Bicluster Analysis of Biomedical Data based on Multi-objective Evolutionary Optimization

Bicluster Analysis of Biomedical Data based on Multi-objective Evolutionary Optimization

Maryam Golchin
M.Sc.

School of Information and Communication Technology
Gold Coast Campus
Griffith University

Submitted in fulfilment of the requirements of the degree of

Doctor of Philosophy

January 2018

Statement of Originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

____

Maryam Golchin

January 25, 2014

Dedicated to my parents, family, and dear friends

for their unconditional love and support

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Associate Professor Alan Wee-Chung Liew for the continuous support of my Ph.D study, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my PhD study.

I am also grateful to the following university staff: Lauren Holness, Maree Hubbard, Kate Schurmann, and Victoria Wheeler for their unfailing support and assistance.

Furthermore, I would like to acknowledge Griffith University for the financial support during this study and for providing the financial supports to attend several conferences.

I thank my friends in Gold Coast and Brisbane for all the fun we have had in the past four years to make my PhD life easier during the stressful times.

Last but not the least; I would like to thank my family: my parents and my brothers for supporting me emotionally and spiritually throughout writing this thesis and my life in general.

Abstract

Knowledge discovery is the process of finding hidden knowledge from a large volume of data that involves data mining. Data mining unveils interesting relationships among data and the results can help to make valuable predictions or recommendation in various applications. Recently, biclustering has become a common method in data mining and pattern recognition. Biclustering is an unsupervised machine learning method that can uncover and extract accurate and useful information from high-dimensional sparse data. Biclustering has found many useful applications for visualization and exploratory analysis in various fields such as knowledge discovery, data mining, pattern classification, information retrieval, collaborative filtering, and especially in gene expression data analysis such as functional annotation, tissue classification, and motif identification.

It has been shown in previous studies that finding biclusters of data is inherently intractable and computationally complex. Generally, the challenges of biclustering include the high dimensionality of data, noisy data, different types of bicluster patterns, and the fact that biclusters can overlap. Although there are several studies in biclustering, after a review of the methods proposed in the literature, we found that these challenges are not addressed properly. Most of the proposed methods in literature can only detect a limited set of bicluster patterns under restrictive assumptions about the data. Moreover, in many methods biclusters are detected sequentially, i.e., the method replaces the detected bicluster with the background and detects the next bicluster, thus preventing the detection of overlapping biclusters.

Given the above statements, there is a need for innovative methods to extract valuable information from the data and to reach a deeper understanding of the outcomes. Therefore, in this study, we first proposed a method (PBD-SPEA) that

uses a new dynamic encoding scheme to detect multiple overlapped biclusters concurrently. However, the implementation is complex as there are several heuristic search procedures in different steps of the proposed method, and it is not able to detect all types of patterns in biclusters. Thus, a second method (LBDP) is proposed based on geometrical biclustering. In this method, we search for hyperplanes from the data using an evolutionary algorithm. Applying this idea, we are able to detect all types of bicluster patterns concurrently.

We defined several scenarios in both synthetic and real data to test the performance of the proposed methods. Although our work is initially targeted for biomedical data (gene expression data), we also tested the generality of the algorithms on other non-medical data, such as image data and social networking data. In all scenarios, our methods achieved reliable results compared to several state-of-the-arts.

List of Publications

GOLCHIN, M. & LIEW, A. W. C. 2017. Parallel Biclustering Detection Using Strength Pareto Front Evolutionary Algorithm. *Information Sciences*, 415-416**,** 283-297.

GOLCHIN, M. & LIEW, A. W. C. 2018. Biclustering by Multi-objective Evolutionary Algorithm for Multimodal and Big Data. *Multimodal Analytics for Next-Generation Big Data Technologies and Applications,* Springer, accepted.

GOLCHIN, M. & LIEW, A. W. C. 2018. Geometric Biclustering by Hyperplane Projection and Multi-objective Evolutionary Algorithm. *Pattern Recognition*, submitted.

GOLCHIN, M. & LIEW, A. W. C. Bicluster Detection by Hyperplane Projection and Evolutionary Optimization. Proceedings of 9th International Conference on Bioinformatics Models, Methods and Algorithms, 2018 Funchal, Madeira, Portugal.

GOLCHIN, M. & LIEW, A. W. C. Bicluster Detection using Strength Pareto Front Evolutionary Algorithm. Proceedings of the Australasian Computer Science Week Multiconference, 2016 Canberra, Australia. ACM, 1-6.

GOLCHIN, M., DAVARPANAH, S. H. & LIEW, A. W. C. Biclustering Analysis of Gene Expression Data using Multi-Objective Evolutionary Algorithms. Proceeding of the 2015 International Conference on Machine Learning and Cybernetics 2015 Guangzhou, China. IEEE, 505-510.

List of Acronyms

AMPP	Additive and multiplicative pattern plot
BBAC	Bregman block average co-clustering
BicAT	Biclustering analysis toolbox
BiMax	Binary inclusion-maximal
BP	GO biological process
CC	Cheng and Church method
cDNA	Complementary deoxyribonucleic acid
DMOIOB	Dynamic multi-objective immune optimization biclustering
DMOPSOB	Dynamic multi-objective particle swarm optimization biclustering
DNA	Deoxyribonucleic acid
EA	Evolutionary algorithm
ECOPSM	Evolutionary computation by the order preserving submatrix
FABIA	Factor analysis for bicluster acquisition
FG	Factor graph
FDR	False discovery rate
GA	Genetic algorithm
GCC	GO cellular component
GO	Gene ontology
HMOBI	Hybridization of multi-objective evolutionary metaheuristic
HT	Hough transform
Hyp	Hypergeometric p-value

Hyp*	Corrected hypergeometric p-value
ISA	Iterative Signature Algorithm
KEGG	Kyoto encyclopedia of genes and genomes
LAS	Large average submatrices
LBDP	Linear bicluster detection by projection
MF	GO molecular function
MODPSFLB	Multi-objective dynamic population shuffled frog-leaping biclustering
MOIB	Multi-objective immune biclustering
MOM-aiNet	Multi-objective multi-population artificial immune network
MOPSOB	Multi-objective practical swarm optimization biclustering
MPM	Metabolic pathway maps
mRNA	Messenger ribonucleic acid
MSR	Mean square residue
NG	Number of annotated genes in the input list
NGR	Number of annotated genes in the reference list
NSGA	Non-dominated sorting genetic algorithm
NSGA2B	Non-dominated sorting genetic algorithm 2 biclustering
OPSM	Order preserving submatrix
PBD-SPEA	Parallel bicluster detection by strength Pareto front evolutionary algorithm
PCC	Pearson correlation coefficient
PPI	Protein-protein interaction networks
RMSE	Root mean square error
RNA	Ribonucleic acid

SSBiEM	Spike and slab biclustering expectation-maximization
SVD	Singular value decomposition
TNG	Total number of genes in the input list
TNGR	Total number of genes in the reference list
TWCC	Two-way subspace weighting partitioned co-clustering method

List of Symbols

b_{ij}	The element of the i^{th} row and j^{th} column of the bicluster
Bic	A bicluster in the data matrix
C	Column indices of a bicluster
Data	The data matrix
e_{ij}	The element of the i^{th} row and j^{th} column of the data matrix
F	Column indices of a data matrix
i	The i^{th} row in the bicluster or data matrix
j	The j^{th} column in the bicluster or data matrix
m	Number of rows in the bicluster
M	Number of rows in the data matrix
n	Number of columns in the bicluster
N	Number of columns in the data matrix
R	Row indices of a bicluster
S	Row indices of a data matrix

Table of Contents

List of Figures

List of Tables

1

Introduction

This thesis presents a multi-objective evolutionary algorithm to solve the biclustering problem in data mining and data analytic communities especially in the field of gene expression data analysis. The challenge of finding biclusters in data matrices is an NP-complete problem (Cheng and Church, 2000) because the search space for finding biclusters increases exponentially when the volume of data increases.

