0.9 0.9 1]. Noise $\boldsymbol{\varepsilon}[n]$ is drawn from a Gaussian distribution with a mean of $[0\ 0\ 0]^T$ and a covariance matrix of [0.07 0 0; 0 0.07 0; 0 0 0.07]. Calculated by Equation (4.19), the toy dataset has an SNR of 12.4dB. Figure 4.6 shows the scatter plot of the toy dataset.

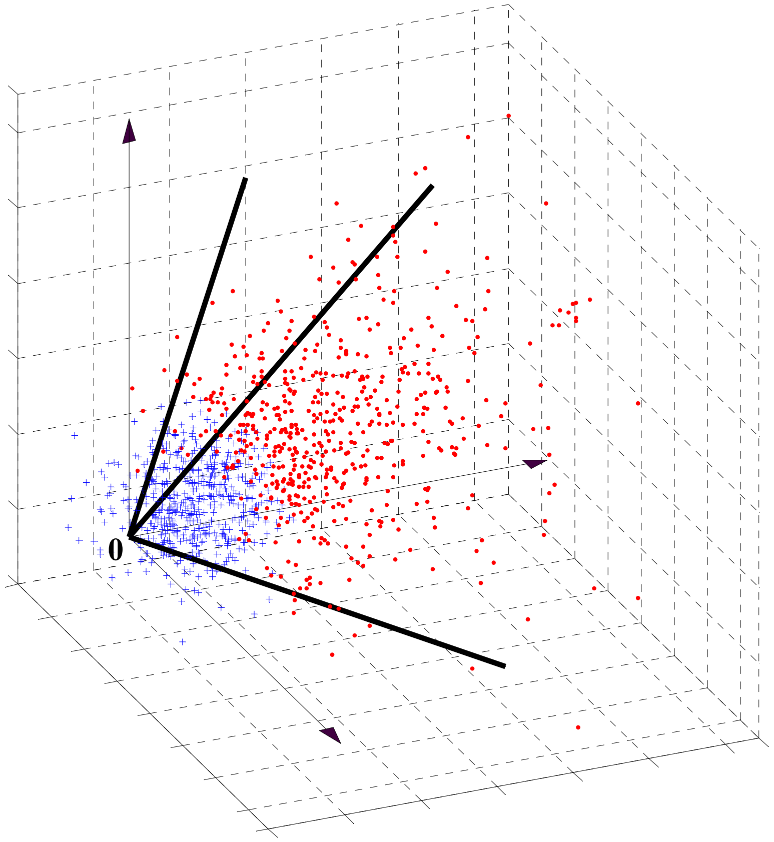


Figure 4.6 Scatter plot of the toy dataset. Lines with arrows are coordinates. Bold lines are mixing matrix column vectors $\{\mathbf{A}\}$ that should be edges of the cone formed by the mixture samples if there is no noise. Blue plus signs are the 800 small-norm samples. Red points are the 800 large-norm samples.

4.4.2 Noise Reduction by Sector-based Clustering

From Figure 4.6, we can see that even after removing the 800 small-norm samples (blue plus signs in Figure 4.6), the remaining 800 large-norm samples (red points in Figure 4.6) still contain significant noise. For example, some of the red points are actually outside of cone($\{\mathbf{A}\}$). Consequently, directly applying the edge detection algorithm on the large-norm samples cannot get an accurate estimate of $\{\mathbf{A}\}$, which is illustrated by Figure 4.7 where we have perspectively projected the large-norm samples onto a plane passing through $[0\ 0\ 1]^T$, $[0\ 1\ 0]^T$, and $[1\ 0\ 0]^T$. Perspective projection is simply positive scaling of samples and illustrated by Figure 4.8. In Figure 4.7, we can see that eleven samples (red circles in Figure 4.7) are identified as edges when we apply the edge detection algorithm on the large-norm samples. Some of the edges are actually far away from the positions of $\{\mathbf{A}\}$ indicated by blue diamonds, and several of the edges may be very similar and close to the same mixing matrix column vector. To further suppress the noise effect in data for estimating $\mathbf{A}$, we use sector-based clustering to average out noise.

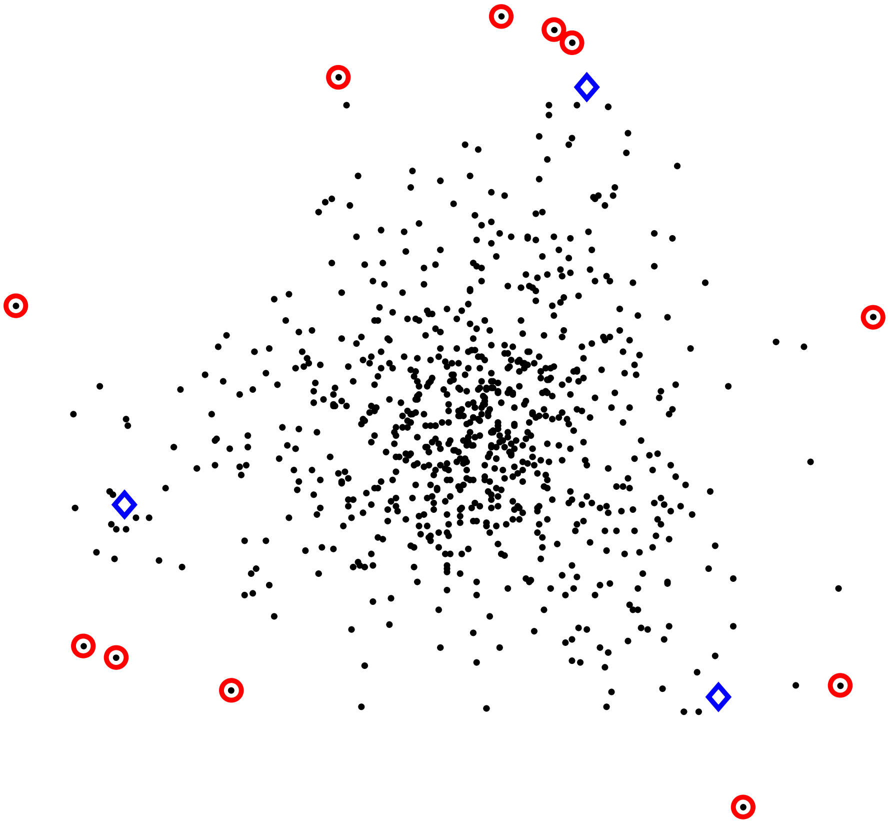


Figure 4.7 Perspective projection of the 800 large-norm samples of the toy dataset. The samples are perspectively projected onto the plane passing through $[1\ 0\ 0]^T$, $[0\ 1\ 0]^T$, and $[0\ 0\ 1]^T$. The black dots are the samples. Red circle markers indicate the edges detected by applying the edge detection algorithm on the samples. Blue diamond markers indicate the position of mixing matrix column vectors.

The sector-based clustering groups the samples into sectors, so that samples within a sector have similar vector directions, while samples in different sectors have much more different vector directions. Each data sector has an exemplar called sector central ray that indicates the sector's direction. Figure 4.3 is an illustration of sector-based clustering in 2-D space. The following is the definition of sector central ray.

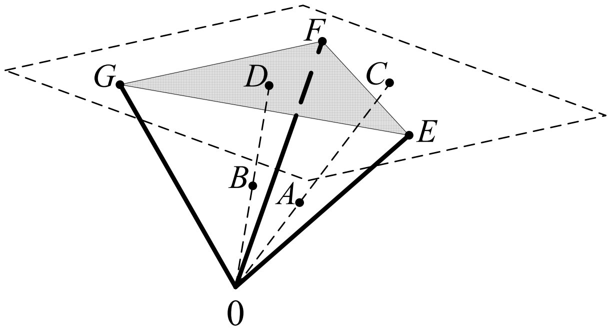


Figure 4.8 Illustration of perspective projection. The quadrangle with dashed border indicates a plane onto which perspective projection will project the samples. The three bold lines are mixing matrix column vectors that are edges of the mixture sample cone if there is no noise. *E*, *F*, and *G* are the perspective projection images of the mixing matrix column vectors. The grey area is the cross-section of the noise-free mixture sample cone in the plane. The perspective projection images of data points *A* and *B* on the plane are *C* and *D*, respectively.

Definition 4.4 **Sector Central Ray**. Given a data sector, the sector central ray minimizes the summation of square distances from the data points to the ray, among all the rays starting from the original.

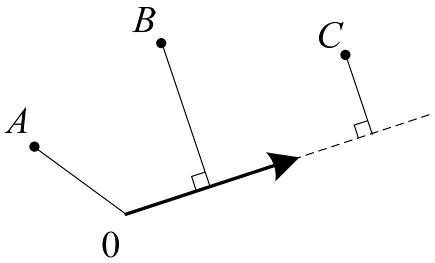


Figure 4.9 Illustration of the distances between data points and a ray. The bold line with an arrow is a unit vector indicating the direction of a ray that starts from the original. The solid fine lines indicate the distances from data points *A*, *B*, and *C* to the ray.

The distance from a data point to a ray is defined as the smallest distance between the data point and any point on the ray, as illustrated by the solid fine lines in Figure 4.9. Let $\{\mathbf{d}_1, \ldots, \mathbf{d}_Q\}$ be a data sector containing Q data points. Let $\mathbf{r}$ denote its sector central ray. Since only the direction of $\mathbf{r}$ is of interest, let $\mathbf{r}$ be a unit vector. We require that $\angle(\mathbf{r}, \mathbf{d}_q) \leq 90°$ for $q \in \{1, \ldots, Q\}$, because if $\angle(\mathbf{r}, \mathbf{d}_q) > 90°$, the distance between data point $\mathbf{d}_q$ and ray $\mathbf{r}$ is no longer related to the direction of $\mathbf{r}$, but always equal to $||\mathbf{d}_q||$, which is illustrated by the case of point A in Figure 4.9. Such a requirement is reasonable, because in practical applications sectors with a very large span angle (e.g. bigger than 90°) are not suitable for data modeling, since they are too large and modeling the data using them will lose the details of data structure. Furthermore, this requirement can be easily satisfied by increasing the number of sectors used to model the data, which consequently decreases the sectors' span angles. When $\angle(\mathbf{r}, \mathbf{d}_q) \leq 90°$, the distance from data point $\mathbf{d}_q$ to the sector central ray $\mathbf{r}$ is $||\mathbf{d}_q - \mathbf{r}^T\mathbf{d}_q\mathbf{r}||$. The mathematical definition of sector central ray is given by

$$\mathbf{r} = \arg\min_{\|\boldsymbol{\psi}\|=1} \sum_{q=1}^{Q} \left\| \mathbf{d}_q - \boldsymbol{\psi}^T \mathbf{d}_q \boldsymbol{\psi} \right\|^2 \tag{4.22}$$

Following the proof given in [131], we can easily show that the sector central ray $\mathbf{r}$ is the eigenvector associated with the largest eigenvalue of the correlation matrix of $\{\mathbf{d}_1, \ldots, \mathbf{d}_Q\}$, which is $\frac{1}{Q}\sum_{q=1}^{Q}\mathbf{d}_q\mathbf{d}_q^T$. Since an eigenvector can take two opposite directions, the sector central ray takes the one that satisfies $\angle(\mathbf{r}, \mathbf{d}_q) \leq 90°$ for $q \in \{1, \ldots, Q\}$.

Table 4.6 Sector-based Clustering Algorithm

Given	Data points and sector number K_s.
Step 1.	Randomly select K_s data points to initialize the sector central rays. Each sector central ray is scaled to have a unit norm.
Step 2.	Assign each data point to its nearest sector central ray measured by the distance between the data point and the sector central ray. Thus a partition of the data points is formed.
Step 3.	For each sector, update the sector central ray according to Equation (4.22).

Step 4.	If the clustering partition does not change as compared to the partition obtained in the previous iteration, the algorithm converges and ends; otherwise, go to step 2.

During the clustering process, if in step 2 there exist data points that have a negative inner product with all sector central rays, or if in step 3 either direction of the eigenvector makes the sector central ray have a negative inner product with some data points in the sector, the algorithm is terminated and rerun with re-initialization and/or an increased K_{s}.

Table 4.6 shows the sector-based clustering algorithm. It performs a *K*-means type clustering scheme. Similar to the *K*-means clustering that converges to the local optimum of its objective function, the sector-based clustering also converges to a local optimum (if not the global optimum) of an objective function that is the summation of square distances from the sample to its sector central ray calculated by

$$\sum_{j=1}^{K_{\mathrm{s}}}\sum_{h=1}^{N_j}\left\|\mathbf{x}[n_h]-\mathbf{r}_j^T\mathbf{x}[n_h]\mathbf{r}_j\right\|^2 \tag{4.23}$$

where K_{s} is the number of sectors, N_j is the number of samples in sector j, $\mathbf{r}_j$ is the central ray of sector j, and n_h is the index of the hth sample in sector j. The following theorem states the local optimality of the sector-based clustering algorithm for minimizing the objective function.

Theorem 4.4 The sector-based clustering algorithm monotonically minimizes the summation of square distances from the sample to its sector central ray and reaches a local minimum (if not the global minimum) when it converges.

The proof of Theorem 4.4 is given in Appendix E. Since the sector-based clustering algorithm finds a local minimum of the objective function, we can run it multiple times with random initialization and select the partition with the minimum objective function value as the final clustering outcome, which will more likely achieve the global minimization.

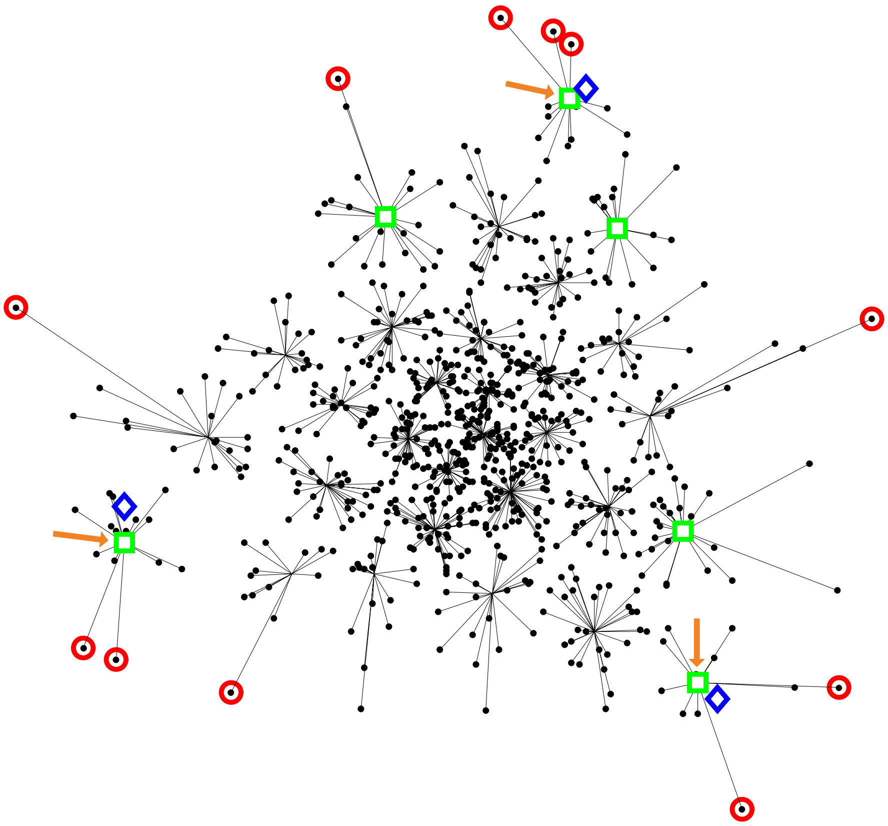


Figure 4.10 The sector-based clustering result obtained on the 800 large-norm samples of the toy dataset. The samples are also perspectively projected onto the plane passing through $[1\ 0\ 0]^T$, $[0\ 1\ 0]^T$, and $[0\ 0\ 1]^T$. Samples are indicated by black dots. Each sample is connected to its sector central ray by a line. Red circles indicate the edges detected by applying the edge detection algorithm on the samples. Green squares indicate the edges detected by applying the edge detection algorithm on the sector central rays obtained by sector-based clustering. Blue diamond markers indicate the positions of mixing matrix column vectors $\{\mathbf{A}\}$.

On the 800 large-norm samples of the toy dataset, we run the sector-based clustering 50 times with K_s equal to 30 and show the partition with the minimum objective function value in Figure 4.10. The obtained sector central rays can be taken as noise reduced data, based on which further edge detection and estimation of $\mathbf{A}$ can be carried out. Using sector central rays also greatly reduces the computational complexity of the proposed edge detection algorithm as compared to directly using the original samples whose number is usually much bigger than K_s. We run the edge detection algorithm on sector central rays and show the positions of the obtained edges in Figure 4.10 using green square markers. We can see that three of the edges identified based on sector central rays are much closer to the mixing matrix column vectors {**A**} than the edges identified based on original samples.

4.4.3 Minimization of Model Fitting Error

From Figure 4.10, we can see that although we detect edges based on sector central rays, not all of the detected edges are close to and can serve as a good estimate for {**A**}, due to the remaining noise and clustering imprecision. However, these edges can serve as a candidate pool, in which an accurate estimate of {**A**} lies. From the candidate pool, we select a *K*-size edge set that minimizes the following model fitting error for the estimate of {**A**},

$$\text{Model Fitting Error} = \sum_{j=1}^{K_s} \frac{N_j}{\sum_{m=1}^{K_s} N_m} \angle\left(\mathbf{r}_j, \mathbf{r}_j^{'}\right), \tag{4.24}$$

where $\mathbf{r}_j^{'}$ is the projection image of $\mathbf{r}_j$ on the cone generated by the edge set in consideration. Equation (4.24) is a weighted summation of the angle between the sector central ray and its projection image, while the weight is the sample proportion of the sector. Subsequently, only the sector central rays outside of the cone generated by the edge set in consideration contribute to this fitting error. When the edge set in consideration enlarges, less sector central rays are outside of the cone and the angles between these sector central rays and their projection images also become smaller. Therefore, the model fitting error is a monotonically decreasing measure when the edge set in consideration enlarges. So search for the best-fitting edge set (i.e. the edge set with the minimum model fitting error) can be fulfilled by the branch and bound algorithm [132], which guarantees the global optimality. The sequential forward floating search can also be used instead of the branch and bound algorithm to reduce the computational complexity at the price of achieving probably a local optimality [133]. We take the *K*-size best-fitting edge set as the estimate of **A** and denote this estimate by $\hat{\mathbf{A}} = \left[\hat{\mathbf{a}}_1 \quad \ldots \quad \hat{\mathbf{a}}_K\right]$.

On the toy dataset, we apply the branch and bound algorithm to search for the best-fitting edge triplet from the candidate pool indicated by green square markers in

Figure 4.10 to form $\hat{\mathbf{A}}$. The resulted best-fitting edge triplet contains the three green square markers that are pointed by the orange arrows in Figure 4.10 and are closest to the three blue diamond markers that indicate the positions of column vectors in **A**.