In this chapter, we introduce common data mining method and their shortcomings to uncover interesting patterns in data. Then, we illustrate the methodologies, the aims, and the achievements of this research in terms of the scientific contributions to solve the biclustering problem. Finally, we outline the organisation of the thesis.

1.1 From Clustering to Biclustering

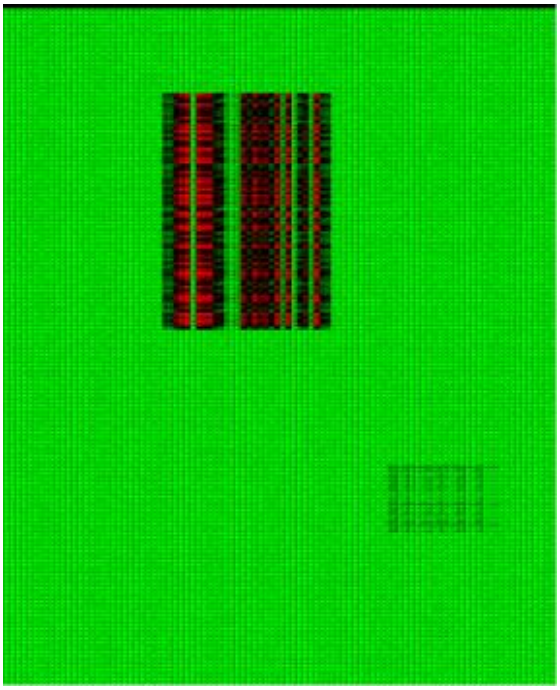
Pattern recognition focuses on finding patterns and regularities in a given data. Extracting interesting patterns in data is an optimization problem (Han et al., 2011). Mining query optimization, performance and pattern evaluation are major issues in data mining (Han et al., 2011) especially when the data tends to be noisy (Fan et al., 2014). Data mining uses a variety of methodologies for analysing and modelling data. It aims at revealing similarity in samples of data while discarding irrelevant samples. A common approach for pattern recognition

and data analysis is cluster analysis (Bailey, 1994). Clustering is the task of partitioning samples or features into set of clusters in such a way that elements inside a cluster are more similar while elements from different clusters have low similarity to each other. Traditional clustering approaches such as partitional clustering (Celebi, 2014), hierarchical clustering (Szeto et al., 2003), and density based clustering (Kriegel et al., 2011) have several limitations. For examples, the entire set of features are considered in a cluster. In addition, most clustering methods assign a given sample or feature to only one cluster. However, in many real world situations, a set of samples exhibit similar patterns only under a subset of features (Zhao et al., 2012). Moreover, some samples or features may participate in several clusters and some samples or features may not be part of any cluster.

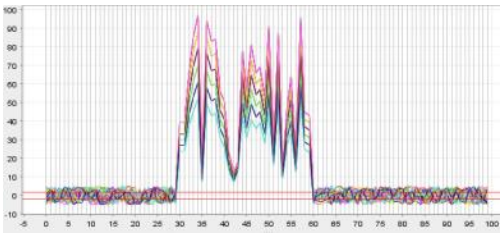
Given the above, Hartigan introduced biclustering (also called co-clustering) in 1972 and called it direct clustering (Hartigan, 1972) and the term is used in the book of Mirkin (Mirkin, 1996). Biclustering is a data mining method that refers to the clustering of both the sample and feature dimensions in a dataset simultaneously, and it finds subgroups to discover the interrelationship and local patterns from data. In biclustering, a sample can take part in several biclusters under different subsets of features. A bicluster determines a subgroup of elements in the dataset with rearranged samples and features that follow a coherent pattern such that subset of samples shows considerable homogeneity within a subset of features.

To illustrate the differences between clustering and biclustering, Figure 1.1 provides a visualization. Figure 1.1 (a) shows a dataset with 200 samples and 100 features, which contains two biclusters. Figure 1.1 (b) and Figure 1.1 (c) show the element values of the detected clusters by a clustering method and Figure 1.1 (d) and Figure 1.1 (e) show the element values of the detected

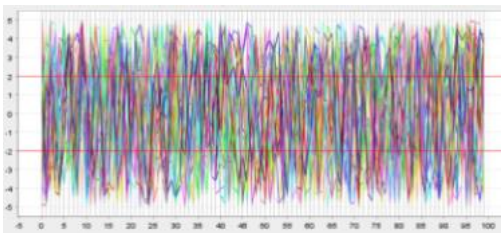
biclusters by a biclustering method. In Figure 1.1 (b), (c), (d) and (e), the x-axis is the column indices that the methods detected in the clusters or biclusters and the y-axis is the element values. Each line in the plots refers to a sample's feature vector under a subset of features. Figure 1.1 (b) and (d), and Figure 1.1 (c) and (e) refer to a same set of samples with different subset of features (Figure 1.1 (b) and (c) include all features in the data while Figure 1.1 (d) and (e) include only a subset of features). As can be seen, Figure 1.1 (d) and (e) (i.e. biclusters) display the same pattern i.e. all samples in the detected bicluster exhibit high coherence and can provide more accurate and valuable information about the data.



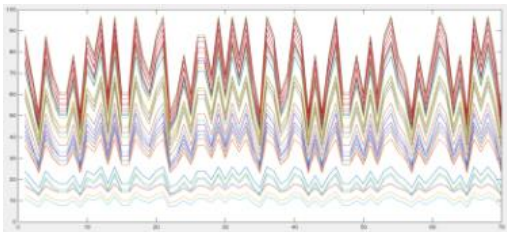
(a)



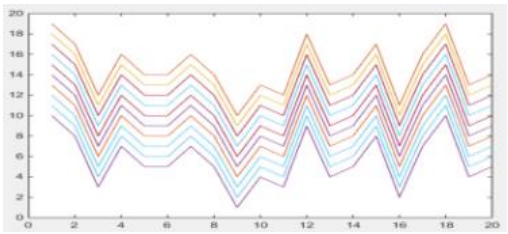
(b)



(c)



(d)



(e)

Figure 1.1. Conceptual difference of clustering methods (b) and (c) versus biclustering methods (d) and (e), original data matrix is shown in (a) with two embedded biclusters.

1.2 Research Aims

In this section, we present the problem statement and motivating factors that encouraged us to conduct this research; the aims and the goals we set for this research; the methods of dealing with the research aims to achieve our goals; and our achievements towards the goals and the contributions to the area of research.

1.2.1 Problem Statement

In the field of biclustering, the dimension of data, i.e. the number of columns, heavily effects the computational cost of a method. In addition, there exist only a couple of methods (Zhao et al., 2008, Gan et al., 2008) which can handle different type of patterns in noisy data. Furthermore, allowing overlap in biclusters can provide a better representation of information in many real world applications, for example, in the study of the biological relationship between genes and functions in gene expression data. These three factors motivated us to define the main problem of this research. The biclustering problem can be expressed as solving a hyperplane detection problem using a multi-objective evolutionary algorithm.

1.2.2 Research Methodology

Previous studies (Divina and Aguilar, 2006, Golchin and Liew, 2017, Liew, 2016, Mitra and Banka, 2006, Seridi et al., 2011) have shown that one way to enhance the biclustering accuracy is to optimize a fitness function via some iterative process. One promising approach is through the use of evolutionary algorithm (EA). However, there has not been extensive study in this area. In this study, we tackle issues to enhance the biclustering accuracy by

- Proposing a novel dynamic encoding scheme with new crossover and mutation operators, which can optimize several biclusters concurrently. Using a newly introduced merit function, our algorithms can handle noisy data and detect overlapping biclusters in a dataset.
- Proposing a new geometric based multi-objective EA biclustering method, which is able to search for hyperplanes in a high dimensional feature space.

- Finding a set of final biclusters from the optimal set of individuals that constitute the Pareto front in a multi-objective optimization process.
- Investigating the generality of the proposed methods to other non-medical datasets such as image data and social media data

1.2.3 Contributions

The focus of this research is to propose methods that are able to discover different bicluster patterns in high dimensional noisy datasets especially gene expression data. Our contributions are as follows.

- We proposed two algorithms that are based on multi-objective evolutionary optimization and the geometric biclustering framework.
- We developed an effective method to detect hyperplane using SVD during EA optimization.
- Our algorithms are able to detect biological meaningful biclusters in noisy and high dimensional data.
- Our algorithms can extract overlapped biclusters better than existing algorithms.
- Our algorithms can detect different types of bicluster patterns via the geometrical biclustering framework.

1.3 Organization of This Thesis

The rest of this thesis is organised as follows. In the second chapter, a critical survey of the literature and state of art methods has been performed to the biclustering problem. We also discuss their merits and their shortcomings.

In Chapter 2, we introduce the biomedical data used in this research, discuss the biclustering problem, the multi-objective evolutionary optimization, and the relevant background knowledge that lead to the proposed methods.

In Chapter 3, we present the first proposed EA based method, called PBD-SPEA (parallel bicluster detection using strength Pareto front algorithm), to search for multiple biclusters concurrently using a novel encoding scheme, crossover operation, and mutation operation.

In Chapter 4, we present our second proposed EA method based on the geometrical biclustering framework, called LBDP (linear bicluster detection by projection). In this method, the hyperplane can be detected effectively using SVD. We also combined niching method to LBDP to search different areas of the multi-objective space simultaneously for multiple bicluster detection.

Finally, in Chapter 5, we sum up this thesis by presenting the conclusions of our research and outline potential future research directions.

2

Background and Literature Survey

During the past couple of decades, a wide range of approaches has been proposed to solve the biclustering problem (Madeira and Oliveira, 2004, Zhao et al., 2012, Pontes et al., 2015, Padilha and Campello, 2017). Among these approaches, evolutionary methods and geometric methods have attained promising results for the biclustering problem.

In this chapter, we will first discuss biomedical data analysis, specifically that of gene expression data. Second, we will discuss bicluster analysis including the types of bicluster patterns, and validation methods. Then we will review different approaches to address the biclustering problem and highlight their advantages and disadvantages. Finally, the theory of evolutionary algorithm in single objective optimization and multi-objective optimization will be explained.

2.1 Biomedical Data Analysis

Microarrays, also called microscope DNA chips or gene chips, are a collection of DNA (Deoxyribonucleic acid) sequence, known as probes or gene in defined positions attached to a solid surface. A microarray monitors the expression level of thousands of gene samples simultaneously under various biological conditions or at the same time phases in experimental molecular biology (Selvaraj and Natarajan, 2011). The probes, which detect gene expression, are also known as the set of messenger RNA (mRNA).

Microarray technologies provide insight about the biological processes at the genomic level enabling the quantitative analysis of gene functions. In microarray analysis, mRNAs are collected from an experimental sample (an individual with a disease like cancer or a treatment) and a reference sample (a healthy individual). Then these two samples are converted into complementary DNA (cDNA) and samples are labeled with different fluorescent colors (usually green and red). The two samples are then mixed and bind to a microarray slide (hybridization). Following hybridization, the expression of each printed gene on the slide is measured. If the expression of a gene is higher in the experimental sample, the corresponding spot in the microarray appears red. This spot appears green otherwise. If there is an equal expression, then the spot appears yellow (Babu, 2004). Figure 2.1 illustrates an overview of the above procedure.

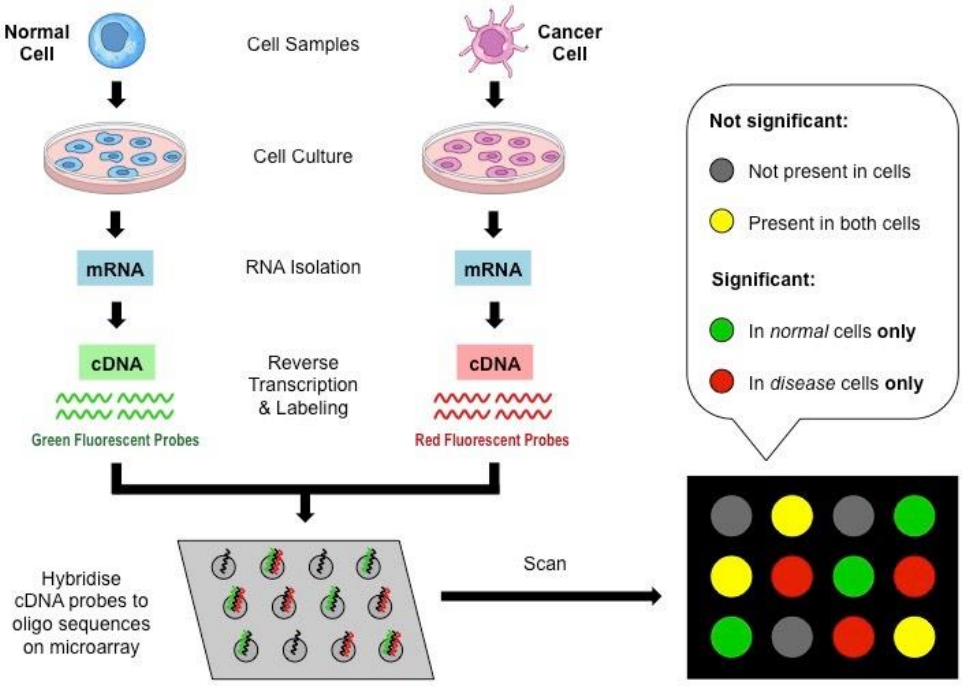


Figure 2.1. Microarray procedure. Figure is from http://ib.bioninja.com.au/standard-level/topic-3-genetics/35-genetic-modification-and/cdna-and-microarrays.html

A gene expression is created by the process of transcribing the genetic information in DNA into mRNA. In the same way, the logarithmic ratios between the intensities of the dyes after some post processing give rise to the gene expression matrix. Ultimately, the data through microarrays that creates the gene expression profiles are used to study the change in the expression of genes in response to a particular condition, diseases, treatment, or development stages. There exists different types of microarrays, namely, cDNA, oligonucleotide arrays, photolithography, ink-jet printing, electrochemical synthesis, single-channel arrays, and multiple-channel arrays (Adomas et al., 2008). Gene expression analysis can be a major tool for identifying biomarkers, classifying

diseases, monitoring the response to a therapy, diagnosis and prognosis of diseases, and the study of new medicine (Tarca et al., 2006).