4.4.4 Recovery of Sources and Flowchart of CAM Framework

To estimate the sources, we need to utilize all the mixture samples including the small-norm samples. Obviously, some mixture samples are outside of cone($\{\hat{\mathbf{A}}\}$). We project all mixture samples in {**X**} onto cone($\{\hat{\mathbf{A}}\}$) and denote the projection images by $\{\mathbf{X}_\mathrm{p}\}$. Such a projection can also be viewed as a noise reduction step and ensures the non-negativity of the source estimate. If $\hat{\mathbf{A}}$ satisfies *A*3, $\hat{\mathbf{S}}$ is calculated by

$$\hat{\mathbf{S}} = \left(\hat{\mathbf{A}}^T\hat{\mathbf{A}}\right)^{-1}\hat{\mathbf{A}}^T\mathbf{X}_\mathrm{p}\,. \tag{4.25}$$

Then the obtained $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ need to be scaled as aforementioned by

$$\begin{aligned}\hat{\mathbf{A}} &= \left(\mathrm{diag}\left(\mathbf{X}_\mathrm{p}\mathbf{1}_N\right)\right)^{-1}\hat{\mathbf{A}}\,\mathrm{diag}\left(\hat{\mathbf{S}}\mathbf{1}_N\right)\\ \hat{\mathbf{S}} &= \left(\mathrm{diag}\left(\hat{\mathbf{S}}\mathbf{1}_N\right)\right)^{-1}\hat{\mathbf{S}}\end{aligned}\,. \tag{4.26}$$

If $\hat{\mathbf{A}}$ does not satisfy *A*3 but satisfies *A*4, we first scale $\hat{\mathbf{A}}$ and $\mathbf{X}_\mathrm{p}$ using the following equations

$$\begin{aligned}\hat{\mathbf{A}} &= \left(\mathrm{diag}\left(\mathbf{X}_\mathrm{p}\mathbf{1}_N\right)\right)^{-1}\hat{\mathbf{A}}\\ \mathbf{X}_\mathrm{p} &= \left(\mathrm{diag}\left(\mathbf{X}_\mathrm{p}\mathbf{1}_N\right)\right)^{-1}\mathbf{X}_\mathrm{p}\end{aligned}\,, \tag{4.27}$$

so that each row of $\mathbf{X}_\mathrm{p}$ has a unit summation, and then $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ are calculated by the iterative algorithm presented in Table 4.2, with $\hat{\mathbf{A}}$, $\hat{\mathbf{S}}$ and $\mathbf{X}_\mathrm{p}$ replacing **A**, **S**, and **X** in Table 4.2, respectively, as $\mathbf{X}_\mathrm{p} = \hat{\mathbf{A}}\,\hat{\mathbf{S}}$ can be taken as a noise-free model that the iterative algorithm is originally designed for. Note that the scaling of Equation (4.27) usually only slightly changes $\hat{\mathbf{A}}$ and $\mathbf{X}_\mathrm{p}$, because **X** has already been scaled to have unit row summations before projection.

Now, we have the foundation for introducing the flowchart of CAM framework in Figure 4.11. It shall be noted that this algorithm flowchart is for the applications where source number *K* is known. On the toy dataset, we recover the sources using all 1600 mixture samples and the mixing matrix estimate obtained in the previous sub-section.

The resulted accuracies of recovering mixing matrix and sources are 0.9826 and 0.9171, respectively.

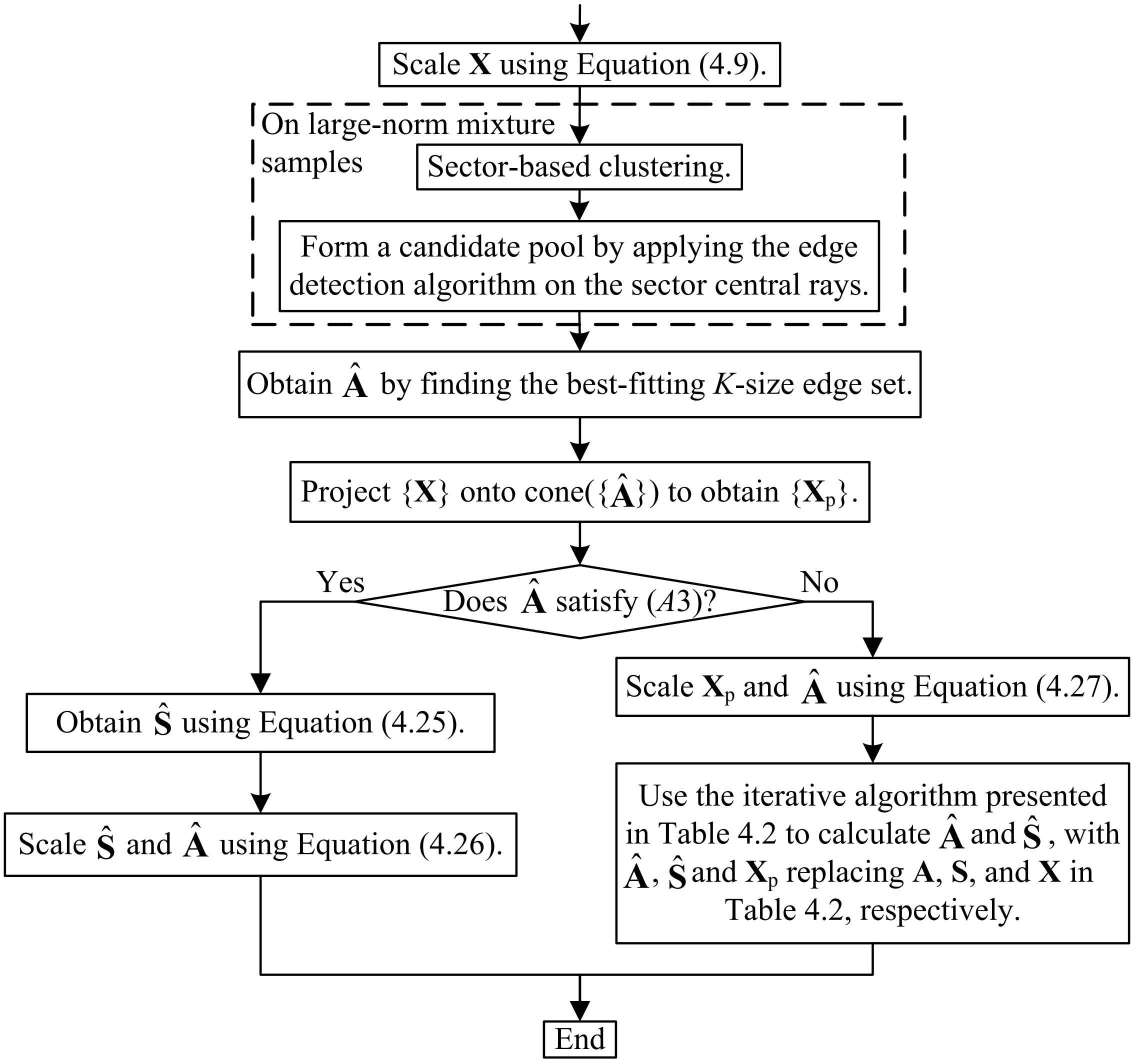

Scale **X** using Equation (4.9).
On large-norm mixture samples
Sector-based clustering.
Form a candidate pool by applying the edge detection algorithm on the sector central rays.
Obtain $\hat{\mathbf{A}}$ by finding the best-fitting *K*-size edge set.
Project {**X**} onto cone({$\hat{\mathbf{A}}$}) to obtain {$\mathbf{X}_p$}.
Yes
Does $\hat{\mathbf{A}}$ satisfy (*A*3)?
No
Obtain $\hat{\mathbf{S}}$ using Equation (4.25).
Scale $\hat{\mathbf{S}}$ and $\hat{\mathbf{A}}$ using Equation (4.26).
Scale $\mathbf{X}_p$ and $\hat{\mathbf{A}}$ using Equation (4.27).
Use the iterative algorithm presented in Table 4.2 to calculate $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$, with $\hat{\mathbf{A}}$, $\hat{\mathbf{S}}$ and $\mathbf{X}_p$ replacing **A**, **S**, and **X** in Table 4.2, respectively.
End

Figure 4.11 Algorithm flowchart of CAM framework when source number *K* is known.

4.4.5 Stability Analysis Based Model Order Selection

To determine the number of underlying sources, CAM uses stability analysis based model order selection, which was first introduced for data clustering to determine the number of clusters [49] and frequently used in applications like gene clustering on microarray gene expression data [47, 49]. CAM applies stability analysis based model order selection to nBSS problems by designing a stability analysis scheme and a model instability index that are suitable for non-negative well-grounded source separation.

CAM examines the stability of the best-fitting model of different model order in the presence of data perturbation through a 2-fold cross-validation type experiment. The stability of the separation solution varies with the number of sources being inferred. For instance, inferring too many sources allows the unnecessary model flexibility to fit random data perturbation and noise, and thus produces unstable solutions. Inferring too few sources might lead to unstable solutions since the lack of degree of freedom forces the algorithm to ambiguously mix sources that should be consistently kept separate. Following these considerations, the "correct" source number is defined as that of the separation solution with maximum stability. Then a stable model (compared among all best-fitting models of different model order) with the minimum fitting error (compared among all models of the selected model order) will be taken as the final solution.

As we have discussed before, the large-norm samples are expected to have a higher SNR. The model order selection scheme only utilizes the large-norm samples (while large-norm outliers, if exist, should also be excluded via the aforementioned pre-processing step). In each cross-validation trial, the large-normal samples are randomly divided into two folds of equal size. On each data fold, estimates for **A** with the model order k varying from 2 to $K_{\max}$ are generated by the CAM solution introduced in the previous sub-sections and Figure 4.11, where $K_{\max}$ is the largest possible model order in consideration. Also note that sector-based clustering in all cross-validation trials is run with a common sector number $K_{\mathrm{s}} > K_{\max}$. Let L denote the number of cross-validation trials. In the lth ($l \in \{1, \ldots, L\}$) cross-validation trial, use $\hat{\mathbf{A}}_{k,l,1}$ and $\hat{\mathbf{A}}_{k,l,2}$ to denote the k order estimates for **A** that are obtained on the first and second data fold, respectively. To measure the stability of the k order model, we define an instability index given by

$$\text{Instability Index} = \frac{1}{L}\sum_{l=1}^{L} \angle\left(\hat{\mathbf{A}}_{k,l,1}, \hat{\mathbf{A}}_{k,l,2}\right). \tag{4.28}$$

However, to apply instability index for model order selection, some additional considerations are needed. Let us consider $\angle(\hat{\mathbf{A}}_{k,l,1}, \hat{\mathbf{A}}_{k,l,2})$, the discrepancy between the two k order estimates for **A** obtained in the lth cross-validation trial. For the simplicity of discussion, we ignore the difference between the sector central rays obtained in the two data folds, i.e. assume that the sector central rays obtained in the first data fold are the same as the sector central rays obtained in the second data fold, and use $\{\mathbf{R}_l\}$ to denote the set of sector central rays. It can be recognized that when k increases, $\angle(\hat{\mathbf{A}}_{k,l,1}, \hat{\mathbf{A}}_{k,l,2})$ has a innate trend to decrease, since more and more vectors are selected from $\{\mathbf{R}_l\}$ to form $\{\hat{\mathbf{A}}_{k,l,1}\}$ and $\{\hat{\mathbf{A}}_{k,l,2}\}$. Such intuition can be further explained as follows.

Because $\{\hat{\mathbf{A}}_{k,l,1}\}$ and $\{\hat{\mathbf{A}}_{k,l,2}\}$ are both selected from $\{\mathbf{R}_l\}$, $\angle(\hat{\mathbf{A}}_{k,l,1}, \hat{\mathbf{A}}_{k,l,2})$ is the result of selecting different vectors in $\{\mathbf{R}_l\}$ to form $\{\hat{\mathbf{A}}_{k,l,1}\}$ and $\{\hat{\mathbf{A}}_{k,l,2}\}$ when minimizing the model fitting error. To evaluate the difference between $\{\hat{\mathbf{A}}_{k,l,1}\}$ and $\{\hat{\mathbf{A}}_{k,l,2}\}$, we define an inconsistency rate as

$$\text{Inconsistency Rate} = 1 - \frac{\operatorname{card}\left(\left\{\hat{\mathbf{A}}_{k,l,1}\right\} \bigcap \left\{\hat{\mathbf{A}}_{k,l,2}\right\}\right)}{k}, \tag{4.29}$$

where card(•) is the cardinality of the input set. When $k > \lfloor K_\mathrm{s}/2 \rfloor$, the upper limit of the inconsistency rate is $1 - (2k - K_\mathrm{s})/k = K_\mathrm{s}/k - 1$ that is a monotonically decreasing function when k increases.

Furthermore, let us consider a scheme that randomly selects sector central rays to form the estimate for $\mathbf{A}$, i.e. each sector central ray has an equal chance to be selected as the estimate for one column vector of $\mathbf{A}$. The random selection scheme can practically be taken as the worst estimation scheme for any model order and thus is equally bad for all model orders. Let $\hat{\mathbf{A}}_{k,l,1,\mathrm{rand}}$ and $\hat{\mathbf{A}}_{k,l,2,\mathrm{rand}}$ denote the k order estimates for $\mathbf{A}$ generated by randomly selecting sector central rays obtained on data fold 1 and data fold 2, respectively. Again, for the simplicity of discussion, we ignore the difference between the sector central rays obtained on the two data folds. Expectation of the inconsistency rate between $\{\hat{\mathbf{A}}_{k,l,1,\mathrm{rand}}\}$ and $\{\hat{\mathbf{A}}_{k,l,2,\mathrm{rand}}\}$ is

$$\sum_{o=\max\{0,2k-K_\mathrm{s}\}}^{k} (1 - o/k) \frac{\binom{K_\mathrm{s}}{k}\binom{k}{o}\binom{K_\mathrm{s}-k}{k-o}}{\binom{K_\mathrm{s}}{k}\binom{K_\mathrm{s}}{k}} = \sum_{o=\max\{0,2k-K_\mathrm{s}\}}^{k} (1 - o/k) \frac{\binom{k}{o}\binom{K_\mathrm{s}-k}{k-o}}{\binom{K_\mathrm{s}}{k}}, \tag{4.30}$$

where

$$\frac{\binom{k}{o}\binom{K_\mathrm{s}-k}{k-o}}{\binom{K_\mathrm{s}}{k}} = \Pr(o), \qquad o \in \{\max\{0, 2k-K_\mathrm{s}\}, \cdots, k\}$$

is the probability mass function of a special case of the hypergeometric distribution. Based on the properties of hypergeometric distribution, we conclude that the expectation of the inconsistency rate presented in Equation (4.30) is equal to 1 − k/K_s, which also monotonically decreases when k increases. Because the random selection scheme is equally bad for all model orders, its monotonically decreasing expectation of inconsistency rate also reflects the innate decreasing trend of the discrepancy between **A** estimates obtained by selecting sector central rays in the two data folds.

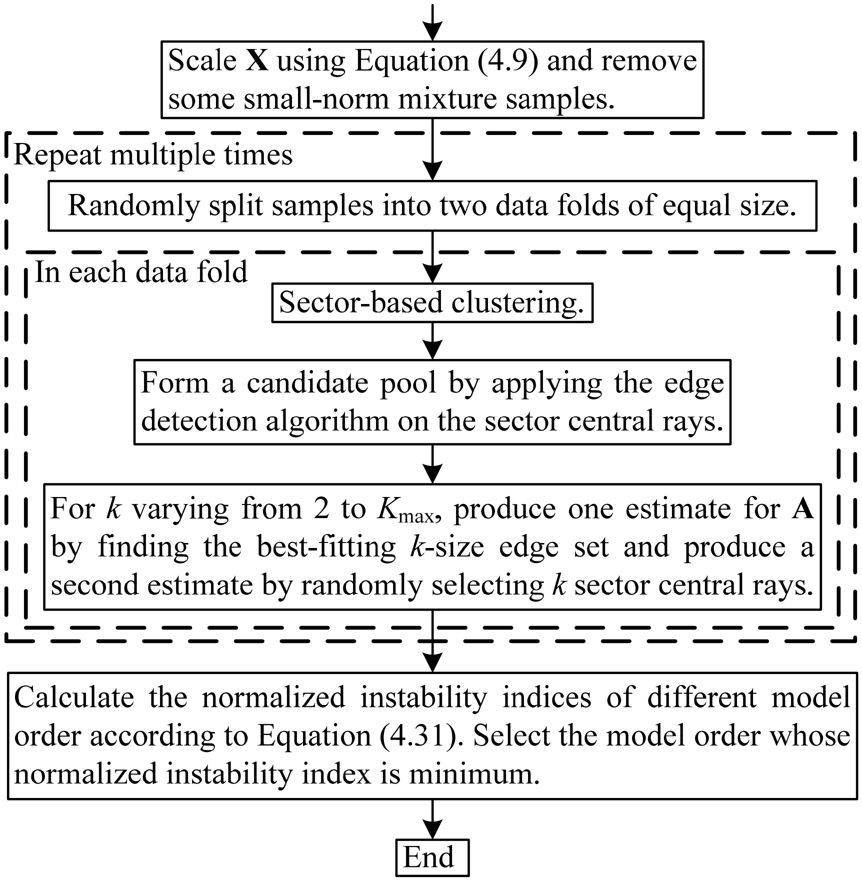

Scale **X** using Equation (4.9) and remove some small-norm mixture samples.
Repeat multiple times
Randomly split samples into two data folds of equal size.
In each data fold
Sector-based clustering.
Form a candidate pool by applying the edge detection algorithm on the sector central rays.
For k varying from 2 to K_max, produce one estimate for **A** by finding the best-fitting k-size edge set and produce a second estimate by randomly selecting k sector central rays.
Calculate the normalized instability indices of different model order according to Equation (4.31). Select the model order whose normalized instability index is minimum.
End

Figure 4.12 Flowchart of the model order selection scheme of CAM framework.

The decreasing trend of the instability index needs to be neutralized so that it can be comparable for different values of model order, which is mandatory for model order selection. The normalization strategy employed here represents one possible way to achieve comparability while keeping the stability measure interpretable. We calculate the discrepancy between the **A** estimate produced by applying the CAM solution in one data fold and the **A** estimate produced by applying the random selection scheme in the other data fold, and use this discrepancy to normalize the instability index given by Equation (4.28). The normalized instability index is formulated as

$$\text{Normalized Instability Index} = \frac{\frac{1}{L}\sum_{l=1}^{L}\angle\left(\hat{\mathbf{A}}_{k,l,1},\hat{\mathbf{A}}_{k,l,2}\right)}{\frac{1}{L}\sum_{l=1}^{L}\frac{1}{2}\left[\angle\left(\hat{\mathbf{A}}_{k,l,1},\hat{\mathbf{A}}_{k,l,2,\text{rand}}\right)+\angle\left(\hat{\mathbf{A}}_{k,l,2},\hat{\mathbf{A}}_{k,l,1,\text{rand}}\right)\right]}. \quad (4.31)$$

We stress here that there does not exist a generally accepted "correct" strategy for normalization, which is also shared by the authors of [49] who applied stability analysis based model order selection for determining the cluster number in data clustering. Figure 4.12 shows the flowchart of the stability analysis based model order selection scheme used by CAM.

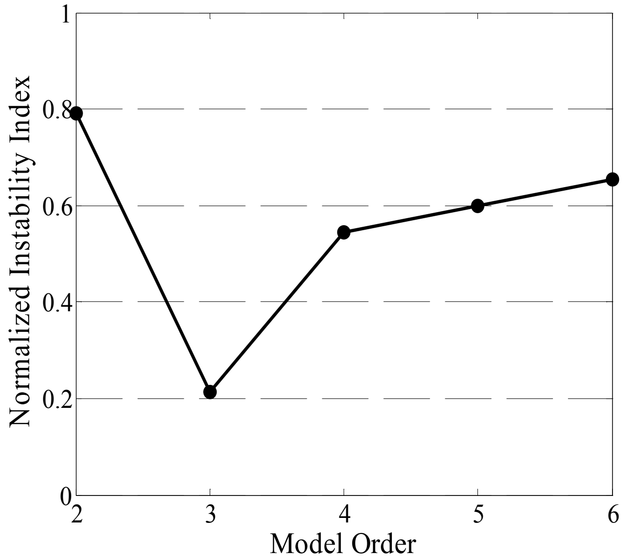


Figure 4.13 Normalized instability indices of different model order obtained on the toy dataset.

Figure 4.13 shows the normalized instability indices of different model order obtained on the toy dataset. 50% of the samples whose norms are small are excluded in model order selection. The sector-based clustering is run with 30 sectors and the clustering outcome is the optimal one from 50 runs. The number of cross-validation trails is 50. From Figure 4.13, we can see that the model order with maximum stability is 3, which is consistent with the ground truth that we generate this toy dataset by mixing three sources.

Note that the model order selection scheme considers model orders starting from 2. We assume that the mixtures contain at least two sources. In our view, the question whether the observed data are generated by a single source should be addressed in advance, for example, by some application specific test. Also note that the stability analysis based model order selection scheme of CAM is uniformly applicable to the determined, over-determined, and under-determined cases, which enables CAM to tell whether an nBSS problem is under-determined in practical use. And if the problem is under-determined, we recognize that it may not be identifiable, according to Remark 4.2.

4.5 Performance Evaluation of CAM Framework

Similar to the validation of CAM principle, we evaluated the performance of CAM framework based on its ability to separate numerically mixed microarray gene expression data. The performance evaluation was also conducted on three cases, i.e. determined case (4 mixtures vs. 4 sources), over-determined case (6 mixtures vs. 4 sources), and under-determined case (3 mixtures vs. 4 sources). In the determined and over-determined cases, (*A*3) is satisfied. In the under-determined case, (*A*3) is not satisfied, but (*A*4) is satisfied. We used the 100 randomly generated mixing matrices and the four ovarian cancer subtype gene expression profiles that have been used for principle validation of CAM in Sub-section 4.3.4 to generate the simulation datasets for the performance evaluation. Noise $\boldsymbol{\varepsilon}[n]$, drawn from a Gaussian distribution with a mean of $\mathbf{0}_M$ and a covariance matrix of $\sigma^2\mathbf{I}_M$, was added to the simulation data according to Equation (4.18). Simulation datasets with multiple SNR levels were generated, where the SNR values were calculated by Equation (4.19). To assure a statistically meaningful evaluation, for each SNR level, 100 simulation datasets were generated using the 100 different mixing matrices and the same sources. To assure the non-negativity of the mixture data, we truncated the negative values (due to additive Gaussian noise) to zero.

We calculated the recovery accuracies of mixing matrix, sources, and distinct patterns of sources obtained by CAM, and compared them to those of nICA, SNICA, NMF, SNMF, and N-FINDR, focusing on the determined and over-determined cases. In the under-determined case, we evaluated CAM's performance. The model order selection ability of CAM was also evaluated by its frequency of detecting the correct source number on the 100 datasets of each SNR level. Additionally, since it is hard to show a 4-D scatter plot of four source profiles, we selected three of them, i.e. CHTN-OS-102, CHTN-OM-029, and CHTN-OE-060, and draw their scatter plot in Figure 4.1. From Figure 4.1, we can see that many genes have relatively small expression levels and thus represent small-norm samples, which is also true if we consider all four source profiles.

In addition to the performance comparison between CAM and the competing methods, we also studied the impact of the proposed noise reduction scheme of CAM framework on the performance of competing methods when they also paritially applied and benefited from CAM's noise reduction scheme. As aforementioned, the proposed noise reduction scheme in CAM framework consists of three basic components, i.e., estimating **A** only based on large-norm samples, projecting all samples onto cone($\{\hat{\mathbf{A}}\}$) to calculate $\hat{\mathbf{S}}$, and replacing the original data with sector central rays when estimating **A**. In this study, the first two components of the noise reduction scheme were also employed by the competing methods. Specifically, in the improved, noise-resistant scenario, the competing methods were first run on the large-norm samples to obtain $\hat{\mathbf{A}}$. Then, $\{\mathbf{X}\}$ were projected onto cone($\{\hat{\mathbf{A}}\}$) to obtain $\{\mathbf{X}_{\mathrm{p}}\}$. Finally, $\hat{\mathbf{S}}$ was calculated by Equation (4.25). The study involved both the determined and over-determined cases.

In performing CAM, we excluded 50% of the mixture samples whose vector norms were small, in estimating **A** and model order selection (accordingly, the same step was taken by the competing methods when they were empowered by excluding small-norm samples for estimating **A**). The sector-based clustering was run with the sector number equal to 20 and 30 to study the impact of different sector numbers on the separation performance, where the two performance evaluation results were denoted by CAM-20S and CAM-30S, respectively. The sector-based clustering chose the optimal clustering partition from 10 independent runs. 30 cross-validation trials were conducted for model order selection of CAM. In the iterative algorithm for estimating **S** when ($A3$) is not satisfied by the estimated mixing matrix, the parameter N_{minPSG} was set at 100 and the learning rate η was set at 0.5. For the competing methods, the weight of the source sparseness measure term in SNMF's objective function was set at 1; and in the over-determined case, nICA, SNICA, and N-FINDR required dimension reduction from 6 dimensions to 4 dimensions, which was fulfilled by the principal component analysis methods [109, 116, 130].

4.5.1 Accuracies of Recovering Sources, Mixing Matrix, and Distinct Patterns of Sources

Figure 4.14 and Figure 4.15 show the comparative performances of CAM and the competing methods in the determined and over-determined cases, respectively. Figure 4.16 shows CAM's performance in the under-determined case. The accuracies of recovering mixing matrix, sources, and distinct patterns of sources shown in the figures are the average value of the performance measures obtained on the 100 simulation datasets generated by the same sources but different mixing matrices and with the same SNR level. It can be seen that CAM outperforms all other five competing methods in both the determined and over-determined cases (see Figure 4.14a, c, e and Figure 4.15a, c, e). This study demonstrates the superior performance of CAM framework, as compared to the competing methods, in solving noisy non-negative well-grounded source separation problems. Specifically, from Figure 4.14e and Figure 4.15e, we can see that CAM consistently achieves a significant accuracy improvement in recovering the distinct patterns of sources that is a major objective of the separation algorithm and is of significant practical importance, as aforementioned. From Figure 4.16, we can see that CAM obtains good accuracies of recovering the mixing matrix, sources, and distinct patterns of sources in the under-determined case.

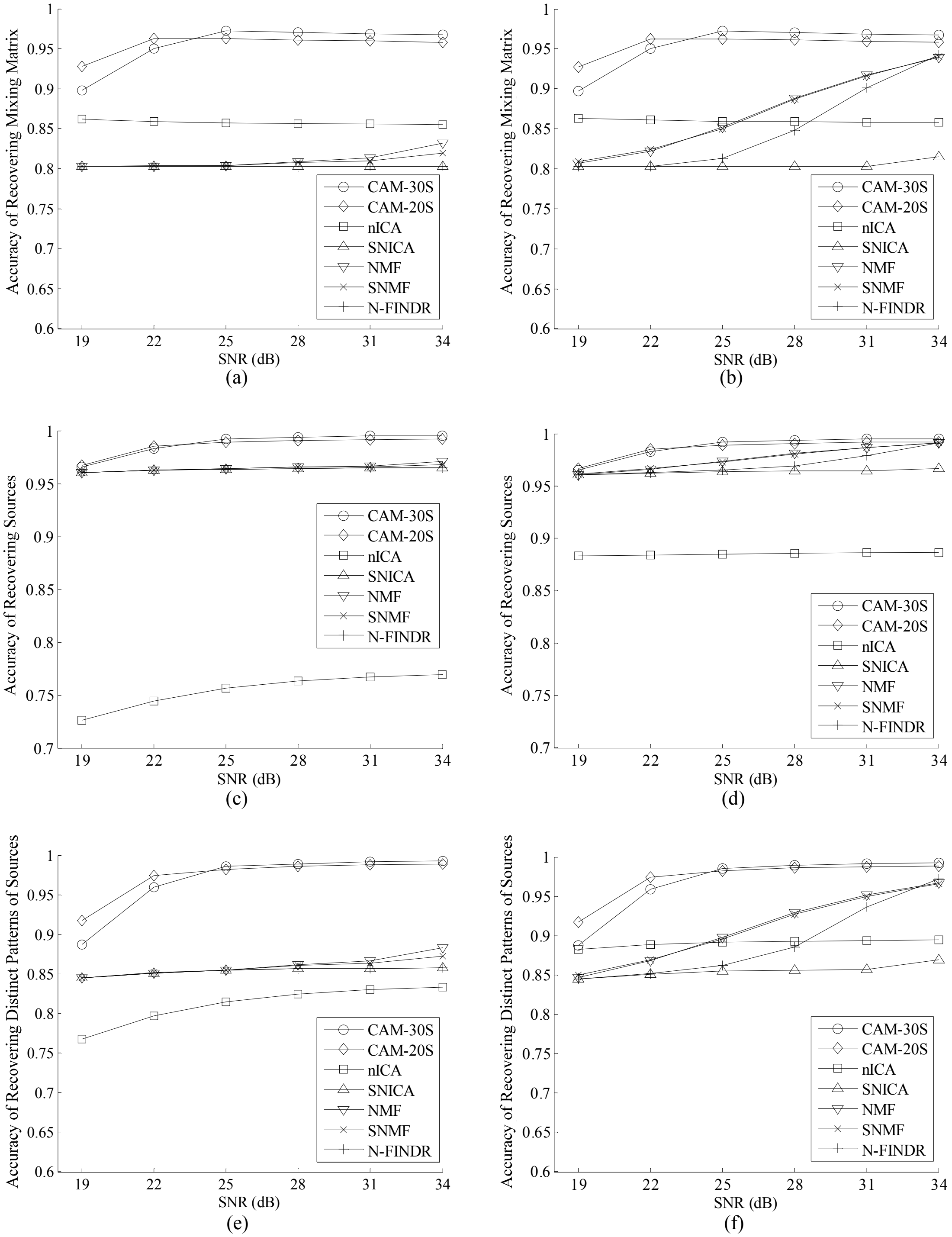


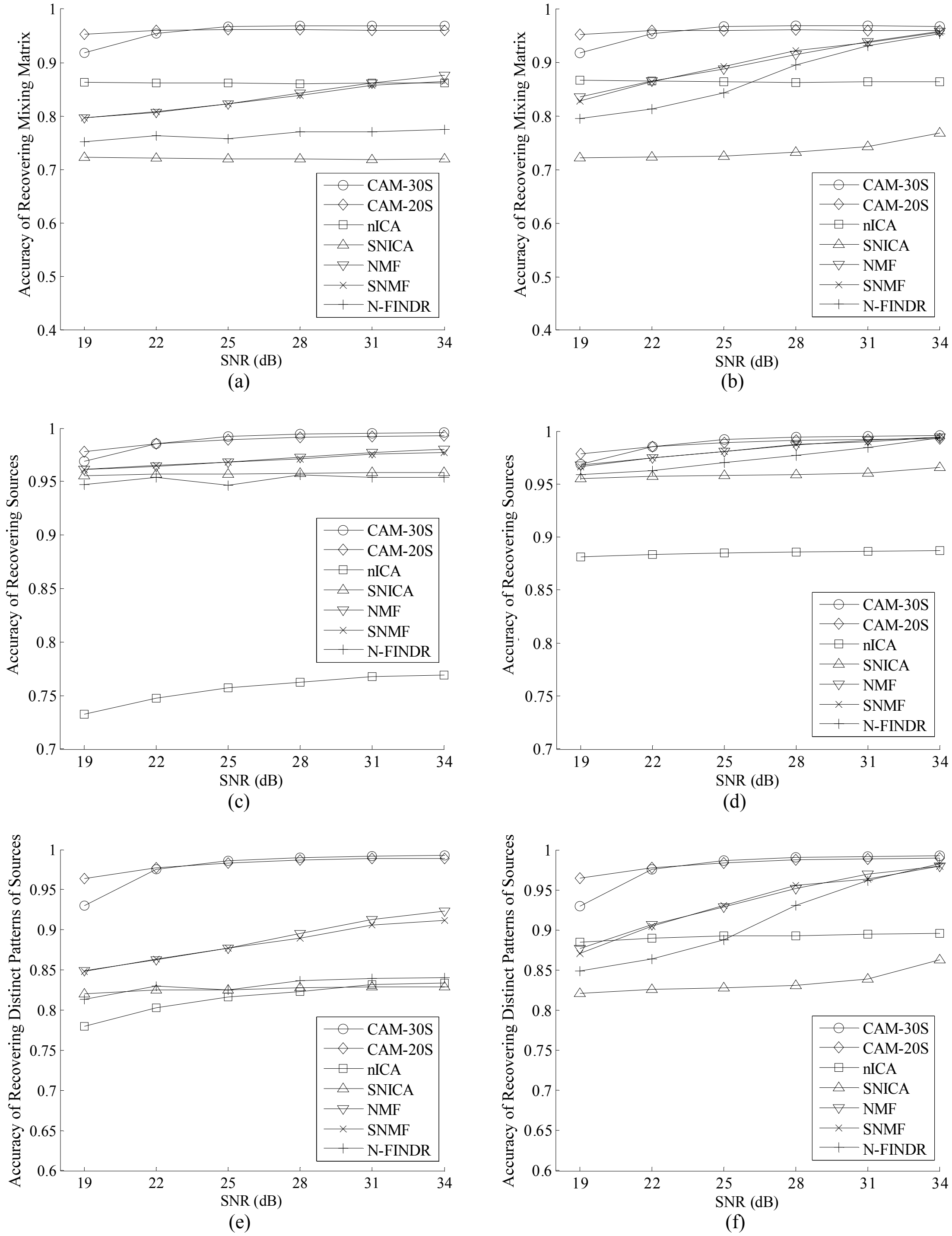
Figure 4.14 Performance comparison in the determined case. (a) Comparison on the accuracy of recovering mixing matrix. (b) Comparison on the accuracy of recovering mixing matrix when the competing methods were empowered by two of CAM's noise reduction steps. (c) Comparison on the accuracy of recovering sources. (d) Comparison on the accuracy of recovering sources when the competing methods were empowered by two of CAM's noise reduction steps. (e) Comparison on the accuracy of

recovering distinct patterns of sources. (f) Comparison on the accuracy of recovering distinct patterns of sources when the competing methods were empowered by two of CAM's noise reduction steps.

There are several interesting observations from our experiments. We found that performances of some competing methods stay unchanged or do not improve much when SNR increases, especially in the determined case (see Figure 4.14a, c, e). For example, the performance curves of SNICA and N-FINDR are (almost) flat in Figure 4.14a, c, e. And the performance curves of NMF and SNMF are also (almost) flat in the relatively low SNR region, i.e. 19dB ~ 25dB. By examining the recovery results, we found that the recovered mixing matrices produced by these competing methods (i.e. NMF, SNMF, N-FINDR and SNICA) are close or equal to a scaling and permutation matrix. Such result means that these methods simply took the mixtures as the estimates of sources, indicating an algorithm collapse in the presence of significant noise. Since the mixing matrix estimate is always close or equal to a scaling and permutation matrix and the source estimate is always close or equal to the mixtures while the 100 mixture datasets for each SNR level were generated using the same 100 mixing matrices and sources, the performance of these methods does not change (much) when SNR increases. It shall be noted that the "local" SNR of small-norm samples calculated by Equation (4.20) is much smaller than the overall SNR of all samples calculated by Equation (4.19). We found that due to the truncation of negative values to zero, some small-norm mixture samples are very close to or on the axes of the mixture sample scatter plot. These "extreme" points in terms of vector direction significantly influenced these competing methods' estimation of mixing matrix, which makes them estimate the mixing matrix as a scaling and permutation matrix. The performances of NMF and SNMF start to improve after the SNR level reaches 28dB, indicating that NMF and SNMF are not as sensitive as N-FINDR and SNICA to these small-norm "extreme" points. From the low SNR region in Figure 4.14c, we can see that although these competing methods almost simply took the mixtures as the estimates of sources, the average correlation coefficient between the source estimates and the true sources is about 0.96. Such high correlation coefficient is expected while could be misleading, since the sources are highly correlated and the condition numbers of the mixing matrices are small, which lead to high correlation between sources and mixtures.

Therefore, we propose that in the determined case the baseline correlation coefficient 0.96 shall be taken into account when evaluating how good the accuracy of recovering sources is. A similar situation of performance changelessness over different SNR levels due to noise impact and that the simulation datasets of each SNR level were generated using the same mixing matrices and source, is also observed on some methods in the over-determined case (see Figure 4.15a, c, e), where the performance curves of N-FINDR and SNICA largely keep flat.

We found that nICA shows consistently unsatisfactory performance in recovering the underlying sources as compared to the other methods in both the determined and over-determined cases (see Figure 4.14c and Figure 4.15c), simply due to the violation of the basic assumption in nICA model that the sources must be uncorrelated. However, considering only the recovery of the distinct patterns of sources, from Figure 4.14e and Figure 4.15e, we can see that the relative performance of nICA becomes reasonable. We also observed that nICA's accuracy of recovering mixing matrix keeps almost unchanged when SNR increases in both the determined and over-determined cases. By examining its recovered mixing matrices, we found that on datasets generated by the same mixing matrix (and also the same sources) but with different SNR levels, the mixing matrix estimates produced by nICA are very similar, which makes its performance curve almost flat. A possible reason is that it is the impact of source correlatedness rather than the SNR level that dominates nICA's performance on estimating mixing matrix in the experiments, while the source dependence/correlatedness level does not change over different SNR levels.

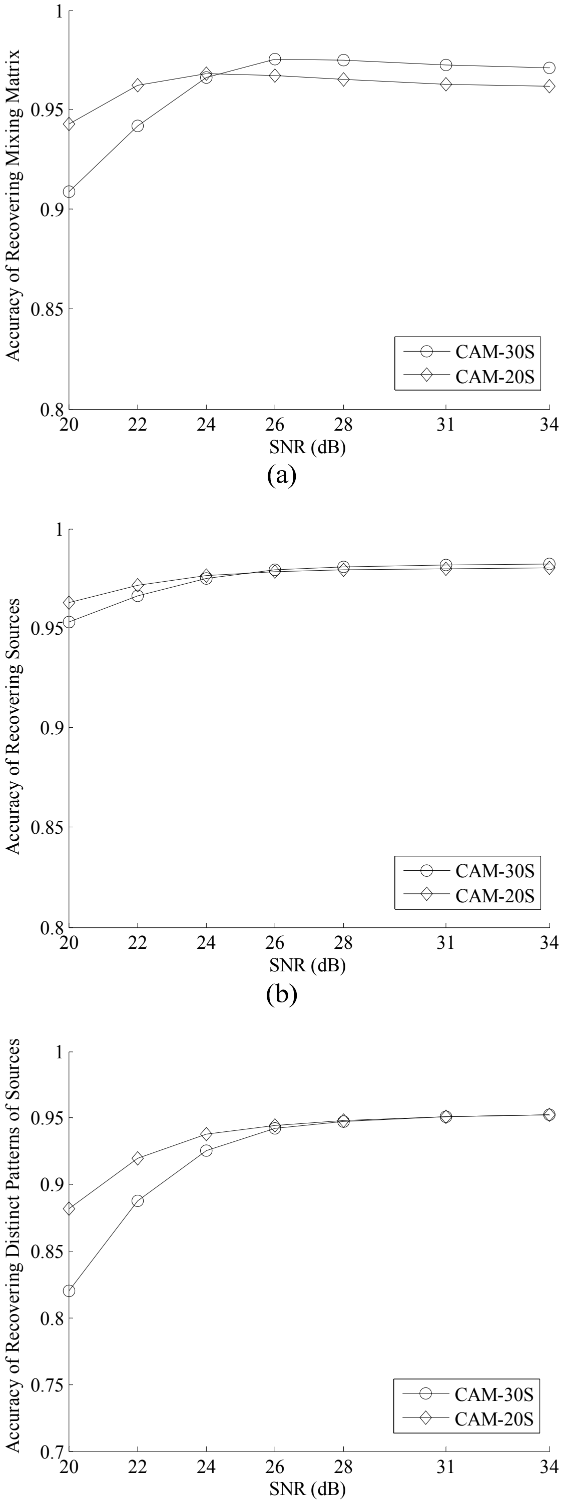


(a) (b) (c) (d) (e) (f)

Figure 4.15 Performance comparison in the over-determined case. (a) Comparison on the accuracy of recovering mixing matrix. (b) Comparison on the accuracy of recovering mixing matrix when the competing methods were empowered by two of CAM's noise reduction steps. (c) Comparison on the accuracy of recovering sources. (d) Comparison on the accuracy of recovering sources when the competing methods were empowered by two of CAM's noise reduction steps. (e) Comparison on the accuracy of

recovering distinct patterns of sources. (f) Comparison on the accuracy of recovering distinct patterns of sources when the competing methods were empowered by two of CAM's noise reduction steps.

We found that the performances of NMF and SNMF are consistently similar in all circumstances, probably due to the fact that the sources are not overall sparse, leading to the insignificant performance difference between SNMF and NMF. We also found that the performance of SNICA is unsatisfactory and sensitive to even small noise, because SNICA strictly requires zero model fitting error and non-negative recovered source values. Its performance curves are (almost) flat in both determined and over-determined cases. Similar to CAM, N-FINDR assumes the sources to be well-grounded and recovers the mixing matrix through identifying WGPs. However, since N-FINDR utilizes only a single sample to estimate each mixing matrix column vector, it is very sensitive to noise or outliers, which leads to the flat performance curves of N-FINDR in the determined case and the unstable performance of N-FINDR in the over-determined case, see Figure 4.15a, c, e, in which its performance curves fluctuate (not monotonically increase) when the SNR level increases. Compared to N-FINDR's perfect recovery accuracy obtained on noise-free data in Sub-section 4.3.4, the experimental results obtained here show that although the principle of N-FINDR is valid for non-negative well-grounded source separation, its recovery result is very sensitive to and can be easily damaged by noise. Compared to N-FINDR, CAM not only has a valid principle for non-negative well-grounded source separation, but also has a robust estimation scheme for identifying the model.