In this research, we are interested in the biclustering of gene expression data where the aim is to infer the biological roles and processes of an unknown gene and unknown genetic pathway by association with known annotated genes in a bicluster (Liew, 2016). Biclustering is also used in the detection of marker genes that are associated with certain tissue, treatment or disease when their expression values are changed (MacDonald et al., 2001, Cha et al., 2014). We used six different gene expression data in our research and an overview of each gene expression dataset is given below. Table 2.1 summarises the main features of the biomedical databases.

Table 2.1. Biomedical databases features

Data matrix	**# Genes**	**# Conditions**	**Organism**
Yeast *Saccharomyces Cerevisiae*	2884	17	*Saccharomyces Cerevisiae*
Unfolded protein response	6091	13	*Saccharomyces Cerevisiae*
Multiple human organs	18927	158	Homo sapiens
Human B-cell Lymphoma	4026	96	Homo sapiens
Medulloblastoma tumor	2059	23	Homo sapiens
Breast cancer	1259	19	Homo sapiens

2.1.1 Yeast *Saccharomyces Cerevisiae* Database

The yeast *Saccharomyces cerevisiae* gene expression data (Cho et al., 1998) is one of the most intensively studied dataset in molecular and cell biology. There are many important proteins in common between yeast and human biology. In 60-70% of patients with Crohn's disease and 10-15% of patients with ulcerative

colitis, the antibodies against *s. cerevisiae* are found (Walker et al., 2004). *s. cerevisiae* is a significant tool to study DNA damage and repair mechanisms (Nickoloff and Haber, 2001). For this reason, we used this database to study the effectiveness of the proposed methods.

2.1.2 Unfolded Protein Response Database

Unfolded protein response (GDS750) in *saccharomyces cerevisiae* is the *HAC1* transcription and unfolded protein response that strains the expression of *HAC1* under various promoters (Leber et al., 2004). There are unrecognized pathways that operates in yeast to regulate the transcription of *HAC1*. Biclustering aims to distinguish these pathways.

2.1.3 Multiple Human Organs Database

Multiple human organs (Son et al., 2005) originally includes 42421 genes and 158 conditions with 18927 unique genes and 19 different organs. This database helps to understand the cause of diseased organs. Studying this database can help to develop the new targeted genes that can be used in the treatment of patients.

2.1.4 Human B-cell Lymphoma Database

Diffuse large B-cell lymphoma database (Alizadeh et al., 2000) is from an aggressive malignancy of mature B-lymphocytes. Human B-cell lymphoma has attracted lots of attention in the past two decades because over 25000 new cases are obtained annually (Alizadeh et al., 2000). However, most attempts to generate diagnostic solutions failed. Consequently, there is a need to study human B-cell lymphoma in a genomic scale gene expression profiling.

2.1.5 Medulloblastoma Tumor Database

Medulloblastoma tumor database (GDS232) is a study on Medulloblastoma metastasis that identifies genes causing medulloblastoma tumors to metastasize (MacDonald et al., 2001). Twenty-three primaries medulloblastoma are analyzed and designated as either metastatic or non-metastatic and 85 genes have already been identified where their expression differs highly from one class to another. Little is known about the genetic regulation in metastatic medulloblastoma and this leads to poor outcomes. Therefore, new methods are needed to study this database.

2.1.6 Breast Cancer Database

Breast cancer database (GDS4085) is from estrogen receptor-positive and -negative breast cancer tumors that analyses the primary breast carcinoma tumors from estrogen receptor positive or negative (ER+/-) patients. ER+ tumors tend to metastasize to the bone while ER- tumors tend to induce visceral metastasis. Nine conditions represent positive tumor and ten conditions represents negative tumor are in this database (Julien et al., 2011). Studying this database provide insight into the molecular basis of different metastatic phenotypes in breast cancer.

2.2 Bicluster Analysis

Let $Data = (e_{ij})_{M\times N}$ be a data matrix representing the values of M rows (samples) denoted by $S = \{s_1, s_2, \ldots, s_M\}$, and N columns (N-dimensional feature vector) denoted by $F = \{f_1, f_2, \ldots, f_N\}$. The matrix element e_{ij} is the i^{th} row value under j^{th} column. To illustrate, Figure 2.2 shows $Data = (S,F) \in \mathbb{R}^{M\times N}$.

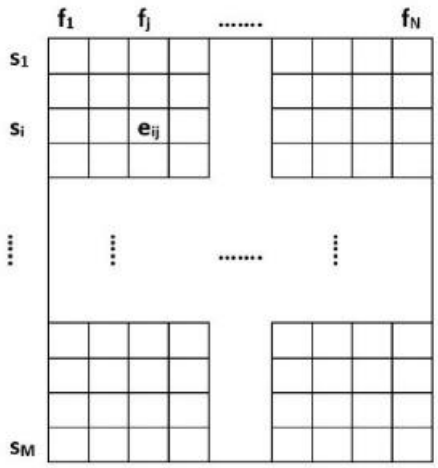


Figure 2.2. Data representation

Generally, a bicluster is a subset of rows that exhibit similar behaviour across a subset of columns. Therefore, a bicluster $Bic = (R,C) \subseteq Data$ exhibits some coherent pattern, where $R = \{s_1, \ldots, s_m\} \subseteq S$ and $C = \{f_1, \ldots, f_n\} \subseteq F$.

Biclustering aims to discover a set of biclusters such that each bicluster satisfies certain coherent pattern. There are different types of coherent pattern in a bicluster (Cheng et al., 2008). The most common patterns are highlighted here.

Constant value pattern has constant value in the entire pattern as shown in Figure 2.3 (b), which can be represented as $c_k = \ldots = c_j = a_1$. Constant row pattern has constant value for each row as in Figure 2.3 (c), which satisfies $c_k = \ldots = c_j$. Constant column pattern has constant value for each column as in Figure 2.3 (d) that corresponds to $c_k = b_1$ and $c_j = b_2$. Linear pattern has values that satisfies $c_k = a_2 \times c_j + b_3$ as in Figure 2.3 (e). Additive pattern has additive values that satisfies $c_k = c_j + b_4$ as in Figure 2.3 (f). Multiplicative pattern has multiplicative values that satisfies $c_k = a_3 \times c_j$ as in Figure 2.3 (g). where a_1, a_2, a_3, b_1, b_2, b_3, b_4 are constant values.