(a)

(b)

(c)

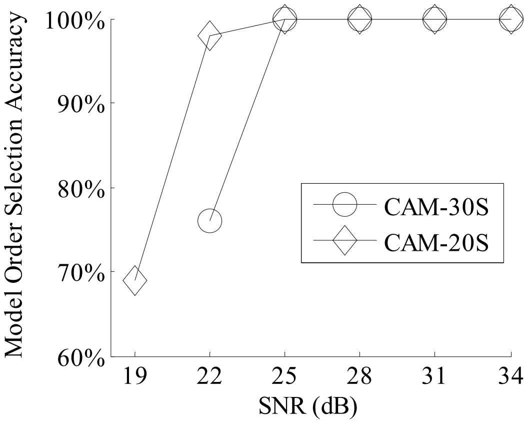
Figure 4.16 Performance evaluation of CAM in the under-determined case. (a) The accuracy of recovering mixing matrix. (b) The accuracy of recovering sources. (c) The accuracy of recovering distinct patterns of sources.

It is also observed that CAM-20S performs better than CAM-30S when the noise level is high, but worse when the noise level is low, in all cases. Explanation of this observation is that when data are modeled with fewer sectors, noise reduction resulted from data averaging (i.e. replacing the original data with sector central rays for estimating the mixing matrix) is stronger, and hence CAM-20S performs better than CAM-30S when there is significant noise. However, modeling data with fewer sectors, i.e. more severe data averaging, may cause significant loss of detailed data structure. For example, sector central rays become further away from the boundary of their sectors due to bigger sector size. When the noise level is low, the impact of losing more data structure details cannot be compensated by the benefit of stronger data averaging for noise reduction, and thus CAM-20S performs worse than CAM-30S in the high SNR region. This explanation also enlightens that modeling data with more sectors is suitable for data containing less noise and that for a particular sector number (sector size in another word) there may be a certain SNR region where clustering with this number of sectors can produce the best performance of CAM. Consequently, in Figure 4.14a, Figure 4.15a, and Figure 4.16a, we can see that CAM's accuracy of recovering mixing matrix may have a slight drop when SNR continues to increase beyond 24dB ~ 28dB.

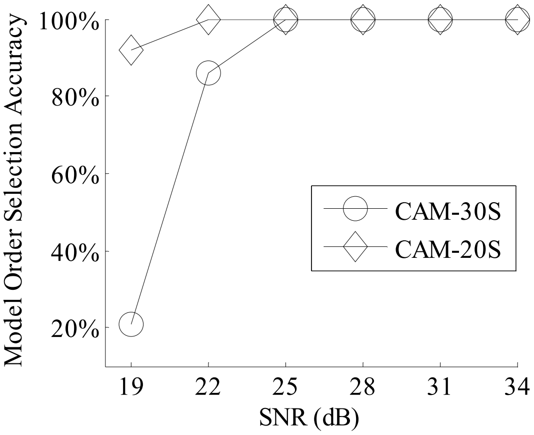
The comprehensive noise reduction scheme of CAM includes three components, i.e., estimating $\mathbf{A}$ only based on large-norm samples, projecting all samples onto cone($\{\hat{\mathbf{A}}\}$) for calculating $\hat{\mathbf{S}}$, and replacing the original data with sector central rays for estimating $\mathbf{A}$. The power of the first two components is demonstrated not only by CAM's outperformance over the competing methods, but also by the performance improvement gained by the competing methods when they also applied and benefited from these two noise reduction steps of CAM. The performance improvement gained by the competing methods can be clearly seen by comparing Figure 4.14b, d, f and Figure 4.15b, d, f to Figure 4.14a, c, e and Figure 4.15a, c, e, respectively. We also observe that when empowered by the two noise reduction steps of CAM, the competing methods, i.e. NMF, SNMF, N-FINDR, and SNICA, show a clear performance improvement trend when the SNR increases, which also verifies our previous analysis that the small-norm "extreme" data points may heavily influence these methods and make them show flat performance curves. The importance of the third component of CAM's noise reduction scheme, i.e. replacing the original data with sector central rays for estimating **A**, is also reflected in Figure 4.14b, d, f and Figure 4.15b, d, f, in which CAM still outperforms the competing methods, even though they utilized the first two noise reduction steps of CAM. The only exception is that in the over-determined case CAM-20S does not perform as well as NMF, SNMF, and N-FINDR in recovering sources at 34dB SNR level (see Figure 4.15d), while CAM-30S still outperforms all competing methods. CAM's outperformance over N-FINDR empowered by CAM's first two noise reduction steps particularly highlights the power of the third noise reduction step of CAM, i.e. estimating **A** using sector central rays, where CAM and N-FINDR share the source well-groundedness assumption and both of them recover the mixing matrix by detecting/estimating WGPs. As aforementioned, N-FINDR uses a single sample to estimate each column vector of **A**, while CAM estimates **A** using sector central rays that are more robust against noise than single samples.

4.5.2 Accuracy of Model Order Selection Using Stability Analysis

Since for each SNR level we have 100 simulation datasets, each of which was produced using a randomly generated mixing matrix, the frequency that CAM correctly detected the source number on the 100 simulation datasets was taken as the model order selection accuracy. Figure 4.17a, b, c shows the model order selection accuracy obtained by CAM in the determined, over-determined, and under-determined cases, respectively. In the determined and over-determined cases, both CAM-30S and CAM-20S have a model order selection accuracy of 100% when SNR was higher than 25dB. In all three cases, we see that CAM-20S outperforms CAM-30S when there is significant noise. In the under-determined case, we see a slight drop of the model order selection accuracy when SNR is high, which is similar to the situation that the accuracy of recovering mixing matrix slightly drops in the high SNR region. A possible reason for the drop of the model order selection accuracy in the high SNR region is that when the noise level is low, the impact of losing data structure details resulted from data averaging by clustering is severer than the benefit of noise reduction resulted from data averaging, and the performance of model order selection scheme that utilizes estimates of mixing matrix is affected. In most of the cases that the model order was not correctly detected, we found that the model order with minimum normalized instability index is 3, and the difference between the normalized instability indices of model order 3 and ground truth model order 4 is usually very small (less than 0.1 or even 0.05), which means that the 3 order model and the 4 order model are actually both relatively stable in these cases.



(a)



(b)

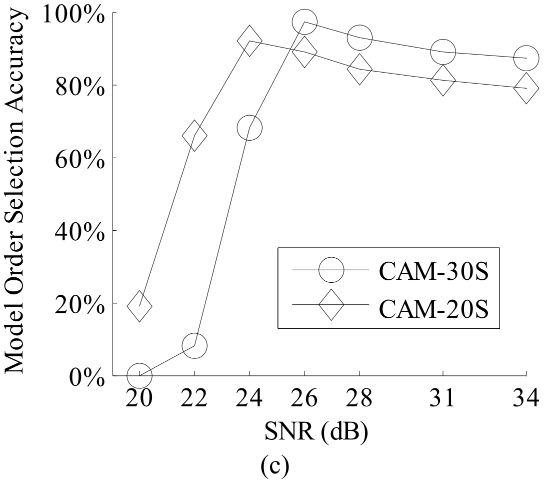


Figure 4.17 Model order selection accuracy of CAM. (a), (b), and (c) are the model order selection accuracies obtained in the determined, over-determined, and under-determined cases, respectively.

4.6 Biological Processes Discovered on Muscle Regeneration Data

Skeletal muscle regeneration is a highly synchronized process involving the activation of various cellular processes. The muscle regeneration is induced by the injection of cardiotoxin into the mouse muscle, which damages the muscle tissue and inducts staged muscle regeneration [79]. The initial phase of muscle regeneration is characterized by necrosis of the damaged tissue and activation of an inflammatory response; and this phase is rapidly followed by activation of myogenic cells to proliferate, differentiate, and fuse, leading to new myofiber formation and reconstitution of a functional contractile apparatus [26, 134].

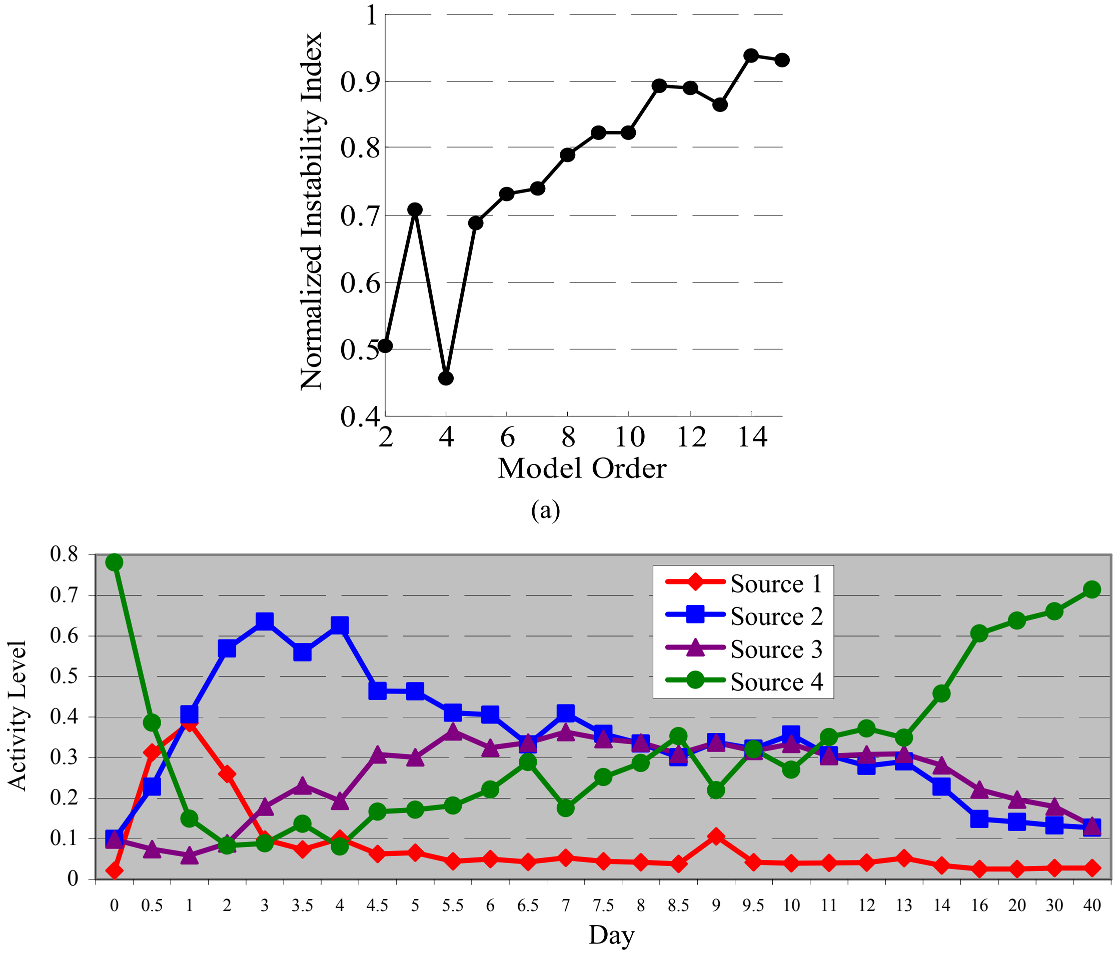
We used CAM to dissect the gene expression time series of muscle regeneration into sources that are gene expression profiles of putative dominant biological processes activated in the muscle regeneration process. Since the biological processes have distinct genomic functions, a reasonable assumption is that each biological process has some genes exclusively highly expressed as compared to their expression values in other biological processes and these genes are the key players for fulfilling the functional roles of the biological process. Apparently, the existence of such genes makes the biological processes approximately well-grounded, which provides the basis for applying CAM. By exploiting the existing biological knowledge on the recovered sources and mixing matrix, we shall ask whether the recovery results are biologically plausible.

The muscle regeneration data contain 54 microarray gene expression profiles that were taken at 27 successive time points, two replicates at each time point [79]. After pre-filtering by “present call”, 7570 genes were believed to be significantly present and were used for gene expression dissection [79]. We took the average of the two profiles at each time point and formed a dataset containing 27 profiles. When applying CAM on the dataset, 40% of the genes whose vector norms were sufficiently small were excluded for the estimation of mixing matrix and model order selection. The sector-based clustering chose the optimal result from 30 independent runs with the sector number equal to 30. 30 cross-validation trials were performed for calculating the normalized instability index for model order selection.

The experimental results are shown in Figure 4.18. Figure 4.18a shows the normalized instability index of model order varying from 2 to 15 and indicates that the best model order is 4. Accordingly, we identified 4 sources and their time activity curves (i.e. column vectors of mixing matrix) as shown in Figure 4.18b. For each source, we selected 200 genes most dominated by the source to form its PSG group, where the dominance level of source i on gene n is calculated by $\hat{s}_i[n]/(\mathbf{1}_K^T\hat{\mathbf{s}}[n])$, for $i \in \{1, \ldots, K\}$ and $n \in \{1, \ldots, N\}$. We input the four gene groups into Ingenuity Pathway Analysis (IPA) [78], a comprehensive database of gene regulation networks and gene function annotations, to analyze the genomic functions of the gene groups.

IPA showed that the PSG group of source 1 is associated with inflammatory disease, connective tissue disorders, skeletal and muscular disorders, and immune response with a p-value of 6.77E-39, 9.02E-35, 9.02E-35, and 9.62E-32, respectively. The analysis reveals that these genes play an important role in the necrosis of damaged muscle tissue and the activation of an inflammatory response. Consistently, from Figure 4.18b, we see that source 1 activates immediately after the muscle is damaged and then diminishes quickly.

IPA showed that the PSG group of source 2 is associated with cell cycle, DNA replication & recombination & repair, and cellular growth & proliferation with a p-value of 7.07E-25, 3.77E-17, and 2.10E-8, respectively. The analysis reveals that these genes are actively involved in myogenic cell proliferation to prepare sufficient myoblasts for later differentiation into myocytes. The time activity curve of source 2 peaks from day 2 to day 4 and then goes down.



(a)

(b)

Figure 4.18 Results obtained on the 27 time-point skeletal muscle regeneration dataset. (a) Normalized instability index of different model order. (b) The time activity curves of the four detected sources.

IPA showed that the PSG group of source 3 is associated with tissue development, skeletal & muscular system development, cell to cell signaling & interaction, and connective tissue development & function with a *p*-value of 9.09E-16, 4.91E-11, 2.33E-08, and 4.35E-07, respectively. The analysis reveals that these genes facilitate the differentiation of myoblast into mononucleated myocyte and the fusion of myocytes to form multinucleated myofibers. Consistent with the fact that myogenic cell differentiation and fusion need sufficient myoblasts produced by myogenic cell proliferation, the time activity curve of source 3 goes up later than that of source 2 and keeps a high level from day 5 to day 13. Note that the time activity curve of source 3 goes down after day 13, which is consistent with the observation that muscle regeneration almost accomplishes in two weeks.

IPA showed that the PSG group of source 4 is associated with skeletal & muscular system function and tissue morphology, both with a *p*-value of 3.49E-10. It also associates with many other metabolism related genomic function categories. These genes are typical active genes in normal/regenerated muscle cells, and consistently, we see that the time activity curve of source 4 is high at the beginning (day 0, before the injection of cardiotoxin and damage of muscle tissue) and at the end (day 16 to day 40, when the muscle tissue is regenerated).

Based on the obtained experimental results, we can see that CAM successfully detected four underlying biological processes actively involved in skeletal muscle regeneration. These four biological processes accurately represent the four successive phases of muscle regeneration, i.e., (1) tissue necrosis and immune response, (2) myogenic cell proliferation, (3) myoblast differentiation and fusion to form myofibers, and (4) normal/regenerated muscle tissue. The time activity curves of the detected biological processes are also consistent with the existing understanding about muscle regeneration. The consistency between CAM's recovery results and the prior knowledge shows the potential application of CAM to study biological events about which we do not have much prior knowledge.

We also applied CAM on a newly generated mouse skeletal muscle regeneration dataset, called the acute skeletal muscle regeneration dataset. The dataset was produced by measuring the gene expression levels of mouse muscle tissue in a muscle regeneration process induced by the injection of notexin into mouse muscle. This dataset was provided by our collaborators, Dr. Eric P. Hoffman's group, in Children's National Medical Center and the experiment focused on the time period from day 2 to day 4 after the injection of toxin. The mice were sacrificed at each of the following 10 time points: 0day, 2day, 2.25day, 2.5day, 2.75day, 3day, 3.25day, 3.5day, 3.75day, and 4day. For each time point, there are 2~4 replicates. The gene expression microarrays used in this experiment were

Affymetrix mouse 430_2. We took the average of the replicates of each time point, and formed a dataset containing 10 profiles and 45037 probe sets.