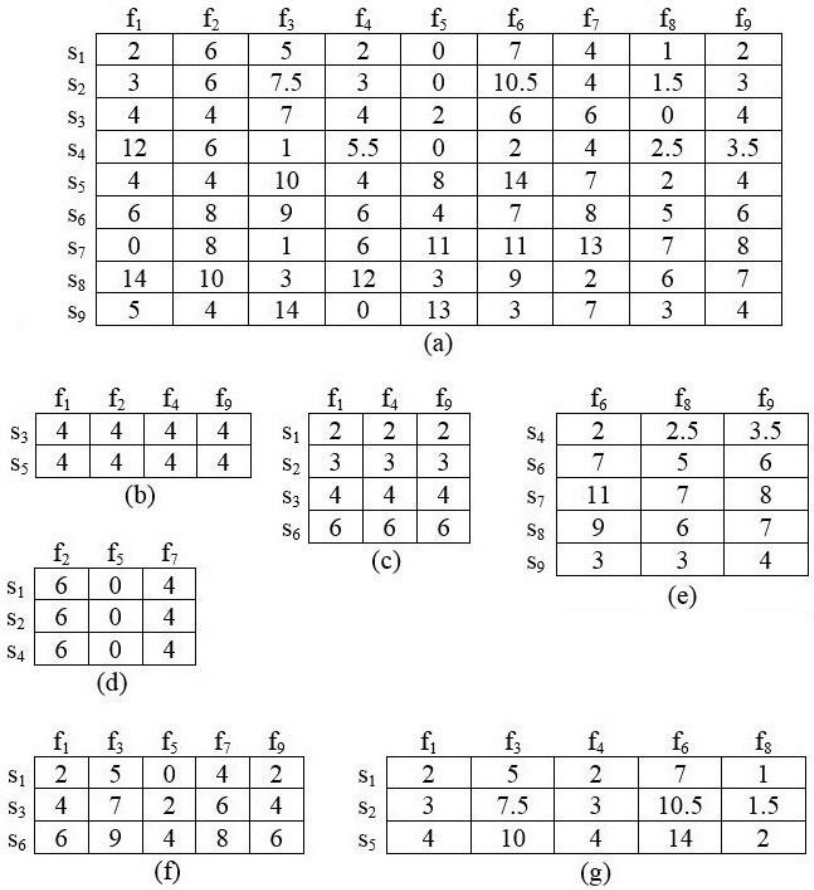


Figure 2.3. (a) A 9 × 9 data matrix with hidden biclusters; (b) a constant value pattern bicluster; (c) a constant row pattern bicluster; (d) a constant column pattern bicluster; (e) a linear pattern bicluster; (f) an additive pattern bicluster; (g) a multiplicative pattern bicluster

From the geometrical viewpoint, we can consider each pattern as a special case of linear pattern $c_k = A \times c_j + B$, i.e. in constant value pattern, $A = 0$ and $B = a$; in constant row pattern, $A = 1$ and $B = 0$; in constant column pattern, $A = 0$; in additive pattern, $A = 1$; in multiplicative pattern, $B = 0$, where A and B are constant values.

Given the above, if the columns are considered as coordinate axes in a high-dimensional space, a constant value pattern and constant column pattern form a point, constant row pattern forms a line, whereas additive pattern, multiplicative pattern, and linear pattern form a hyperplane in the space.

The geometric interpretation of linear bicluster patterns unifies all bicluster patterns into a single linear class and allows a unified treatment in detecting them simultaneously (Du et al., 2014). This is in contrast to most existing biclustering methods where the cost function implicitly imposes a constraint on the type of bicluster patterns that could be discovered.

In principle, any method for detecting linear patterns i.e. hyperplane detection methods, can be employed to detect biclusters in data matrices. This characteristic of linear patterns is the basis of the geometric biclustering frameworks.

The other challenging issue in biclustering is how to validate a detected bicluster. Validating the quality of biclusters is accomplished by using measures derived from the data itself or by using domain knowledge (Zhao et al., 2012).

We can compute some quality measures of a biclustering result to evaluate the accuracy of the detected biclusters when the true biclusters are known in the data such as in synthetic data. In this case, Jaccard index, also called matching score, counts the number of rows and columns that are common between the detected bicluster and the ground truth as in Equation (2.1). The value of Jaccard index varies from zero to one where zero indicates no similarity and one indicates 100% similarity. The higher the value of Jaccard index, the better the accuracy of the detected bicluster and ultimately the better the performance of the method. The Jaccard index J is defined by

$$J(Bic, G) = \frac{1}{|Bic|} \sum_{R,C} \max_{R_G, C_G} \frac{|R \cap R_G| + |C \cap C_G|}{|R \cup R_G| + |C \cup C_G|} \quad \textbf{(2.1)}$$

where G is the bicluster ground truth, R_G and C_G are rows and columns of the bicluster ground truth. |•| represents the number of elements. J calculates the ratio of similar rows and columns over the total number of rows and columns in the detected bicluster and the ground truth.

Gene expression data measures the expression level of thousands of genes (rows) under various biological conditions (columns). For gene expression analysis, the domain knowledge about the gene expression data helps to assess the biological relationship of the detected genes and enables the quantitative analysis of genes. A common way to check the enrichment in the biclusters is by using p-value statistics. The p-value measures the probability of finding the number of genes with a specific GO term in a bicluster by chance and shows the statistical significance of the results. Smaller p-value indicates strong evidence that the selected genes are highly correlated. Equation (2.2) calculates the p-value, where A is the number of annotated genes in the background set, and k is the number of annotated genes within the detected genes.

$$P = 1 - \sum_{i=0}^{k-1} \frac{\binom{A}{i}\binom{M-A}{R-i}}{\binom{M}{i}} \quad \textbf{(2.2)}$$

In order to study the biological relevance of extracted biclusters we can use Gene Ontology (GO), metabolic pathway maps (MPM), protein-protein interaction networks (PPI), and Kyoto Encyclopaedia of Genes and Genomes (KEGG) pathway. GO is organized into hierarchical annotations (Ashburner et al., 2000) while the KEGG database organizes the genes products into pathway reaction maps (Kanehisa and Goto, 2000).

In addition, KEGG pathway represents the known biological knowledge about the molecular interaction and reaction networks for genetic information processing, cellular process, human diseases, and drug development (Kanehisa and Goto, 2000).

Three ontologies are available in GO which provides a dynamic, controlled vocabulary for various genomic databases, namely, biological process that represents the effects of genes in a biological objective, molecular function that shows biochemical activity, and cellular component, which refers to a location in the cell where a gene product is activated (Ashburner et al., 2000).

Tools such as GeneCodis (Tabas Madrid et al., 2012, Nogales Cadenas et al., 2009, Carmona Saez et al., 2007), GO-TermFinder (Boyle et al., 2004), and ClueGO (Bindea et al., 2009) study the biological relationship of extracted biclusters by analyzing modular and singular enrichment. GeneCodis is available online at http://genecodis.cnb.csic.es/ and calculates the p-value for different annotations. This toolbox also calculates modular enrichment analysis that provides a complete functional analysis by detecting additional significant terms (Mahé et al., 2014).

Depending on the bicluster pattern and the search strategy, a number of biclustering methods have been proposed (Zhao et al., 2012). These include distance based biclustering, spectral based biclustering, probabilistic based biclustering, geometric based biclustering, and evolutionary based biclustering. We will next review the general concepts and several well-known methods in each category. An overview of these methods is summarised in Table 2.2.

Table 2.2. A summary of different biclustering techniques

Category	Method	Reference
Distance based biclustering	Iterative greedy search - CC	(Cheng and Church, 2000)
Spectral based biclustering	Eigenvector decomposition of normalized data	(Kluger et al., 2003)
	Factor based coherence optimization by max-sum	(Denitto et al., 2017b)
	Two-way subspace partitioning - TWCC	(Chen et al., 2018)
Probabilistic based biclustering	Data decomposition - FABIA	(Hochreiter et al., 2010)
	Data decomposition - SSBiEM	(Denitto et al., 2017a)
	Semantic enrichment	(Kléma et al., 2017)
	Convex optimization - COBRA	(Eric et al., 2017)
Geometric based biclustering	Fast Hough transform	(Gan et al., 2005)
	Column pair Hough transform	(Zhao et al., 2008)
	Column pair Hough transform	(Wang and Yan, 2010)
	Graph spectrum column pair Hough transform	(Wang et al., 2012)
	Graph spectrum column pair Hough transform	(Wang and Yan, 2013)
	Graph spectrum column pair Hough transform	(Liu et al., 2014)
	Genetic algorithm hyperplane detection	(To and Liew, 2014)
Evolutionary based biclustering	Multi-objective multi-population artificial immune network – MOM-aiNet	(Coelho et al., 2008)
	Dynamic multi-objective immune network optimization - DMOIOB	(Liu et al., 2011)
	Genetic algorithm optimization	(Divina and Aguilar Ruiz, 2006)
	Optimal reordering of samples and features - OPSM	(Roh and Park, 2008)
	Multi-objective non-dominated sorting genetic algorithm	(Mitra and Banka, 2006)

Category	Method	Reference
	Multi-objective genetic optimization	(Maulik et al., 2009)
	Multi-objective evolutionary optimization	(Seridi et al., 2011)

2.2.1 Distance based Biclustering

This method is one of the earliest methods in literature and it has applications in many fields. Distance based biclustering measures the quality of the biclusters using a distance metric and an iterative search by adding and/or removing rows or columns such that the residual sum of squares cost is minimized. Among different introduced residual measures, mean square residue (MSR) introduced by (Cheng and Church, 2000) is the most famous one. Although these methods may fail to detect biclusters and mostly cover constant and additive structures only, they have the potential to be very fast (Madeira and Oliveira, 2004).

Cheng and Church (CC) (Cheng and Church, 2000) were the first to apply biclustering method to analyse the biological gene expression data. They used a heuristic greedy search method to detect δ-biclusters one at a time. For a predefined number of biclusters, they iteratively removed and inserted rows and columns to a detected bicluster while the MSR error remained below δ. Some recent evolutionary based methods have used this method as a local search strategy (Golchin and Liew, 2017, Golchin and Liew, 2016, Golchin et al., 2015, Seridi et al., 2012, Seridi et al., 2011).

MSR measures the degree of coherence of a bicluster. Equation (2.3) calculates MSR value where e_{iC}, e_{Rj} and e_{RC} are the mean of the i^{th} row, the mean of the j^{th} column, and the mean of the bicluster $Bic = (R,C)$, respectively, and are

calculated by Equations (2.4)-(2.6). If MSR(R,C) $\leq \delta$ then a bicluster is called a δ-bicluster. The smaller δ is, the better the coherence of the rows and columns.

$$MSR(R,C) = (\sum_{i\in R, j\in C} \left(e_{ij} - e_{iC} - e_{Rj} + e_{RC}\right)^2)/|R||C| \quad \textbf{(2.3)}$$

$$e_{iC} = (\sum_{j\in C} e_{ij})/|C| \quad \textbf{(2.4)}$$

$$e_{Rj} = (\sum_{i\in R} e_{ij})/|R| \quad \textbf{(2.5)}$$

$$e_{RC} = (\sum_{i\in R, j\in C} e_{ij})/|R||C| \quad \textbf{(2.6)}$$

2.2.2 Spectral based Biclustering

In contrast to the methods that apply greedy heuristic search for biclustering problem, spectral based biclustering methods use spectral decomposition techniques. Spectral decomposition uncovers natural structures in the data that are related to hidden biclusters. The main drawback of these methods is the long computation time.

Kluger et al. (Kluger et al., 2003) found distinctive checkerboard structure patterns in the data matrices by eigenvectors corresponding to characteristic patterns across rows or columns. They assumed that the data matrix has a block diagonal structure in a way that elements outside the blocks equal to zero. In this case, each block corresponded to a bicluster. They used singular value decomposition to identify columns with the same conserved linear structure across the rows in a normalized data matrix. The normalization was done with a standard linear algebra manipulation and the whole data matrix was used in a global manner to eliminate the effects of different experimental columns. The

authors applied their method to cancer genomic data matrices and they achieved reasonable results.

In (Denitto et al., 2017b), the biclustering problem was solved by performing sequential search for one bicluster at a time. In order to make their proposed method scalable in comparison to the previous factor based (FG) methods, the authors used a compact binary FG based on a given coherence optimization criteria by the max-sum method. In their method, each bicluster was represented as a binary matrix $Bic \in \{0,1\}^{n \times m}$ where $b_{ij} = 1$ if the entry (i,j) belongs to the bicluster and zero otherwise. They found the largest biclusters by measuring the incoherence of a bicluster as a constant-type incoherence by $I(e_{ij},e_{tk}) = (e_{ij} - e_{tk})^2$ or an additive-wise incoherence by $I(e_{ij},e_{tk}) = (e_{ij} - e_{tj} + e_{tk} - e_{ik})^2$. They also rewarded the solutions containing entries with high values by assuming that all the entries in a data matrix contained only positive values. Finally, their method tried to maximize the following function, where $A_{ij}(b_{ij}) = e_{ij}$ if $b_{ij} = 1$ and 0 otherwise, $O_{ij,tk} = -I(e_{ij},b_{tk})b_{ijcik}$, and $B_{jk}(b_{:j},b_{:k}) = 0$ if $(\sum_i b_{ij})(\sum_i b_{ik})(\sum_i |b_{ij}-b_{ik}|) = 0$ and $-\infty$ otherwise.

$$F(C) = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}(b_{ij}) + \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{t=1}^{m}\sum_{k=1}^{n} O_{ij,tk}(b_{ij}, b_{tk}) + \sum_{j=1}^{n}\sum_{k=1}^{n} B_{jk}(b_{:j}, b_{:k})$$

In their method, they use three types of factors namely, A_{ij}, $O_{ij,tk}$, and B_{jk} to derive six types of messages. The authors tested the accuracy of their method by both synthetic and real dataset and achieved promising results compared to state-of-the-art methods. Although compared to other FG methods, their method has

achieved acceptable results, there are tight assumptions that limit their method as summarised in the following.