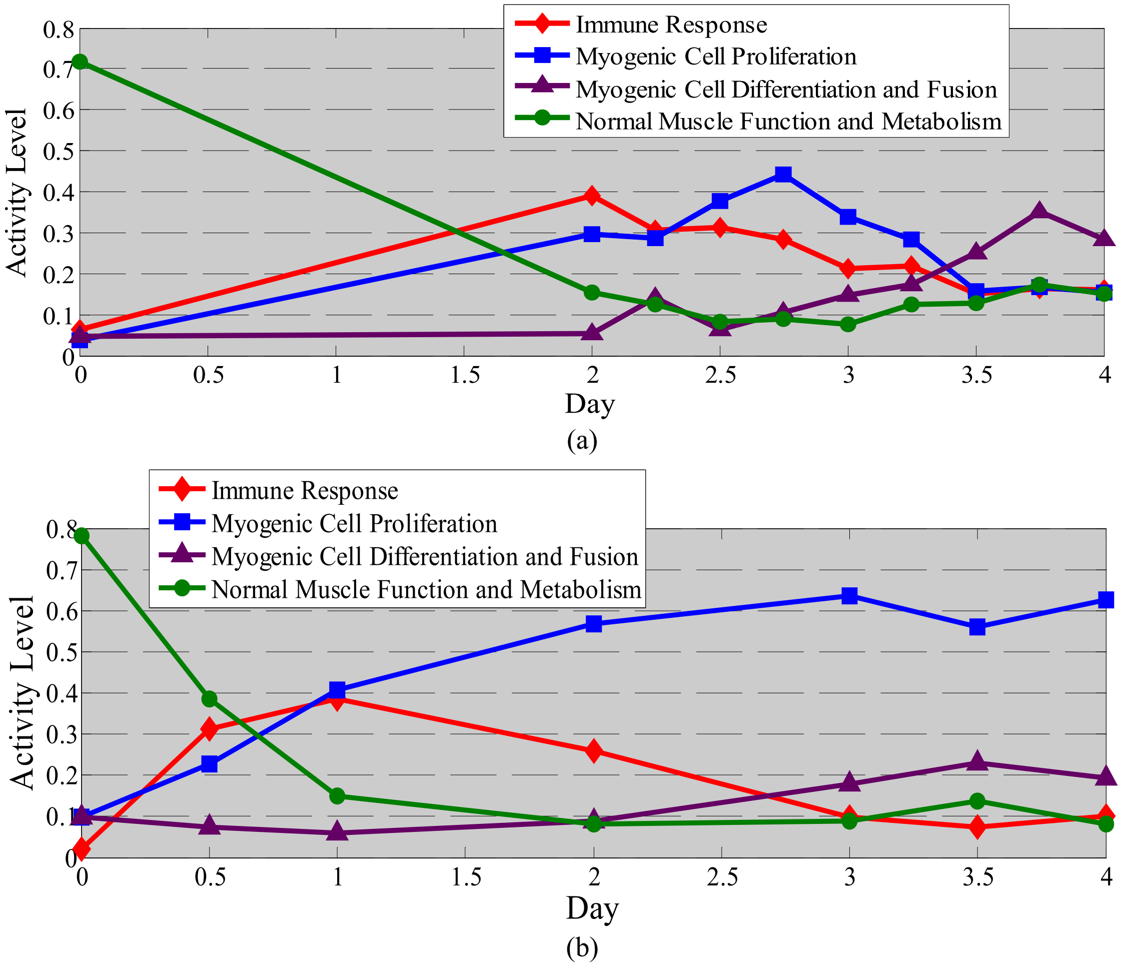


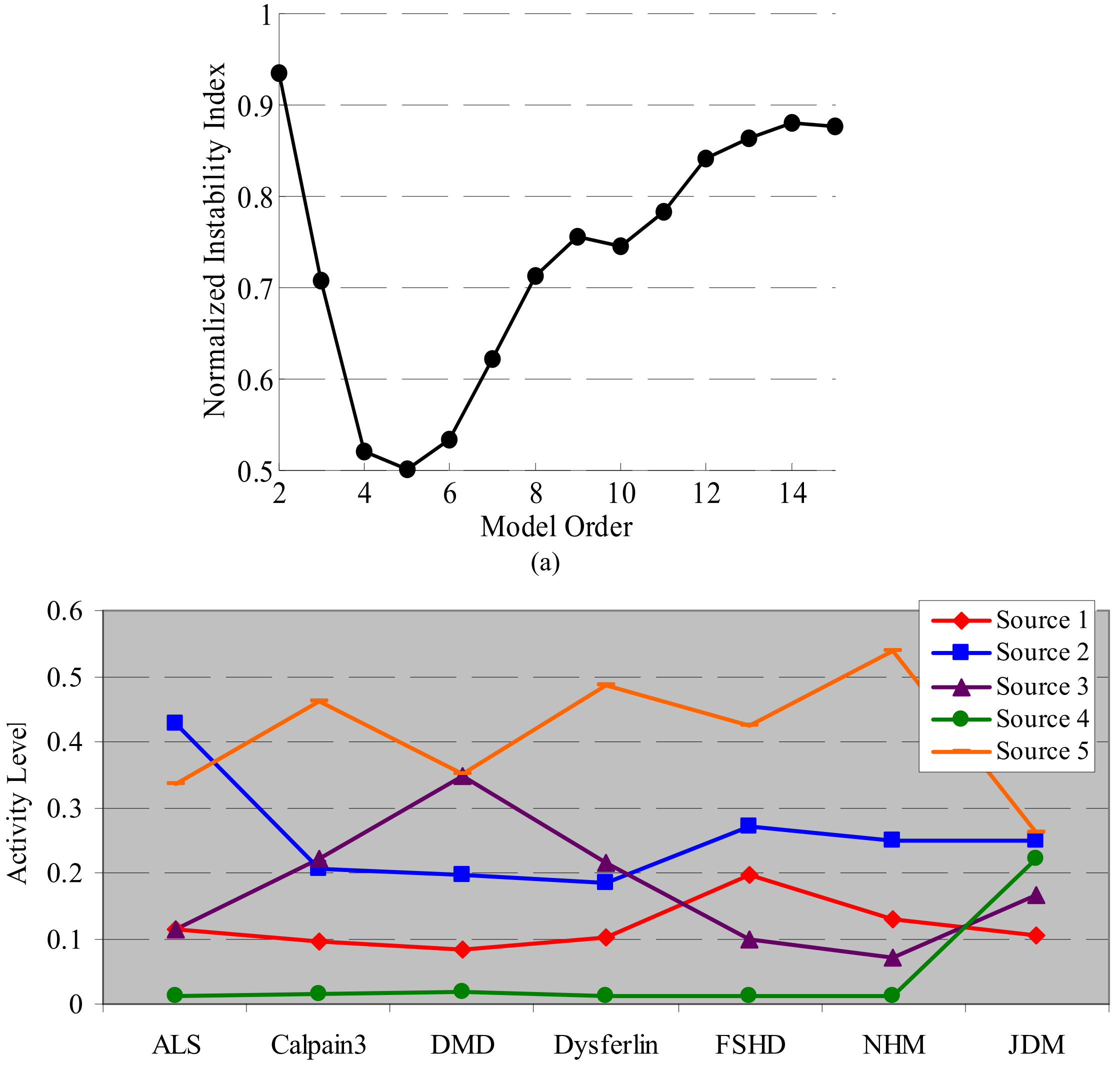
Figure 4.19 Comparison between the time activity curves of biological processes obtained on the two skeletal muscle regeneration datasets. (a) The time activity curves of biological processes obtained on the acute muscle regeneration dataset. (b) The time activity curves of biological processes obtained on the original 27 time-point muscle regeneration dataset.

When applying CAM on the dataset, 20% of the genes whose vector norms were sufficiently small and 3% of the genes whose norms were significantly large were excluded in the estimation of mixing matrix and the model order selection. The sector-based clustering chose the optimal result from 30 independent runs with the sector number equal to 60. 30 cross-validation trials were performed for calculating the normalized instability index for model order selection.

CAM identified five sources on the acute skeletal muscle regeneration dataset. Again for each source, we used IPA to analyze the top 200 source-dominant genes. We found that four out of the five sources have statistically significantly associated genomic functions. These genomic functions are also consistent with the four biological processes discovered on the original 27 time-point muscle regeneration dataset, i.e. immune response, myogenic cell proliferation, myogenic cell differentiation and fusion, and normal muscle function and metabolism. Each of the four sources is associated with and represents one biological process. We present the time activity curves of these four biological processes obtained on the acute skeletal muscle regeneration dataset in Figure 4.19a. For the purpose of comparison, we also show the time activity curves of these biological processes obtained on the original 27 time-point muscle regeneration dataset in Figure 4.19b, while focusing on the period between day 0 and day 4. We can see that the time activity curves of the biological processes obtained on the two different datasets are quite similar. The only exception is the time activity curve of myogenic cell proliferation (i.e. the blue lines in Figure 4.19). It goes down earlier on the acute muscle regeneration dataset. A possible reason is that in the experiment of generating the acute muscle regeneration dataset, notexin was used instead of cardiotoxin to damage the mouse muscle, where notexin is a relatively mild toxin compared to cardiotoxin. So the muscle tissue might not be fully damaged and thus the myogenic cell proliferation was not required as much as it was required in the original experiment using cardiotoxin. Accordingly, we see that the obtained activity level of myogenic cell proliferation goes down earlier on the acute muscle regeneration dataset.

4.7 Biological Processes Discovered on Muscular Dystrophy Data

From the muscular dystrophy dataset introduced in Table 2.1, we selected seven phenotypes, namely, ALS, Calpain3, DMD, Dysferlin, FSHD, NHM, and JDM, for our study. For every selected phenotype, we took the average of the profiles from the phenotype to form a dataset containing seven gene expression profiles and 22215 genes. In the application of CAM to this dataset, 30% of the genes whose vector norms were sufficiently small and 1% of the genes whose vector norms were significantly large were excluded in the estimation of mixing matrix and the model order selection. The sector-based clustering chose the optimal result from 30 independent runs with the sector number equal to 40. 30 cross-validation trials were performed for calculating the normalized instability index for model order selection.

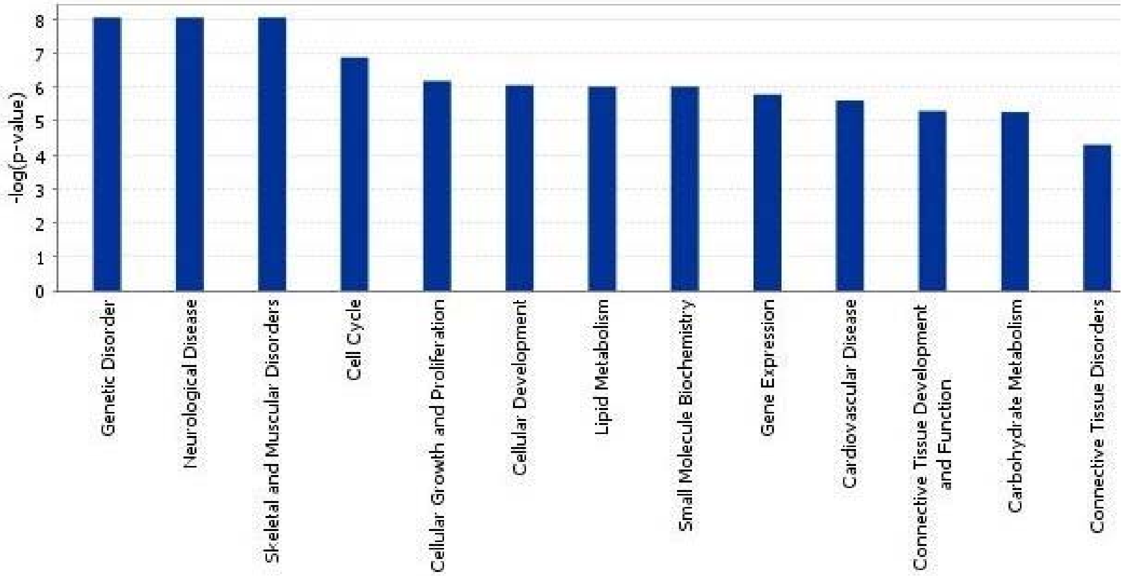


(a)

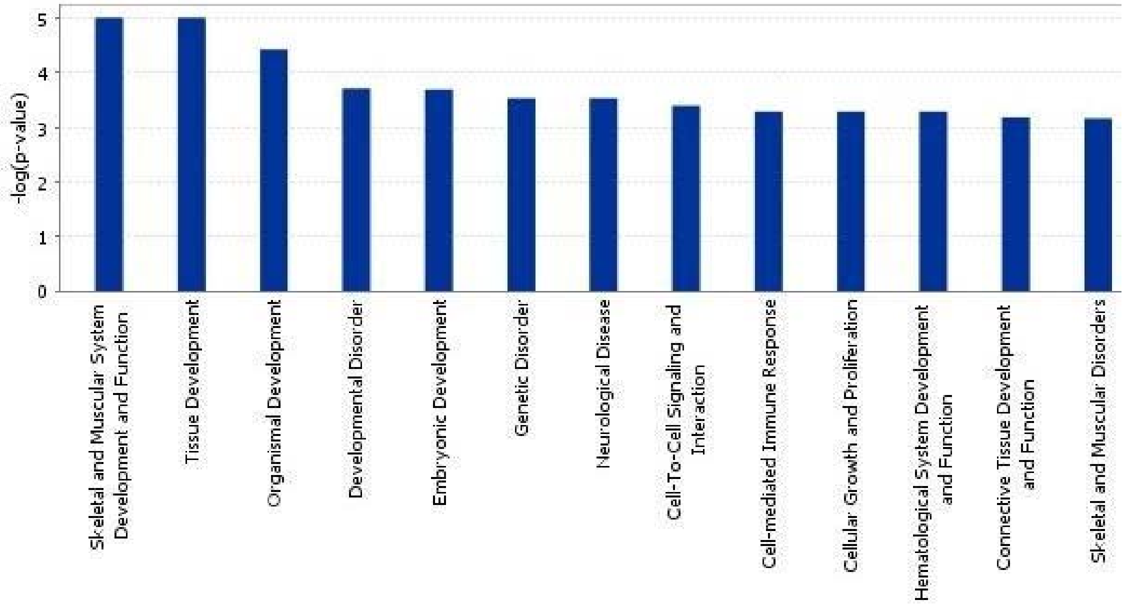
(b)

Figure 4.20 Identification results obtained on the seven-phenotype muscular dystrophy data. (a) Normalized instability index of different model order. (b) Activity levels of the five sources in different phenotypes.

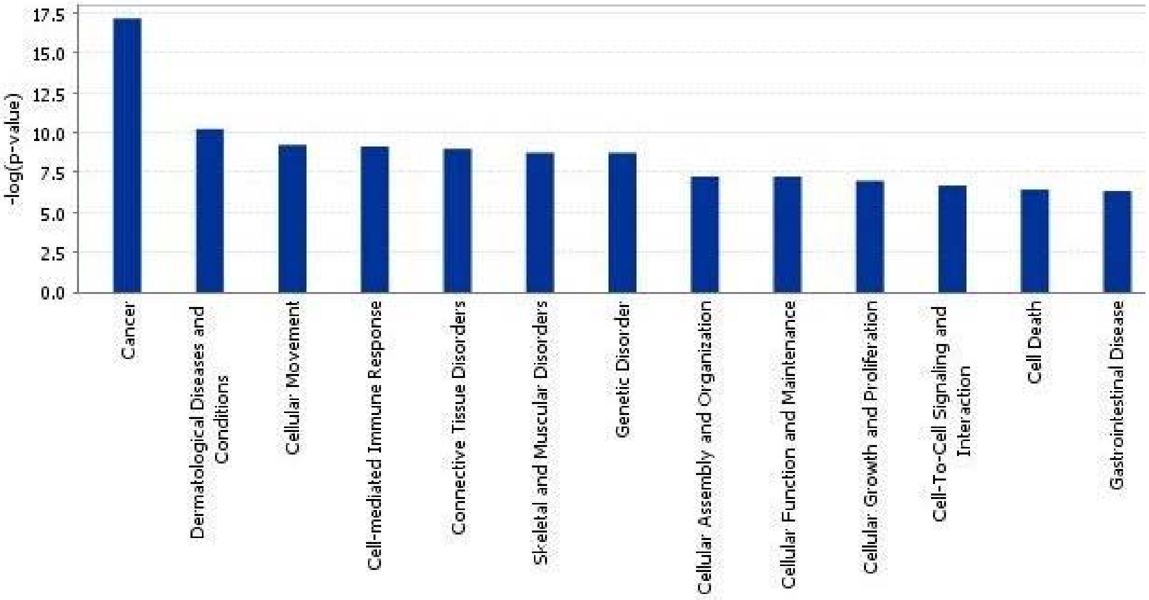
CAM discovered five sources on this dataset. The instability index of different model orders is shown in Figure 4.20a, and the model with five sources shows the maximum stability. The activity levels of the five identified sources across different phenotypes are shown in Figure 4.20b. For each identified source, we selected top 200 source-dominant genes to form its PSG group and analyzed their genomic functions using IPA **[78]**, and present the results in Figure 4.21.



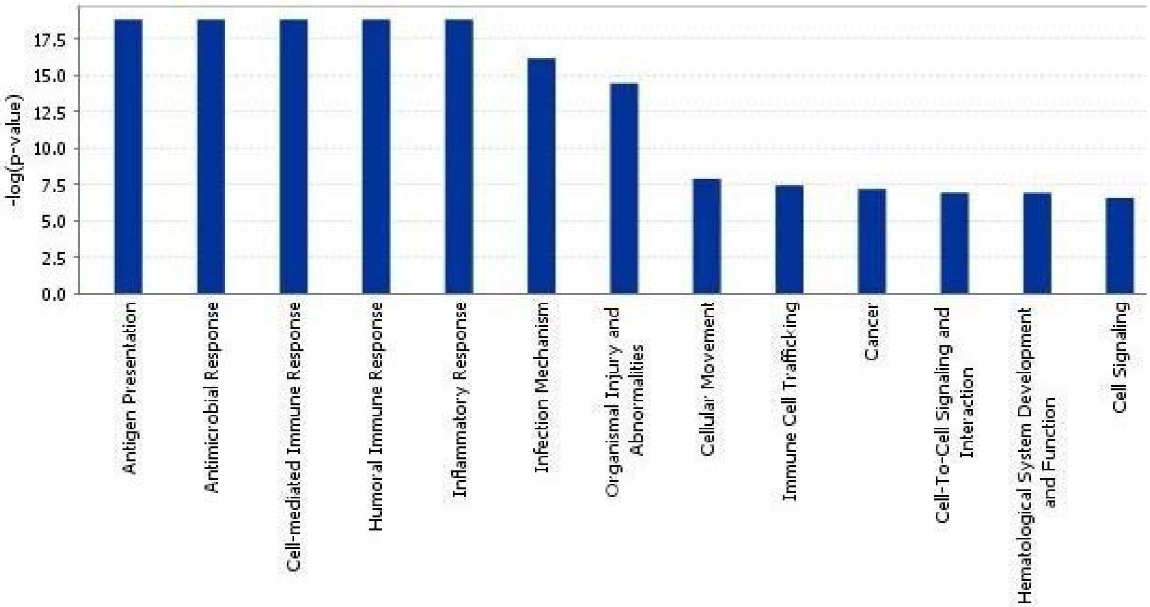
(a)



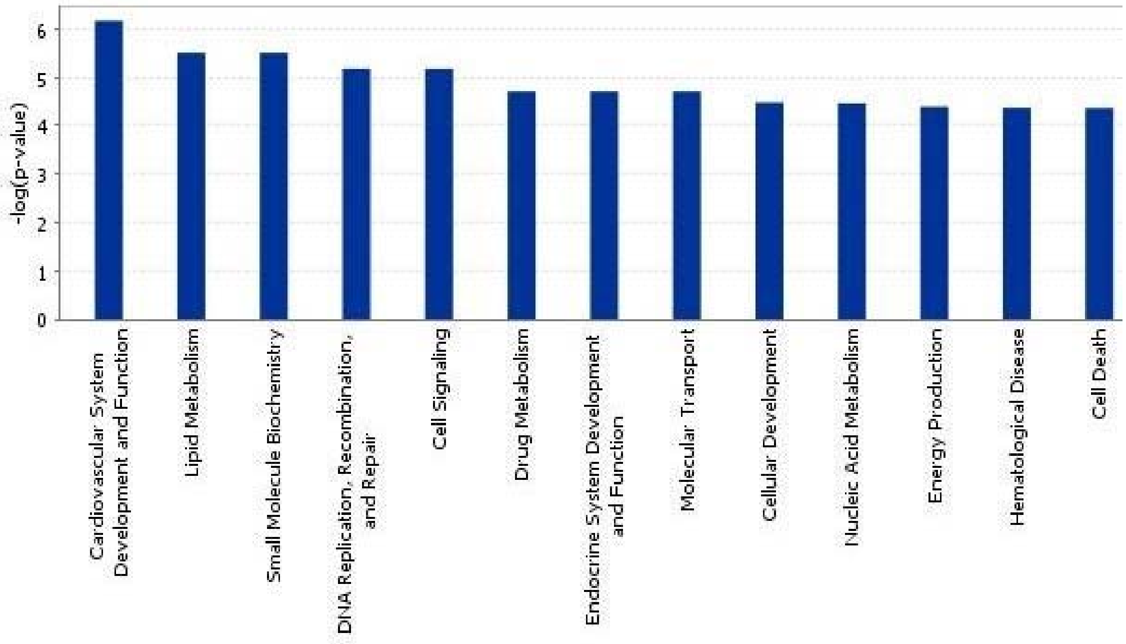
(b)


-log(p-value)
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Cancer
Dermatological Diseases and Conditions
Cellular Movement
Cell-mediated Immune Response
Connective Tissue Disorders
Skeletal and Muscular Disorders
Genetic Disorder
Cellular Assembly and Organization
Cellular Function and Maintenance
Cellular Growth and Proliferation
Cell-To-Cell Signaling and Interaction
Cell Death
Gastrointestinal Disease

(c)


-log(p-value)
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Antigen Presentation
Antimicrobial Response
Cell-mediated Immune Response
Humoral Immune Response
Inflammatory Response
Infection Mechanism
Organismal Injury and Abnormalities
Cellular Movement
Immune Cell Trafficking
Cancer
Cell-To-Cell Signaling and Interaction
Hematological System Development and Function
Cell Signaling

(d)



(e)

Figure 4.21 The most significant genomic function categories associated with the PSG groups of the five identified sources. (a), (b), (c), (d), and (e) show the negative log *p*-values of the PSG group of source 1, 2, 3, 4, and 5 being associated with various genomic function categories, respectively. The analysis results were generated through the use of IPA (Ingenuity® Systems, www.ingenuity.com).

The activity level of source 1 is similar in almost all phenotypes except FSHD, in which a relatively higher activity level is observed. From Figure 4.21a, we can see that one of the three most significant genomic function categories associated with the PSGs of source 1 is skeletal and muscular disorders. The activity level of source 2 is high in ALS and similarly low in all other phenotypes. The two genomic function categories most significantly associated with the PSGs of source 2 are skeletal & muscular system development & function and tissue development. The activity level of source 3 achieves its highest point in DMD. Connective tissue disorders and skeletal & muscular disorders are among the most significant genomic function categories associated with the PSGs of source 3. Source 4 is active only in JDM. Its associated genomic functions include antimicrobial response, cell-mediated immune response, humoral immune response, inflammatory response, infection mechanism, and organismal injury & abnormalities, which is consistent with our domain knowledge that JDM is a childhood autoimmune disorder associated with viral infections that stimulate muscle destruction by inflammatory cells and ischemic processes. The activity level of source 5 is high in NHM that is actually normal muscle tissue and relatively low in JDM and DMD that are dystrophic phenotypes associated with significant failed muscle regeneration due to fibrosis. Among the most significant gene function categories associated with the PSGs of source 5, we see lipid metabolism, nucleic acid metabolism, and energy production that are basic biological functions of normal muscle cells and tissue.

4.8 Discussion

For the performance evaluation and comparison, we set the testing SNR region at 19dB ~ 34dB. At SNR levels (much) lower than 19dB, CAM's performance will also be significantly affected by noise and gradually become similar to those of some competing methods like N-FINDR, SNICA, NMF, and SNMF, and thus no method will perform reasonably well. When the SNR level increases beyond 34dB, the performance of the competing methods will gradually approach their performance obtained on noise-free data shown in Sub-section 4.3.4. The performance curves of NMF and SNMF go up earlier than those of N-FINDR and SNICA when SNR increases, because the later two are sensitive to even small noise.

By comparing the performances of CAM-20S and CAM-30S, we found that modeling data with more sectors is suitable for data containing less noise. An extreme case is that the data are noise-free, where the suitable sector number for applying CAM is the sample number, i.e. every sample forms its own sector. Then the CAM framework degenerates into the edge detection algorithm proposed in Table 4.1. If the noise level in the data is known or can be estimated a priori, a suitable sector number may be selected. For applications where the noise level is unknown or cannot be accurately estimated, all different sector numbers in a reasonably big range may enable CAM to obtain good recovery accuracy, although finding the optimum sector number for applying CAM still requires further research efforts. For example, in our evaluation experiments, CAM-20S and CAM-30S both outperform the competing methods, which leads to a reasonable conjecture that CAM with any sector number between 20 and 30 will maintain a good outperformance compared to the competing methods.

The sector-based clustering algorithm follows a K-means type scheme that randomly chooses samples to initialize the sector central rays. Its clustering outcome is sensitive to the algorithm initialization and may converge to local optimum of the objective function to be minimized. So we run the algorithm multiple times with random initialization and select the best outcome according to the objective function value. An alternative approach to obtain good clustering outcome is to use a more sophisticated initialization scheme instead of random initialization. For example, we randomly initialize a large number (much bigger than K_s) of sector central rays. Based on the initialization, a sector partition of the data is generated by the sector-based clustering algorithm. Then we remove the sector central ray, whose data sector has the smallest sample size, and use the remaining sector central rays to initialize the sector-based clustering algorithm and re-cluster the data. This process ends until only K_s sector central rays remain and serve as the initialization of the algorithm. Another initialization scheme that can be used is to first calculate/estimate the density/population of samples with different vector directions and the sector central rays are then initialized by vector directions with the biggest sample densities. It shall be noticed that although we can use the sophisticated initialization schemes to possibly improve the performance of sector-based clustering, it is still not guaranteed that the obtained result will reach the global optimum of the objective function.

The current sector-based clustering algorithm utilized by CAM monotonically minimizes the summation of squared distance from the sample to its sector central ray and reaches a local minimum of the summation when it converges. Considering samples with the same vector direction but different vector norms, since the distance from the sample to its sector central ray is proportional to the sample norm, large-norm samples have a bigger contribution to the clustering objective function than small-norm samples. As we discussed before, when the noise is small and stationary with a zero mean, large-norm samples potentially have a high SNR and thus are more reliable, which also supports the use of the proposed sector-based clustering algorithm. However, if the noise is not stationary with a zero mean, the large-norm samples may also be heavily influenced by noise and hence jeopardize the clustering outcome due to their significant contribution to the clustering objective function. Here, we propose another sector-based clustering scheme, where the clustering outcome is not affected by the sample norm, and thus may be preferred in the case that large-norm samples do not have a higher SNR than small-norm samples. This alternative clustering scheme first positively scales all samples to have a unit norm, so that all samples are located on a hypersphere with a unit radius, and then, uses K-means clustering with Euclidian distance as the dissimilarity measure to cluster the samples. The resulted cluster centers indicate the directions of samples in the clusters.

In introduction section of this chapter, we have discussed that besides PSGs there are a large number of PCGs that have similar gene expression levels in all biological processes and also some intermediate genes between PSGs and PCGs (see Figure 4.1). The PCGs are commonly required by the different biological processes for the basic cellular function and structure based on which the PSGs can function properly to fulfill the distinct genomic function of the biological process. However, for the relationship between PSGs and PCGs (including the intermediate genes), there is another interpretation which is that the PSGs are the driving forces that form the observed biological phenomenon and regulate the PCGs and the intermediate genes, and the source values in the linear mixture model actually indicate the regulation strengths. As we can see that these two different interpretations of the relationship between PSGs and PCGs (including intermediate genes) both use the same mathematical form (i.e. the linear mixture model), but have different cause-effect directions, i.e. PCGs supporting PSGs (including intermediate genes) vs. PSGs regulating PCGs (including intermediate genes). Although in this chapter we mainly follow the first interpretation, we can not decide which interpretation is correct purely based on the mathematical form, and it is also possible that both mechanisms function simultaneously in real biological systems.

The source well-groundedness is a critical assumption and foundation for the CAM model and its solution. The identifiability of CAM model relies on source well-groundedness. However, it will be interesting to discuss how the CAM solution handles the nBSS problem when the sources are not (fully) well-grounded, which may happen in real-world data analysis tasks. Figure 4.22 shows a noise-free dataset that does not satisfy source well-groundedness. The samples are perspectively projected onto the cross-section of the cone formed by the mixing matrix column vectors. The dashed lines are the boundary of the cone and all samples are confined within the cone. We can see that there is no sample at the corner of the triangle formed by the dashed lines due to the absence of WGPs. We apply the edge detection algorithm on the data and obtain 15 edges indicated by the red circle markers. Taking the detected edges as a candidate pool, in which we search for an edge triplet with the minimum model fitting error calculated by Equation (4.24). Notice that here each sample forms its own singleton data sector, because on noise-free data we do not need to apply the sector-based clustering, which is equivalent to that each sample forms its won data sector. The optimum edge triplet contains the three edges pointed by the orange arrows in Figure 4.22. We can see that they are the samples closest to the mixing matrix column vectors.

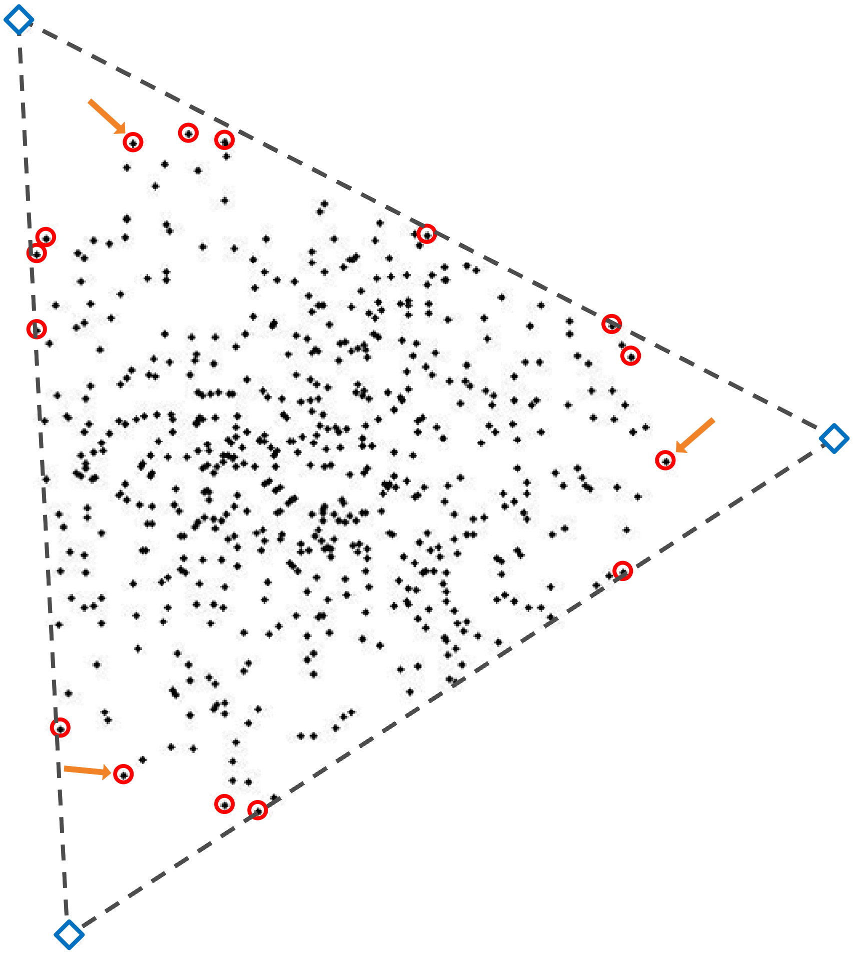


Figure 4.22 Illustration of a noise-free situation when sources are not well-grounded. The samples are perspectively projected onto the cross-section of the cone formed by the mixing matrix column vectors

whose positions are indicated by the blue diamond markers. Dashed lines indicate the boundary of the cone. Each black point is a sample. Red circle markers indicate the edges obtained by applying the edge detection algorithm on the data.

4.9 Conclusion

We develop CAM to blindly separate non-negative well-grounded sources based on the observed mixture samples. CAM does not require that the sources are independent, uncorrelated, or overall sparse. The identifiability of the noise-free CAM model is proven through a set of theorems. For real noisy data analysis tasks, CAM recovers the mixing matrix by identifying/estimating the WGPs, suppresses the noise effect using a comprehensive noise reduction scheme including sector-based clustering, and selects the optimum model order through stability analysis. In the under-determined case, where the sources are usually unidentifiable, CAM can estimate the sources with good accuracy for the particular task of gene expression dissection.

On numerically mixed gene expression data, CAM showed significantly improved accuracy of recovering mixing matrix, sources, and distinct patterns of sources, as compared to several benchmark nBSS methods, i.e. nICA, SNICA, NMF, SNMF, and N-FINDR. CAM also showed good model order selection accuracy in the determined, over-determined, and under-determined cases. When applied on muscle regeneration gene expression data, CAM detected four biological processes actively involved in skeletal muscle regeneration. The genomic functions and the time activity curves (i.e. mixing matrix column vectors) of the identified biological processes are also consistent with existing knowledge about the skeletal muscle regeneration process. When applied on muscular dystrophy data containing multiple muscular dystrophies, CAM detected biological processes closely related to the disease mechanisms of some of the muscular dystrophies.

5 Contributions, Future Work, and Conclusions

5.1 Summary of Original Contributions

In this dissertation, we developed data modeling and analysis methods for learning statistical and geometric models from microarray gene expression data and subsequently discovered data structure and information associated with underlying disease mechanisms. The original contributions of this research work are summarized as follows.

5.1.1 Comprehensive and Effective Cluster Modeling, Visualization, and Discovery

We developed VISDA for cluster modeling, visualization, and discovery on gene expression data. VISDA addresses some of the major limitations associated with existing clustering methods and improves the accuracy and biological relevance of clustering outcome. VISDA performs progressive, coarse-to-fine divisive hierarchical clustering and visualization to discover hidden clusters within complex, high-dimensional data, supported by hierarchical mixture modeling, informative gene selection, data visualization by projections, and user/prior knowledge guidance through human-data interactions. Multiple structure-preserving projections, each sensitive to a distinct type of clustering tendency, are used to reveal potential cluster/sub-cluster structure. Human intelligence and database knowledge are incorporated into the clustering process for model initialization, data visualization, and cluster validation. Cluster number detection on the high-dimensional data is achieved by Bayesian theoretic criteria and user justification applied via a hierarchy of low-dimensional, locally-discriminative subspaces.

VISDA is comprehensively suitable for multiple data clustering tasks, including gene clustering, sample clustering, and phenotype clustering, albeit with customized modifications for each of these tasks. The main clustering and visualization algorithm of VISDA is readily applicable for gene clustering where the attribute-to-object ratio is low. To apply VISDA to sample clustering, where the attribute-to-object ratio is high, we designed a front-end dimensionality reduction via unsupervised informative feature/gene selection by variation filtering and discriminative power analysis. By exploiting the knowledge of phenotype labels in performing supervised informative feature/gene selection, supervised data visualization, and statistical modeling that preserves the unity of samples from the same phenotype, we also extended VISDA algorithm for discovering and visualizing the relative relationship between known phenotypes in the dataset.

We applied VISDA to muscular dystrophy, muscle regeneration, and cancer gene expression data analysis. The obtained results demonstrated VISDA's capability to identify functionally enriched or co-regulated gene groups, discover/validate disease phenotypes/sub-phenotypes, and capture the pathological relationship between multiple diseases reflected at the mRNA level.

5.1.2 Ground Truth Based Clustering Evaluation and Comparison

We developed an objective and reliable clustering evaluation scheme based on sample clustering and definitive ground truth (phenotype categories) to assess the performance of clustering algorithms in analyzing gene expression data. Using this evaluation scheme, we conducted a comparative study on the functionality, accuracy, and stability/reproducibility of five clustering methods, namely, VISDA, HC, KMC, SOM, and SFNM fitting, tested on seven published microarray gene expression datasets and one synthetic dataset. The experimental results reflected both existing knowledge and novel insights about the competing clustering methods.

VISDA was shown to be a stable performer with the highest partition accuracy among the competing methods. The model order selection scheme in VISDA was shown to be effective for high-dimensional gene expression data clustering. By examining the variations of partition accuracy and cluster number detection accuracy obtained by multiple users with and without significant experience in data clustering analysis, we found that VISDA does not require special skills of users and common sense about cluster structure and data visualization is sufficient for using VISDA.

5.1.3 Convex Analysis of Mixtures for Non-negative Well-grounded Source Separation

We designed a latent linear mixture model to dissect the gene expression data into components that are putative underlying biological processes that drive/form the observed biological event. Sources in the mixture model are assumed to be non-negative and well-grounded, which is a realistic assumption consistent with existing understanding about biological processes and possessing certain advantages over some existing nBSS models/methods that make biologically implausible assumptions about biological processes.

Identifiability of the noise-free model was proven by a series of theorems. An efficient edge detection algorithm was developed based on convex analysis and optimization to identify the model. For real noisy non-negative well-grounded source separation tasks, we developed CAM, a robust nBSS framework, to recover mixing matrix and sources based on noise-reduction data preprocessing, sector-based clustering, convex analysis based edge detection, and minimization of model fitting error. Stability analysis based model order selection scheme was designed to detect the source number. For the under-determined case, where the sources are usually unidentifiable, an iterative algorithm was also developed to estimate the sources with good accuracy for the particular task of gene expression dissection.

Based on separating numerically mixed gene expression data, CAM demonstrated improved accuracies of recovering mixing matrix, sources, and distinct patterns of sources over several benchmark nBSS methods, i.e. nICA, SNICA, NMF, SNMF, and N-FINDR. CAM also achieved good model order selection accuracy in all of the determined, over-determined, and under-determined cases. When applied to skeletal muscle regeneration gene expression data, CAM successfully identified the four biological processes actively involved in the skeletal muscle regeneration process, as well as their time activity curves. When applied to muscular dystrophy data containing multiple muscular dystrophies, CAM detected biological processes closely related to the disease mechanisms of some of the muscular dystrophies.

5.2 Future Work

This section outlines several remaining problems/topics to be further explored, which have emerged for consideration during the research work of this dissertation. The discussion presented here can be viewed as a starting point for future research.

5.2.1 Possible Improvements on VISDA and Ensemble Clustering

VISDA provides users with an extensible visualization capability by a projection suite that can incorporate complementary and effective projection methods to increase the likelihood of revealing the data/application-dependent cluster structure of interest. As we have mentioned in Section 2.8, there are various visualization techniques for gene expression data analysis [17, 37, 83, 84], and new and effective visualization techniques are expected to keep emerging. So one of the possible improvements on VISDA in future research is to incorporate novel projection methods that are effective and complementary to the ones already utilized by VISDA. And there is no theoretical barrier to use 3-D visualization and parameter initialization instead of the current 2-D visualization and parameter initialization in VISDA. Keeping one more dimension for visualization will significantly increase the information that visualization delivers. For 3-D visualization, we can also allow the user to rotate the visualization and change the view angle, so that the problem caused by occlusion can be alleviated.

In sample clustering, dimension reduction via unsupervised informative gene selection is a very challenging task due to no prior knowledge and potentially complex gene-gene interactions embedded within high-dimensional data. The current unsupervised informative gene selection scheme for sample clustering used in VISDA is still a pilot study whose success is often highly data-dependent and limited. Since unsupervised gene/feature selection and sample clustering are usually a chicken-egg problem, i.e. without first possessing correct sample clusters, it is difficult to identify relevant genes, and without good gene selection to eliminate many noisy/irrelevant ones, it is very difficult to discern the true underlying sample cluster structure, wrapping gene selection around sample clustering in an iterative, refining procedure may be a good strategy for future improvement on VISDA's sample clustering algorithm [31, 46, 50].

It is well known that clustering algorithms always reflect some structural bias associated with the involved grouping principle [13, 16, 28]. Thus it is recommended that for a new dataset without much prior knowledge one should try several different clustering methods or use an ensemble scheme that combines the results of different algorithms [41]. Ensemble clustering produces analysis result by fusing multiple clustering outcomes obtained through one or a combination of the following three schemes.

(1) Apply multiple algorithms with different cluster models/assumptions, e.g. connectivity/path based clustering vs. Gaussian mixture modeling based clustering.

(2) Apply multiple runs of a single clustering algorithm on different subsets of the data, such as the consensus clustering scheme proposed in [37].

(3) Apply multiple runs of a single clustering algorithm with different parameter setting, such as running KMC on a dataset for multiple times with random cluster center initialization.

Clustering ensembles have been shown to be a powerful tool for improving the adaptivity, robustness, as well as stability of clustering solution [37, 43, 62, 63].

VISDA presently assumes that each cluster follows a Gaussian distribution largely driven by mathematical convenience. Small sample size problem can easily defeat this assumption and clusters in some datasets may have a data distribution far away from Gaussian distribution, for example, a cluster with its data points forming a ring [43]. Therefore, a potential future improvement is to incorporate into the VISDA framework multiple different cluster models, each sensitive to a distinct type of clustering tendency, to improve VISDA’s ability to adapt to the underlying data characteristics to increase the likelihood of robustly detecting cluster structures relevant to the study interest. Fusing multiple runs of VISDA for an ensemble outcome is also a potential improvement option.

5.2.2 Alternative Model Order Selection Scheme for Non-negative Well-grounded Source Separation

In CAM, the source number is determined by the stability analysis based model order selection, where we examine the stability of models with different number of sources in the presence of data perturbation and the "correct" model order is defined as that of the most stable model. The stability analysis based model order selection of CAM has certain advantages. For example, (1) it is uniformly applicable to the determined, over-determined, and under-determined cases; (2) it does not require statistical modeling, which is important for applications where the data distribution cannot be well-modeled by simple parametric probabilistic models; and (3) it does not require any preset threshold that may be subjective. However, although stability/reproducibility is critical for any scientific discovery, a problem associated with the stability analysis based model order selection is that the "correct" model order being associated with maximum stability has not been theoretically proved. Actually, from the point of view of signal detection and estimation, an estimate can be very stable (e.g. having a small variance) but biased [96]. So practically, to improve the reliability of model order selection of CAM, we may need an alternative model order selection scheme that can be used besides the stability analysis based scheme to confirm or validate its result.

One potential alternative model order selection scheme is based on a simple principle/assumption that among all the edges of the mixture sample cone only edges associated with true sources are salient. An edge being salient means that the difference between the cones generated by the edge set before and after the particular edge is removed is significant, while the difference can be measured in multiple ways. For example, we can measure the change of the solid angle of cone. Or as we have reviewed in Section 4.1.1, we can perspectively project the mixture samples onto a hyperplane, in which the projected mixture samples form a convex hull and the edge points, i.e. WGPs, become convex hull vertices. Then the saliency of an edge can be measured by its corresponding vertex's contribution to the volume of the convex hull, which is the decrease of the convex hull's volume if the vertex is removed. The assumption that only edges associated with true sources are salient is valid when the mixing matrix column vectors (i.e. edges) have very different vector directions and the noise level is not too high. Besides designing a suitable saliency measure, we also need to design an efficient and effective scheme to search for the salient edge set. A possible scheme may be that we one-by-one remove the least salient edge. At each time after an edge is removed, we recalculate the saliency measure of each remaining edge based on its influence on the cone (or convex hull) generated by all remaining edges and then further remove the least salient one, until a stop criterion or threshold is met, and thus the number of salient edges, i.e. source number, is determined.