Using a sequential bicluster detection approach would miss the overlapped area in the detected biclusters. For this reason, in their experiments they used only one bicluster in their synthetic dataset. In addition, to calculate the incoherence of the biclusters they proposed using two models namely additive coherent and constant wise coherent. This assumption makes their method unable to detect the more general type of biclusters with linear structure. Further, they assumed the values of the data matrix are all positive, which is unrealistic for gene expression data. In their paper, the authors mentioned "*the entry level counts*" and they rewarded biclusters that contain elements with larger value in comparison to the background elements. However, what happen if the element values of a bicluster are close to the background noise?

Most importantly, they demonstrated their method for a 50 × 50 data matrix but in order to scale up, they used a divide and conquer strategy to randomly partition the data matrix into non-overlapping submatrices. In addition, in order to overcome the curse of dimensionality, they ignore a reasonable portion of the data matrix. The authors adopt an affinity propagation method to merge the obtained biclusters in real life datasets. However, this process is time consuming and there is no guarantee to obtain the optimal biclusters.

A two-way subspace weighting partitioned co-clustering method TWCC was proposed in (Chen et al., 2018). Two types of binary weight matrices separate the data matrix into subspaces i.e. columns in row clusters and rows in column clusters. These weights that deal with noisy data, are combined by Hadamard product of the two binary matrices to simultaneously determine the contribution of each row or column in co-clusters. The Bregman block average co-clustering

(BBAC) uses a squared Euclidean distance function based on these two types of weights to determine the co-clusters of data, and an iterative algorithm, is used to optimize the distance function.

Equation (2.7) is the objective function that are minimized, where K and L are the number of row clusters and column clusters, respectively. U and V are the two binary matrices, in which $u_{ig} = 1$ indicates that the i^{th} row is assigned to the g^{th} row cluster and $v_{jh} = 1$ indicates that the j^{th} column is assigned to the h^{th} column cluster. Z is the centres of the $K \times L$ co-clusters. R and C are the weight matrices for rows and columns, respectively. λ And η are two positive parameters. $d(e_{ij}, z_{gh})$ is the Bregman divergence and is defined as $(e_{ij} - z_{gh})^2$.

The authors had run experiment on both real and synthetic datasets and claimed that their proposed method is robust and effective for large high-dimensional data.

$$
\begin{aligned}
&P(U, V, Z, R, C) \\
&= \frac{1}{MN} \sum_{g=1}^{K} \sum_{h=1}^{L} \sum_{i=1}^{M} \sum_{j=1}^{N} u_{ig} v_{jh} r_{hi} c_{gj} d(e_{ij}, z_{gh}) \\
&+ \frac{\lambda}{M} \sum_{h=1}^{L} \sum_{i=1}^{M} r_{hi} \log r_{hi} + \frac{\eta}{N} \sum_{g=1}^{K} \sum_{j=1}^{N} c_{gj} \log c_{gj}
\end{aligned}
\qquad \textbf{(2.7)}
$$

The performance of TWCC is sensitive to the threshold values of the penalty terms in the objective function. Furthermore, they have no strategy to handle overlapped biclusters which is an important issue in real life data.

2.2.3 Probabilistic based Biclustering

In this category, a probabilistic model of biclusters and statistical parameter estimation techniques is used to search for the biclusters. The idea is that in order to detect the existence of a bicluster, non-deterministic methods are designed

that detect a bicluster if the probability of a failure is less than a threshold. Not only these methods are technically complex but also these methods may not detect the desired bicluster (Alon and Spencer, 2004).

Hochreiter et al., (Hochreiter et al., 2010) proposed a generative approach based on a multiplicative model called FABIA (factor analysis for bicluster acquisition). Their method was able to detect linear dependencies between rows and columns and it had been designed for heavy tailed distributions as found in many real world data. The authors decomposed a data matrix in levels and assumed the data was the sum of several biclusters with additive noise, i.e. $Data = \sum_{i=1}^{p} \lambda_i z_i^T + \Upsilon = \Lambda Z + \Upsilon$, where $\Lambda_i \in \mathbb{R}^M$ and $z_i \in \mathbb{R}^N$ are the sparse vectors corresponding to the i^{th} bicluster, $\Upsilon \in \mathbb{R}^{n \times m}$ is the additive noise.

Despite of using variational expectation maximization to estimate the model parameters, the corresponding likelihood is not tractable. Further, a post-processing step is needed to provide the biclusters membership information.

Following the work of (Hochreiter et al., 2010) and in order to overcome the drawbacks of their models, in (Denitto et al., 2017a) the authors addressed the problem of biclustering by approaching it as a probabilistic sparse low-rank matrix factorization problem. They designed a probabilistic model to describe the factorization of a given data matrix into the product of two matrices that provided the rows and columns of the biclusters, i.e. $Data = \sum_{i=1}^{k} v_i z_i^T + Y = VZ + Y$, where Y is random noise. Each bicluster had its own parameter values and is the outer product of two sparse vectors, with the data matrix being modelled as the sum of k (number of biclusters which is known beforehand) outer products (VZ). The authors modelled the data as a Gaussian distribution having VZ as mean and the estimated noise as the variance, and used a spike and slab sparsity inducing prior (SSBi).

Spike and slab is a probabilistic model for variable selection. In order to estimate the model parameters and bicluster membership information, expectation-maximization (EM) method was proposed (SSBiEM) to solve the problem of low-rank factorization problem by the augmented Lagrangian method.

The main ingredient of their method is that they assumed the vectors corresponded to the biclusters are sparse. This is a drawback since sometimes a bicluster can be a big part of the original data matrix. The EM method proposed in their paper involves an approximate M-step that minimizes a non-convex function and there is no guarantee that the method will converge. Furthermore, the performance may depend on the initialization step. The authors also claimed that their methods allows for overlapped biclusters. We tested their method (available online at https://github.com/emme-di/ssbiem/) with the data matrix designed in Experiment 4 of Section 4.3.2. The Jaccard index is 0.94 when there is no overlapped in the data matrix, but the performance drops significantly when the overlapped degree increases.

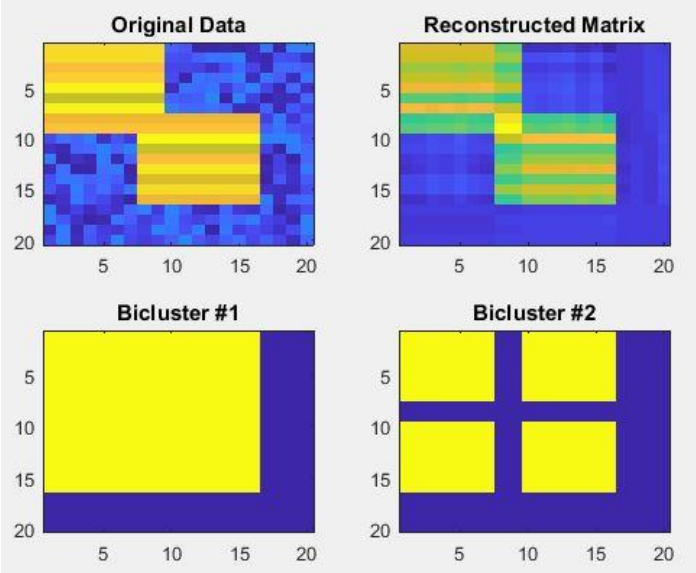


Figure 2.4. The accuracy of detected biclusters in (Denitto et al., 2017a) when the level of overlapped is 2

As can be seen from Figure 2.4, SSBiEM (Denitto et al., 2017a) detects both biclusters as one bicluster including the overlapped rows and columns and a second bicluster excluding the overlapped rows and columns.