Another potential alternative model order selection scheme is to first fit the data using probabilistic models with different number of sources and then use information theoretical criteria such as MDL to select the optimal model and determine the source number. Here, we propose a probability density function for fitting the gene expression patterns of underlying biological processes in the source scatter plot. This probability density function is a mixture of two components. One component is an exponential distribution with independent variables primarily for the purpose of modeling the PSGs and the very dense part of the gene distribution that is close to the origin of coordinates in the scatter plot (see Figure 4.1). The second component is a Gaussian distribution centered on the diagonal of the first quadrant to model the other genes in the middle of the scatter plot. And because gene expressions can only be non-negative, this probability density function has non-zero values only in the first quadrant. Equation (5.1) gives the probability density function

$$\mathrm{f}\left(\mathbf{s}[n] \,\|\, \boldsymbol{\pi}, \boldsymbol{\lambda}, \mu, \boldsymbol{\Sigma}\right) = \begin{cases} \pi_{\mathrm{e}} \prod_{i=1}^{K} \lambda_i \exp\left(-\lambda_i s_i[n]\right) + \pi_{\mathrm{g}}\, \mathrm{g}\left(\mathbf{s}[n] \mid \mu \mathbf{1}_K, \boldsymbol{\Sigma}\right), & \text{for } \mathbf{s}[n] \geq \mathbf{0}_K \\ 0, & \text{o.w.} \end{cases}, \quad (5.1)$$

where $\mathbf{s}[n]$ is the nth source sample, $s_i[n]$ is the ith entry of $\mathbf{s}[n]$, K is the number of sources, π_{e} and π_{g} are the sample proportions of the two components in the mixture, λ_i is the parameter associated with the exponential distribution at the ith dimension, $\mu\mathbf{1}_K$ and $\boldsymbol{\Sigma}$ are the parameters associated with the Gaussian distribution, and exp(•) and g(•) are the probability density function of the exponential distribution and Gaussian distribution, respectively. The proposed probability density function has the flexibility/complexity necessary for fitting the gene expression data well in the scatter plot. But since it is not a simple parametric probabilistic model, fitting this model to the observed gene expression data that is after mixing and has additive noise remains a challenging problem to be solved.

5.2.3 Modeling of Regulation Relationship between Discovered Biological Processes

Using CAM, we can identify the underlying active biological processes that are the “driving forces” forming the observed dynamic biological event, as well as the time activity curves of the biological processes. Identification of the biological processes and their time activity curves answers the question -- what are the components and their activation dynamics that may have caused the observed biological event, but to thoroughly understand the mechanism of the biological system we need to further study the inter-relationship between the biological processes, which is a reverse engineering problem. Based on the recovered time activity curves of the biological processes, we can use dynamic models like recurrent neural networks and state space models [131] to model and learn the regulation and signaling relationship between the biological processes. The learned model will not only reveal potential interactions between the biological processes, but also enables us to do in silico experiments of manipulating the biological system, such as knocking out a particular biological process, for the purpose of studying the system, and in silico experiments are usually much easier, cheaper, and faster than real biological experiments.

5.3 Conclusions

Microarray gene expressions are important data source for disease study at the molecular and genomic level. Due to the huge data size and significant noise in the data, computational methods are needed to extract from gene expression data meaningful and specific information about the biological system being studied. In this dissertation, we discussed three research topics, i.e., cluster modeling, visualization, and discovery for providing a high-level overview of gene expression data, clustering evaluation for guiding users in the selection of clustering algorithm for gene expression data analysis, and gene expression dissection for identification of underlying biological processes that jointly form the observed biological event. To address these research problems, we developed novel and effective data clustering and visualization algorithm, clustering evaluation scheme, and non-negative blind source separation method. These methods and schemes realized new data analysis functionalities, overcame some of the major limitations of existing methods, and achieved superior performances over some existing benchmark methods, which also contributed to the research progress in machine learning and pattern recognition fields. Data analysis results obtained by our proposed methods demonstrated their ability to extract important information from microarray gene expression data to improve the understanding about disease mechanisms and stimulate novel hypotheses for further research. Additionally, several remaining research problems were also proposed for future study.

Appendix A Proof of Lemma 4.1

First, we prove that (A4) is a sufficient condition for $\{\mathbf{A}\}$ to constitute the unique edges of cone($\{\mathbf{A}\}$) up to a positive scaling. Based on (A4), we have the following inference. For any $\mathbf{a}_i$ ($i \in \{1, \ldots, K\}$), since $\mathbf{a}_i \in$ cone($\{\mathbf{A}\}$), we have $\mathbf{a}_i = \mathbf{A}\boldsymbol{\alpha}$ ($\boldsymbol{\alpha} = [\alpha_1 \ldots \alpha_K]^T \in \mathrm{R}_+^K$), which is equivalent to

$$\left(1-\alpha_i\right)\mathbf{a}_i = \mathbf{A}_{(i)}\boldsymbol{\alpha}_{(i)}. \tag{6.1}$$

α_i must be 1, otherwise Equation (6.1) indicates $\mathbf{a}_i \in$ cone($\{\mathbf{A}_{(i)}\}$) or $\mathbf{a}_i \in -$ cone($\{\mathbf{A}_{(i)}\}$), which conflicts with (A4). Substituting 1 for α_i in Equation (6.1), we have $\mathbf{A}_{(i)}\boldsymbol{\alpha}_{(i)} = \mathbf{0}_M$, which indicates $\boldsymbol{\alpha}_{(i)} = \mathbf{0}_{K-1}$, because otherwise some vector in $\{\mathbf{A}_{(i)}\}$ will be a non-positive combination of other vectors in $\{\mathbf{A}_{(i)}\}$ and (A4) is violated. Therefore, $\mathbf{a}_i$ can only be a trivial non-negative combination of $\{\mathbf{A}\}$ and thus it is an edge of cone($\{\mathbf{A}\}$). So $\{\mathbf{A}\}$ are edges of cone($\{\mathbf{A}\}$). Moreover, by the definition of edge, any edge of cone($\{\mathbf{A}\}$) must be a positive scaling of a vector in $\{\mathbf{A}\}$. Therefore, $\{\mathbf{A}\}$ are the unique edges of cone($\{\mathbf{A}\}$) up to a positive scaling.

Second, we prove that (A4) is a necessary condition for $\{\mathbf{A}\}$ to constitute the unique edges of cone($\{\mathbf{A}\}$). Notice that any vector in $\{\mathbf{A}\}$ is not a positive scaling of another vector in $\{\mathbf{A}\}$, otherwise the model is degenerated. Suppose that (A4) is not satisfied, i.e. $\exists\, \mathbf{a}_i \in \{\mathbf{A}\}$, and $\mathbf{a}_i \in$ cone($\{\mathbf{A}_{(i)}\}$) or $\mathbf{a}_i \in -$ cone($\{\mathbf{A}_{(i)}\}$). If $\mathbf{a}_i \in$ cone($\{\mathbf{A}_{(i)}\}$), which means $\mathbf{a}_i = \mathbf{A}_{(i)}\boldsymbol{\beta}$ ($\boldsymbol{\beta} \in \mathrm{R}_+^{K-1}$ and $\boldsymbol{\beta} \neq \mathbf{0}_{K-1}$), then $\mathbf{a}_i = (\mathbf{A}_{(i)}\boldsymbol{\beta} + \mathbf{a}_i) / 2$; or if $\mathbf{a}_i \in -$ cone($\{\mathbf{A}_{(i)}\}$), which means $\mathbf{a}_i = -\,\mathbf{A}_{(i)}\boldsymbol{\beta}$ ($\boldsymbol{\beta} \in \mathrm{R}_+^{K-1}$ and $\boldsymbol{\beta} \neq \mathbf{0}_{K-1}$), then $\mathbf{a}_i = \mathbf{A}_{(i)}\boldsymbol{\beta} + 2\mathbf{a}_i$. Clearly, in both of these two cases, $\mathbf{a}_i$ can be represented by a non-trivial non-negative combination of $\{\mathbf{A}\}$. Therefore, $\mathbf{a}_i$ is not an edge of cone($\{\mathbf{A}\}$). Violating (A4) leads to that at least one vector in $\{\mathbf{A}\}$ is not an edge of cone($\{\mathbf{A}\}$). So (A4) is a necessary condition for $\{\mathbf{A}\}$ to constitute the unique edges of cone($\{\mathbf{A}\}$).

Appendix B Proof of Lemma 4.2

Any vector **v** ∈ cone({**X**}) can be represented by **v** = **Xα** = **ASα** ($\boldsymbol{\alpha} \in \mathrm{R}_{+}^{N}$). Because $\boldsymbol{\alpha} \in \mathrm{R}_{+}^{N}$ and $\mathbf{S} \in \mathrm{R}_{+}^{K\times N}$, we have $\mathbf{S}\boldsymbol{\alpha} \in \mathrm{R}_{+}^{K}$. By the definition of finite cone, **v** ∈ cone({**A**}). Therefore, cone({**X**}) ⊆ cone({**A**}).

Since ($A2$) is satisfied, let $\{n_1, \ldots, n_K\}$ be the indices of a WGP set, where $\mathbf{x}[n_i]$ is a WGP of source i ($i \in \{1, \ldots, K\}$). The mixing matrix **A** can be represented by

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_K \end{bmatrix} = \begin{bmatrix} \mathbf{x}[n_1]/s_1[n_1] & \cdots & \mathbf{x}[n_K]/s_K[n_K] \end{bmatrix}, \tag{6.2}$$

Any vector $\boldsymbol{\gamma} \in$ cone({**A**}) can be represented by $\boldsymbol{\gamma} = \mathbf{A}\boldsymbol{\beta}$, where $\boldsymbol{\beta} = [\beta_1 \ldots \beta_K]^T \in \mathrm{R}_{+}^{K}$. Replace **A** with its representation in Equation (6.2), we have

$$\boldsymbol{\gamma} = \mathbf{A}\boldsymbol{\beta} = \begin{bmatrix} \mathbf{x}[n_1] & \cdots & \mathbf{x}[n_K] \end{bmatrix} \begin{bmatrix} \beta_1/s_1[n_1] & \cdots & \beta_K/s_K[n_K] \end{bmatrix}^T, \tag{6.3}$$

where $[\beta_1/s_1[n_1] \cdots \beta_K/s_K[n_K]]^T \in \mathrm{R}_{+}^{K}$. Therefore, $\boldsymbol{\gamma} \in$ cone({$\mathbf{x}[n_1], \cdots, \mathbf{x}[n_K]$}) ⊆ cone({**X**}), which indicates cone({**A**}) ⊆ cone({**X**}).

Because cone({**X**}) ⊆ cone({**A**}) and cone({**A**}) ⊆ cone({**X**}), we reach the conclusion of cone({**A**}) = cone({**X**}).

Appendix C Proof of Theorem 4.2

To prove Theorem 4.2, we need to utilize the fact that (A4) ensures $\mathbf{x}[n] \notin -\mathrm{cone}(\{\mathbf{X}_{(n)}\})$, i.e. any vector in $\{\mathbf{X}\}$ is not a non-positive combination of other vectors in $\{\mathbf{X}\}$, which can be proved by the following inference. Suppose $\mathbf{x}[n] \in -\mathrm{cone}(\{\mathbf{X}_{(n)}\})$, which means $\mathbf{x}[n] = \mathbf{A}\mathbf{s}[n] = -\mathbf{X}_{(n)}\boldsymbol{\alpha} = -\mathbf{A}\mathbf{S}_{(n)}\boldsymbol{\alpha}$ ($\boldsymbol{\alpha} \in \mathrm{R}_{+}^{N-1}$, $\boldsymbol{\alpha} \neq \mathbf{0}_{N-1}$). Then we can derive

$$\mathbf{A}\left(\mathbf{s}[n] + \mathbf{S}_{(n)}\boldsymbol{\alpha}\right) = \mathbf{0}_M \tag{6.4}$$

where $\mathbf{s}[n] + \mathbf{S}_{(n)}\boldsymbol{\alpha} \in \mathrm{R}_{+}^{K}$ and $\mathbf{s}[n] + \mathbf{S}_{(n)}\boldsymbol{\alpha} \neq \mathbf{0}_K$, which indicates that some column vector of $\mathbf{A}$ is a non-positive combination of other column vectors of $\mathbf{A}$ and conflicts with the second part of the statement of (A4). Therefore, (A4) guarantees $\mathbf{x}[n] \notin -\mathrm{cone}(\{\mathbf{X}_{(n)}\})$.

The proof of Theorem 4.2 is accomplished by two steps. First, we prove that $\angle(\mathbf{x}[n], \mathbf{x}'[n]) > 0$ is a sufficient condition for $\mathbf{x}[n]$ being an edge of cone($\{\mathbf{X}\}$). Because $\mathbf{x}[n] \in \mathrm{cone}(\{\mathbf{X}\})$, $\mathbf{x}[n]$ can be represented by $\mathbf{x}[n] = \mathbf{X}\boldsymbol{\beta}$ ($\boldsymbol{\beta} = [\beta_1 \cdots \beta_N]^T \in \mathrm{R}_{+}^{N}$), from which we derive

$$\mathbf{x}[n](1 - \beta_n) = \mathbf{X}_{(n)}\boldsymbol{\beta}_{(n)} \tag{6.5}$$

As we have mentioned before, $\angle(\mathbf{x}[n], \mathbf{x}'[n]) > 0$ is equivalent to $\mathbf{x}[n] \notin \mathrm{cone}(\{\mathbf{X}_{(n)}\}$. So β_n must be 1, otherwise Equation (6.5) will indicate $\mathbf{x}[n] \in \mathrm{cone}(\{\mathbf{X}_{(n)}\})$ or $\mathbf{x}[n] \in -\mathrm{cone}(\{\mathbf{X}_{(n)}\})$. Substituting 1 for β_n in Equation (6.5), we get $\mathbf{X}_{(n)}\boldsymbol{\beta}_{(n)} = \mathbf{0}_M$, which indicates $\boldsymbol{\beta}_{(n)} = \mathbf{0}_{N-1}$, because otherwise some vector in $\{\mathbf{X}_{(n)}\}$ will be a non-positive combination of other vectors in $\{\mathbf{X}_{(n)}\}$. So $\boldsymbol{\beta}$ must be equal to $\mathbf{e}_n$ and $\mathbf{x}[n]$ must be a trivial combination of $\{\mathbf{X}\}$, which means that $\mathbf{x}[n]$ is an edge of cone($\{\mathbf{X}\}$).

Second, we prove that $\angle(\mathbf{x}[n], \mathbf{x}'[n]) > 0$ is a necessary condition for $\mathbf{x}[n]$ being an edge of cone($\{\mathbf{X}\}$). Suppose $\angle(\mathbf{x}[n], \mathbf{x}'[n]) = 0$, which is equivalent to $\mathbf{x}[n] \in \mathrm{cone}(\{\mathbf{X}_{(n)}\})$. $\mathbf{x}[n]$ can be represented by $\mathbf{x}[n] = \mathbf{X}_{(n)}\boldsymbol{\gamma}$ ($\boldsymbol{\gamma} \in \mathrm{R}_{+}^{N-1}$ and $\boldsymbol{\gamma} \neq \mathbf{0}_{N-1}$). Then we have $\mathbf{x}[n] = (\mathbf{x}[n] + \mathbf{X}_{(n)}\boldsymbol{\gamma}) / 2$, which represents $\mathbf{x}[n]$ by a non-trivial non-negative combination of $\{\mathbf{X}\}$, because all vectors in $\{\mathbf{X}_{(n)}\}$ are not a positive scaling of $\mathbf{x}[n]$. So $\mathbf{x}[n]$ is not an edge of cone($\{\mathbf{X}\}$). Therefore, if $\mathbf{x}[n]$ is an edge of cone($\{\mathbf{X}\}$), $\angle(\mathbf{x}[n], \mathbf{x}'[n])$ must be bigger than 0.

Appendix D Proof of Theorem 4.3

For a linear mixture model **X** = **AS** that satisfies (*A*1), (*A*2), and (*A*3), because (*A*3) is sufficient for (*A*4) to hold, according to Remark 4.1, **A** can be identified up to a permutation and a positive scaling of column vectors through edge detection. Then **S** can be identified through Equation (4.7). Assume that there is another set of sources denoted by **T** that satisfy (*A*1), (*A*2), and **X** = **AT**. Because **X** = **AS** = **AT**, we derive that

$$\mathbf{0}_{M\times N} = \mathbf{AS} - \mathbf{AT} = \mathbf{A}\left(\mathbf{S} - \mathbf{T}\right) \tag{6.6}$$

Since **A** has a full column rank, **S** − **T** must be equal to $\mathbf{0}_{K\times N}$, i.e., **T** is the same as **S**. So **S** is also identifiable up to a positive scaling and a permutation of its row vectors.

Appendix E Proof of Theorem 4.4

Step 2 of the sector-based clustering algorithm fixes {$\mathbf{r}_1, \ldots, \mathbf{r}_{Ks}$} and minimizes the distance between the mixture sample and its sector central ray by changing the sector membership of the mixture samples, thus also minimizes the total summation in Equation (4.23). The algorithm step 3 fixes the mixture samples' sector memberships and minimizes $\sum_{h=1}^{N_j} \left\| \mathbf{x}[n_h] - \mathbf{r}_j^T \mathbf{x}[n_h] \mathbf{r}_j \right\|^2$ by updating $\mathbf{r}_j$, thus also minimizes the total summation. Step 2 and step 3 iteratively, monotonically minimize the summation of square distances. Obviously, this summation must have a low bound. So when the algorithm converges, it will reach a local minimum, if not a global minimum.

Bibliography

[1] E. Segal, N. Friedman, N. Kaminski, A. Regev, and D. Koller, "From signatures to models: understanding cancer using microarrays," *Nature Genetics,* vol. 37, pp. 38-45, 2005.

[2] V. E. Velculescu, L. Zhang, B. Vogelstein, and K. W. Kinzler, "Serial analysis of gene expression," *Science,* vol. 270, pp. 484-7, 1995

[3] E. R. Mardis, "Next-Generation DNA Sequencing Methods," *Annual Review of Genomics and Human Genetics,* vol. 9, pp. 387-402, 2008.

[4] Affymetrix, "Guide to probe logarithmic intensity error (PLIER) estimation," edited by Affymetrix I, Santa Clara, CA 2005.

[5] Y. Zhao, M. Li, and R. Simon, "An adaptive method for cDNA microarray normalization," *BMC Bioinformatics,* vol. 6, 2005.

[6] D. K. Slonim, "From patterns to pathways: gene expression data analysis comes of age," *Nature Genetics,* vol. 32, pp. 502-8, 2002.

[7] C. Li and W. H. Wong, "Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application," *Genome Biology,* vol. 2, pp. research0032.1-research0032.11, 2001.

[8] J. D. Storey, W. Xiao, J. T. Leek, R. G. Tompkins, and R. W. Davis, "Significance analysis of time course microarray experiments," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 102, pp. 12837-42, 2005.

[9] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub, "Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 96, pp. 2907-12, 1999.

[10] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science,* vol. 286, pp. 531-7, 1999.

[11] K. A. Shedden, J. M. Taylor, T. J. Giordano, R. Kuick, D. E. Misek, G. Rennert, D. R. Schwartz, S. B. Gruber, C. Logsdon, D. Simeone, S. L. Kardia, J. K. Greenson, K. R. Cho, D. G. Beer, E. R. Fearon, and S. Hanash, "Accurate molecular classification of human cancers based on gene expression using a

simple classifier with a pathological tree-based framework," *American Journal of Pathology,* vol. 163, pp. 1985-95, 2003.

[12] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman, "Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data," *Nature Genetics,* vol. 34, pp. 166-76, 2003.

[13] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys,* vol. 31, pp. 264-323, 1999.

[14] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine,* vol. 7, pp. 673-9, 2001.

[15] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *Journal of Computational Biology,* vol. 6, pp. 281-97, 1999.

[16] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: a survey," *IEEE Transactions on Knowledge and Data Engineering,* vol. 16, pp. 1370-86, 2004.

[17] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 95, pp. 14863-8, 1998.

[18] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: tool for the unification of biology," *Nature Genetics,* vol. 25, pp. 25-9, 2000.

[19] S. I. Lee and S. Batzoglou, "Application of independent component analysis to microarrays," *Genome Biology,* vol. 4, Oct. 24, 2003.

[20] A. E. Teschendorff, M. Journée, P. A. Absil, R. Sepulchre, and C. Caldas, "Elucidating the altered transcriptional programs in breast cancer using independent component analysis," *PLoS Computational Biology,* vol. 3, pp. 1539-54, Aug. 2007.

[21] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis,* 1st ed., Wiley-Interscience, 2001.

[22] T.-H. Chan, W.-K. Ma, C.-Y. Chi, and Y. Wang, "A convex analysis framework for blind separation of non-negative sources," *IEEE Transactions on Signal Processing,* vol. 56, pp. 5120-34, 2008.

[23] T. Gong, J. Xuan, C. Wang, H. Li, E. P. Hoffman, R. Clarke, and Y. Wang, "Gene module identification from microarray data using nonnegative independent component analysis," *Gene Regulation and Systems Biology,* vol. 1, pp. 349-63, 2007.

[24] C. J. Wu, Y. Fu, T. M. Murali, and S. Kasif, "Gene expression module discovery using gibbs sampling," *Genome Informatics,* vol. 15, pp. 239-48, 2004.

[25] D. J. Miller, Y. Wang, and G. Kesidis, "Emergent unsupervised clustering paradigms with potential application to bioinformatics," *Frontiers in Bioscience,* vol. 13, pp. 677-90, 2008.

[26] M. Bakay, Z. Wang, G. Melcon, L. Schiltz, J. Xuan, P. Zhao, V. Sartorelli, J. Seo, E. Pegoraro, C. Angelini, B. Shneiderman, D. Escolar, Y. Chen, S. T. Winokur, L. M. Pachman, C. Fan, R. Mandler, Y. Nevo, E. Gordon, Y. Zhu, Y. Dong, Y. Wang, and E. P. Hoffman, "Nuclear envelope dystrophies show a transcriptional fingerprint suggesting disruption of Rb-MyoD pathways in muscle regeneration," *Brain,* vol. 129, pp. 996-1013, 2006.

[27] Y. Zhu, Z. Wang, Y. Feng, J. Xuan, D. J. Miller, E. P. Hoffman, and Y. Wang, "Phenotypic-specific gene module discovery using a diagnostic tree and caBIG^{TM} VISDA," in *28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 06)*, New York City, 2006, pp. 5767-70.

[28] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks,* vol. 16, pp. 645-78, 2005.

[29] K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo, "Model-based clustering and data transformations for gene expression data," *Bioinformatics,* vol. 17, pp. 977-87, 2001.

[30] W. Pan, "Incorporating gene functions as priors in model-based clustering of microarray gene expression data," *Bioinformatics,* vol. 22, pp. 795-801, 2006.

[31] V. Roth and T. Lange, "Bayesian class discovery in microarray datasets," *IEEE Transactions on Biomedical Engineering,* vol. 51, pp. 707-18, 2004.

[32] C. Huttenhower, A. Flamholz, J. Landis, S. Sahi, C. Myers, K. Olszewski, M. Hibbs, N. Siemers, O. Troyanskaya, and H. Coller, "Nearest neighbor networks: clustering expression data based on gene neighborhoods," *BMC Bioinformatics,* vol. 8, 2007.

[33] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church, "Systematic determination of genetic network architecture," *Nature Genetics,* vol. 22, pp. 281-5, 1999.

[34] A. Gasch and M. Eisen, "Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering," *Genome Biology,* vol. 3, pp. 1-22, 2002.

[35] D. Dembele and P. Kastner, "Fuzzy C-means method for clustering microarray data," *Bioinformatics,* vol. 19, pp. 973-80, 2003.

[36] L. Fu and E. Medico, "FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data," *BMC Bioinformatics,* vol. 8, 2007.

[37] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data," *Machine Learning Journal,* vol. 52, pp. 91-118, 2003.

[38] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proceedings of the IEEE,* vol. 86, pp. 2210-39, 1998.

[39] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, 1995.

[40] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed., John Wiley & Sons Inc., 2001.

[41] Y. Zhu, Z. Wang, D. J. Miller, R. Clarke, J. Xuan, E. P. Hoffman, and Y. Wang, "A ground truth based comparative study on clustering of gene expression data," *Frontiers in Bioscience,* vol. 13, pp. 3839-49, 2008.

[42] B. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.

[43] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, pp. 835-50, 2005.

[44] J. Rissanen, "Modeling by shortest data description," *Automatica,* vol. 14, pp. 465-71, 1978.

[45] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics,* vol. 6, pp. 461-4, 1978.

[46] M. W. Graham and D. J. Miller, "Unsupervised learning of parsimonious mixtures on large spaces with integrated feature and component selection," *IEEE Transactions on Signal Processing,* vol. 54, pp. 1289-303, 2006.

[47] A. Bertoni and G. Valentini, "Model order selection for bio-molecular data clustering," *BMC Bioinformatics,* vol. 8, 2007.

[48] A. Bertoni and G. Valentini, "Discovering multi-level structures in bio-molecular data through the Bernstein inequality," *BMC Bioinformatics,* vol. 9, 2008.

[49] T. Lange, V. Roth, M. L. Braun, and J. M. Buhmann, "Stability-based validation of clustering solutions," *Neural Computation,* vol. 16, pp. 1299-323, 2004.

[50] E. P. Xing and R. M. Karp, "CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts," *Bioinformatics,* vol. 17, pp. 306-15, 2001.

[51] M. Benito, J. Parker, Q. Du, J. Wu, D. Xiang, C. M. Perou, and J. S. Marron, "Adjustment of systematic microarray data biases," *Bioinformatics,* vol. 20, pp. 105-14, 2004.

[52] R. Clarke, H. W. Ressom, A. Wang, J. Xuan, M. C. Liu, E. A. Gehan, and Y. Wang, "The properties of high-dimensional data spaces: implications for exploring gene and protein expression data," *Nature Reviews Cancer,* vol. 8, pp. 37-49, 2008.

[53] Y. Wang, D. J. Miller, and R. Clarke, "Approaches to working in high dimensional data spaces: gene expression microarray," *British Journal of Cancer,* vol. 98, pp. 1023-8, 2008.

[54] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. S. Furey, M. Ares, and D. Haussler, "Knowledge-based analysis of microarray gene expression data using support vector machines," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 97, pp. 262-7, 2000.

[55] Y. Qu and S. Xu, "Supervised cluster analysis for microarray data based on multivariate Gaussian mixture," *Bioinformatics,* vol. 20, pp. 1905-13, 2004.

[56] X. Zhu, "Semi-supervised learning literature survey," in *Computer Science Technical Report 1530*, University of Wisconsin, 2006.

[57] Y. Chien, *Interactive Pattern Recognition*, Marcel Dekker, 1978.

[58] J. Zou and G. Nagy, "Human-computer interaction for complex pattern recognition problems," in *Data Complexity in Pattern Recognition*, Springer, 2006, pp. 271-86.

[59] C. M. Bishop and M. E. Tipping, "A hierarchical latent variable model for data visualization," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, pp. 282-93, 1998.

[60] M. Tipping and C. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation,* vol. 11, pp. 443-82, 1999.

[61] Y. Wang, L. Luo, M. T. Freedman, and S. Kung, "Probabilistic principal component subspaces: a hierarchical finite mixture model for data visualization," *IEEE Transactions on Neural Networks,* vol. 11, pp. 625-36, 2000.

[62] A. Strehl and J. Ghosh, "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research,* vol. 3, pp. 583-617, 2002.

[63] A. Topchy, A. Jain, and W. Punch, "Clustering ensembles: models of consensus and weak partitions," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, pp. 1866-81, 2005.

[64] Y. Zhu, H. Li, D. J. Miller, Z. Wang, J. Xuan, R. Clarke, E. P. Hoffman, and Y. Wang, "caBIG^{TM} VISDA: Modeling, visualization, and discovery for cluster analysis of genomic data," *BMC Bioinformatics,* vol. 9, 2008.

[65] J. Wang, H. Li, Y. Zhu, M. Yousef, M. Nebozhyn, M. Showe, L. Showe, J. Xuan, R. Clarke, and Y. Wang, "VISDA: an open-source caBIG^{TM} analytical tool for data clustering and beyond," *Bioinformatics,* vol. 23, pp. 2024-7, 2007.

[66] Z. Wang, Y. Wang, J. Lu, S. Kung, J. Zhang, R. Lee, J. Xuan, J. Khan, and R. Clarke, "Discriminatory mining of gene expression microarray data," *Journal of VLSI Signal Processing Systems,* vol. 35, pp. 255-72, 2003.

[67] X. He and P. Niyogi, "Locality preserving projections," in *Advances in Neural Information Processing Systems*, vol. 16, Cambridge, M.A., MIT Press, 2004.

[68] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, 2000.

[69] M. Loog, R. P. W. Duin, and R. Haeb-Umbach, "Multiclass linear dimension reduction by weighted pairwise fisher criteria," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, pp. 762-6, 2001.

[70] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science,* vol. 315, pp. 972-6, 2007.

[71] Y. Weiss and W. T. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Transactions on Information Theory,* vol. 47, pp. 736-44, 2001.

[72] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering* 2nd ed.: Prentice Hall.

[73] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm.," *Journal of the Royal Statistical Society: Series B,* vol. 34, pp. 1-38, 1977.

[74] F. D. Ridder, R. Pintelon, J. Schoukens, and D. P. Gillikin, "Modified AIC and MDL model selection criteria for short data records," *IEEE Transactions on Instrumentation and Measurement,* vol. 54, pp. 144-50, 2005.

[75] Z. Liang, R. J. Jaszczak, and R. E. Coleman, "Parameter estimation of finite mixtures using the EM algorithm and information criteria with application to medical image processing," *IEEE Transactions on Nuclear Science,* vol. 39, pp. 1126-33, 1992.

[76] D. J. Miller and J. Browning, "A mixture model and EM-based algorithm for class discovery, robust classification, and outlier rejection in mixed labeled/unlabeled data sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, pp. 1468-83, 2003.

[77] T. J. Giordano, K. A. Shedden, D. R. Schwartz, R. Kuick, J. M. G. Taylor, N. Lee, D. E. Misek, J. K. Greenson, S. L. R. Kardia, D. G. Beer, G. Rennert, K. R. Cho, S. B. Gruber, E. R. Fearon, and S. Hanash, "Organ-specific molecular classification of primary lung, colon, and ovarian adenocarcinomas using gene expression profiles," *American Journal of Pathology,* vol. 159, pp. 1231-8, 2001.

[78] Ingenuity® Systems, www.ingenuity.com.

[79] P. Zhao, J. Seo, Z. Wang, Y. Wang, B. Shneiderman, and E. P. Hoffman, "In vivo filtering of in vitro expression data reveals MyoD targets," *Comptes Rendus Biologies,* vol. 326, pp. 1049-65 October-November 2003.

[80] D. A. Bergstrom, B. H. Penn, A. Strand, R. L. Perry, M. A. Rudnicki, and S. J. Tapscott, "Promoter-specific regulation of MyoD binding and signal transduction cooperate to pattern gene experssion," *Molecular Cell,* vol. 9, pp. 587-600, 2002.

[81] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub, "Multiclass cancer diagnosis using tumor gene expression signatures," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 98, pp. 15149-54, 2001.

[82] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research,* vol. 3, pp. 1157-82, 2003.

[83] R. M. Ewing and J. M. Cherry, "Visualization of expression clusters using Sammon's non-linear mapping," *Bioinformatics,* vol. 17, pp. 658-9, 2001.

[84] L. Zhang, A. Zhang, and M. Ramanathan, "VizStruct: exploratory visualization for gene expression profiling," *Bioinformatics,* vol. 20, pp. 85-92, 2004.

[85] Y. Feng, Z. Wang, Y. Zhu, J. Xuan, D. Miller, R. Clarke, E. Hoffman, and Y. Wang, "Learning the tree of phenotypes using genomic data and VISDA," in

Proceedings of the Sixth IEEE Symposium on BioInformatics and BioEngineering, Arlington, VA, USA, 2006, pp. 165-70.

[86] J. Handl, J. Knowles, and D. B. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics,* vol. 21, pp. 3201-12, 2005.

[87] F. Gibbons and F. Roth, "Judging the quality of gene expression-based clustering methods using gene annotation," *Genome Research,* vol. 12, pp. 1574-81, 2002.

[88] I. Gat-Viks, R. Sharan, and R. Shamir, "Scoring clustering solutions by their biological relevance," *Bioinformatics,* vol. 19, pp. 2381-9, 2003.

[89] S. Datta and S. Datta, "Methods for evaluating clustering algorithm for gene expression data using a reference set of functional classes," *BMC Bioinformatics,* vol. 7, 2006.

[90] R. Loganantharaj, S. Cheepala, and J. Clifford, "Metric for measuring the effectiveness of clustering of DNA microarray expression," *BMC Bioinformatics,* vol. 7(Suppl. 2), 2006.

[91] A. Thalamuthu, I. Mukhopadhyay, X. Zheng, and G. Tseng, "Evaluation and comparison of gene clustering methods in microarray analysis," *Bioinformatics,* vol. 22, pp. 2405-12, 2006.

[92] K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo, "Validating clustering for gene expression data," *Bioinformatics,* vol. 17, pp. 309-18, 2001.

[93] R. Shamir and R. Sharan, "Algorithmic approaches to clustering gene expression data," in *Current Topics in Computation Molecular Biology*, MIT Press, 2002, pp. 269-300.

[94] S. Datta and S. Datta, "Comparisons and validation of statistical clustering techniques for microarray gene expression data," *Bioinformatics,* vol. 19, pp. 459-66, 2003.

[95] K. M. Kerr and G. A. Churchill, "Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 98, pp. 8961-5, 2001.

[96] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed., Springer, 1998.

[97] T. Kohonen, *Self-organizing Maps*, 3rd ed., Springer, 2000.

[98] D. M. Titterington, A. F. M. Smith, and U. E. Markov, *Statistical Analysis of Finite Mixture Distributions*, John Wiley, 1985.

[99] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistic Quarterly,* vol. 2, pp. 83-97, 1955.

[100] J. Xuan, Y. Wang, Y. Dong, Y. Feng, B. Wang, J. Khan, M. Bakay, Z. Wang, L. Pachman, S. Winokur, Y. Chen, R. Clarke, and E. Hoffman, "Gene selection for multiclass prediction by weighted fisher criterion," *EURASIP Journal on Bioinformatics and Systems Biology,* vol. 2007, 2007.

[101] J. Xuan, Y. Dong, J. Khan, E. Hoffman, R. Clarke, and Y. Wang, "Robust feature selection by weighted Fisher criterion for multiclass prediction in gene expression profiling," in *International Conference on Pattern Recognition*, 2004, pp. 291-4.

[102] A. I. Su, J. B. Welsh, L. M. Sapinoso, S. G. Kern, P. Dimitrov, H. Lapp, P. G. Schultz, S. M. Powell, C. A. Moskaluk, H. F. J. Frierson, and G. M. Hampton, "Molecular classification of human carcinomas by use of gene expression signatures," *Cancer Research,* vol. 61, pp. 7388-93, 2001.

[103] A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker, and M. Meyerson, "Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 98, pp. 13790-5, 2001.

[104] D. R. Schwartz, S. L. R. Kardia, K. A. Shedden, R. Kuick, G. Michailidis, J. M. G. Taylor, D. E. Misek, R. Wu, Y. Zhai, D. M. Darrah, H. Reed, L. H. Ellenson, T. J. Giordano, E. R. Fearon, S. M. Hanash, and K. R. Cho, "Gene expression in ovarian cancer reflects both morphology and biological behavior, distinguishing clear cell from other poor-prognosis ovarian carcinomas," *Cancer Research,* vol. 62, pp. 4722-9, 2002.

[105] G. Bloom, I. V. Yang, D. Boulware, K. Y. Kwong, D. Coppola, S. Eschrich, J. Quackenbush, and T. J. Yeatman, "Multi-platform, multi-site, microarray-based human tumor classification," *American Journal of Pathology,* vol. 164, pp. 9-16, 2004.

[106] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Computation,* vol. 9, pp. 1483-92, 1997.

[107] S. A. Astakhov, H. Stögbauer, A. Kraskov, and P. Grassberger, "Monte carlo algorithm for least dependent non-negative mixture decomposition," *Analytical Chemistry,* vol. 78, pp. 1620-27, 2006.

[108] E. Oja and M. Plumbley, "Blind separation of positive sources by globally convergent gradient search," *Neural Computation,* vol. 16, pp. 1811-25, 2004.

[109] F.-Y. Wang, C.-Y. Chi, T.-H. Chan, and Y. Wang, "Non-negative least-correlated component analysis for separation of dependent sources by volume

maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2009. (In press).

[110] R. Tauler and B. Kowalski, "Multivariate curve resolution applied to spectral data from multiple runs of an industrial process," *Analytical Chemistry,* vol. 65, pp. 2040-7, 1993.

[111] C. Lawson and R. J. Hanson, *Solving Least-Squares Problems*. New Jersey: Prentice-Hall, 1974.

[112] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature,* vol. 401, pp. 788-91, 1999.

[113] W. Liu, N. Zheng, and X. Lu, "Nonnegative matrix factorization for visual coding," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003.

[114] M. E. Winter, "N-findr: an algorithm for fast autonomous spectral end-member determination in hyperspectral data," in *Proceedings of SPIE Conference on Imaging Spectrometry V*, 1999, pp. 266-75.

[115] J. M. P. Nascimento and J. M. B. Dias, "Vertex component analysis: a fast algorithm to unmix hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 43, pp. 898 - 910, 2005.

[116] F.-Y. Wang, C.-Y. Chi, T.-H. Chan, and Y. Wang, "Blind separation of positive dependent sources by non-negative least-correlated component analysis," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Maynooth, Ireland, 2006, pp. 73-8.

[117] K. Stadlthanner, F. J. Theis, E. W. Lang, A. M. Tomé, C. G. Puntonet, and J. M. Górriz, "Hybridizing sparse component analysis with genetic algorithms for microarray analysis," *Neurocomputing,* vol. 71, pp. 2356-76, 2008.

[118] S. Boyd and L. Vandenberghe, *Convex Optimization* Cambridge University Press, 2004.

[119] QHull, www.qhull.org

[120] C. B. Barber, D. P. Dopkin, and H. Huhdanpaa, "The quickhull algorithm for convex hull," in *Technical report gcg53*, The Geometry Center, University of Minnesota, 1993.

[121] J. V. Leeuwen, *Algorithms and Complexity,* vol. A, MIT Press, 1994.

[122] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Transactions on Signal Processing,* vol. 33, pp. 387-92, 1985.

[123] E. Fishler, M. Grosmann, and H. Messer, "Detection of signals by information theoretic criteria: general asymptotic performance analysis," *IEEE Transactions on Signal Processing,* vol. 50, pp. 1027-36, May 2002.

[124] D. N. Lawley, "Tests of significance of the latent roots of the covariance and correlation matrices," *Biometrica,* vol. 43, pp. 128-36, 1956.

[125] C.-I. Chang and Q. Du, "Estimation of number of spectrally distinct signal sources in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 42, pp. 608-19, 2004.

[126] Y. Wang, J. Zhang, J. Khan, R. Clarke, and Z. Gu, "Partially-independent component analysis for tissue heterogeneity correction in microarray gene expression analysis," in *IEEE 13th Workshop on Neural Networks for Signal Processing (NNSP)*, Toulouse, France, 2003, pp. 23-32.

[127] Y. Zhu, T. Chan, E. P. Hoffman, and Y. Wang, "Gene expression by non-negative well-grounded source separation," in *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing*, Cancún, Mexico, 2008.

[128] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software,* vol. 11-12, pp. 625-53, 1999.

[129] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications,* vol. 284, pp. 193-228, 1998.

[130] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2001.

[131] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed., Prentice Hall, 1999.

[132] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Transactions on Computers,* vol. C-26, pp. 917-22, Sept. 1977.

[133] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letter,* vol. 15, pp. 1119-25, 1994.

[134] S. B. Chargé and M. A. Rudnicki, "Cellular and molecular regulation of muscle regeneration," *Physiological Reviews,* vol. 84, pp. 209-38, Jan. 2004