(Kléma et al., 2017) proposed a method called semantic biclustering to detect interpretable rectangular patterns in binary data matrices. To do so, they used an existing biclustering method with the semantic ingredient i.e. enrichment, and a rule and tree learning machine learning. The authors assumed a joint probability distribution over a set of rows, a set of columns, and a binary set of expression indices. In order to test the quality of their biclusters, they counted the number of ones inside a bicluster and zeros outside of it as $\sum_{(i,j)\in ext(B)} e_{ij} + \sum_{(i,j)\in M\times N\setminus ext(B)} 1\text{-}e_{ij}$, where $ext(B) = \{(i,j): i\in m, j\in n, (m,n)\in B\}$ and $B = \{Bic\}$. They tested their methods on two real gene expression data and they indicated that the biclustering enrichment method achieves the best performance.

(Eric et al., 2017) presented a simple and interpretable convex optimization formulation of the biclustering that possessed a unique global minimizer ($F_\gamma(U)$) and a penalized regression ($J(U)$) problem. The minimizer is assumed to be continuous in the data. In their method, called COBRA, a single tunning parameter that controlled the number of biclusters, was generated that included an entire solution path of possible biclusters. In order to regenerate their results, COBRA always mapped data to a single biclustering assignment. Each element of the data matrix was assumed to be $e_{ij} = \mu_0 + \mu_{rc} + \varepsilon_{ij}$, where μ_0 is the grand mean shared by all elements and is equal to zero to model a checkerboard mean for the biclustering problem. $\mu_{rc} = 1/(|R||C|)\sum_{i\in R, j\in C} e_{ij}$, is the mean of the bicluster, and ε_{ij} is assumed to be an independent and identical distribution $N(0,\sigma^2)$ for some $\sigma^2 > 0$. The partitions were identified by minimizing the following convex function:

$$F_\gamma(U) = \frac{1}{2}\left\| Data - U \right\|_F^2 + \underbrace{\gamma[\Omega_W(U) + \Omega_{\widetilde{W}}(U^T)]}_{J(U)}, \Omega_W(U)$$

$$= \sum_{i<j} \left\| U_{.i} - U_{j.} \right\|_2$$

where $U_{.i}$ ($U_{i.}$) denotes the i^{th} column (row) of the matrix U and U is the estimate of the mean matrix μ. $J(U)$ penalizes deviations away from a checkerboard pattern and $\gamma \geq 0$ tunes the two terms. COBRA was considered as a principled reformulation of the clustered dendrogram. They showed the stability and reproducibility of biclustering on both simulated and real microarray data. However, their method was not applicable for overlapped biclusters.

2.2.4 Geometric based Biclustering

Geometric biclustering method is a recently introduced biclustering method based on transferring the element values of a data matrix to a higher dimensional space (Zhao et al., 2008, Gan et al., 2008, To and Liew, 2014). In this method, each column corresponds to a dimension in a higher dimensional space and the rows correspond to points in this higher dimensional space. Thus, different patterns of linear biclusters can be categorised as points, lines or hyperplanes in the new dimensional space.

(Liu et al., 2014, Wang and Yan, 2013, Wang et al., 2012, Wang and Yan, 2010, Zhao et al., 2008, Gan et al., 2005) considered the geometrical viewpoint of biclustering, i.e. hyperplane detection, leading to finding linear pattern biclusters in the data matrices. In computer vision, an effective way to detect linear structures is Hough transform (HT) (Hough, 1962, Hough, 1959). HT identifies lines in the data matrix by a voting procedure in Hough space. HT maps the x-y coordinates into ρ-θ parameter space where ρ is the distance of the line to the origin and θ is the angle of the normal vector with the x-axis. HT parametrizes the pattern space then projects all points into the parameter space, from which the local maxima in an accumulator implies the existence of line candidates.

Gan et al. (Gan et al., 2005) proposed to detect biclusters in two main steps. First, they detected a bundle of hyperplanes among a data matrix using fast Hough transform method. Then they analysed the detected planes to see whether the planes contain any coherent values by additive and multiplicative models. They used synthetic data matrices to validate their results.

Generally, HT is computationally expensive and the storage memory requirement is high. This makes HT un-scalable for high dimensional data, thus,

Zhao et al., (Zhao et al., 2008) applied Hough transform in a column pair space. The authors developed a visualization tool, additive and multiplicative pattern plot (AMPP), to figure out the collinear points and to classify points into additive and multiplicative patterns. The intersection of the rows and the union of the columns generated the maximal biclusters. In their method, they detected different types of biclusters one by one.

Wang and Yan (Wang and Yan, 2010) also applied Hough transform in column pair space to find sub-biclusters. A hypergraph model merged the sub-biclusters to generate larger biclusters. In (Wang et al., 2012, Wang and Yan, 2013, Liu et al., 2014), the authors extended their method and they built their graph regarding each Hough vector as a node. The graph spectrum is utilized to produce larger biclusters.

All these efforts in using HT show the effectiveness of HT in detecting biclusters. However, the heuristic combination strategies may fail with regard to detect and combine sub-biclusters to generate bigger biclusters and may converge to a local maximum.

In order to overcome the space usage and complexity of HT, evolutionary algorithms (EA) and heuristic search are introduced into the geometric biclustering (To and Liew, 2014).

In (To and Liew, 2014), To and Liew combined genetic algorithms (GA) and the steepest descent to obtain the parameters of the fittest hyperplane. In their method, the fitness function was the root mean square error (RMSE) of the hyperplane. However, using steepest descent for parameter selection of the hyperplane was very time consuming.

The algorithms based on geometric based biclustering are not only able to detect all types of linear biclusters but can also handle overlapping biclusters. However, the use of HT makes these methods not scalable and the heuristics based on divide-and-conquer may fail to detect maximal biclusters.

2.2.5 Evolutionary based Biclustering

Evolutionary algorithm (EA) is a popular metaheuristic method for global optimization because of its excellence ability to explore a search space and to solve complex problems (De Jong, 2006). Using evolutionary-based approaches to solve the biclustering problem especially in gene expression data analysis has attracted tremendous attention after the seminal work of Cheng and Church in 2000 (Cheng and Church, 2000). Hence, researchers have proposed to apply evolutionary algorithms as a search strategy to the biclustering problem (Golchin and Liew, 2017, Liew, 2016, Golchin and Liew, 2016, Golchin et al., 2015, Pontes et al., 2013, Seridi et al., 2011, Roh and Park, 2008, Mitra and Banka, 2006, Divina and Aguilar Ruiz, 2006). Depending on the fitness function defined in EA, these methods can detect all types of biclusters. EA search strategy includes artificial immune system (AIS) and genetic algorithms (GA).

2.2.5.1 AIS based Biclustering

Artificial immune system (AIS) is a subfield of evolutionary algorithms inspired by the biological immune system. The methods in artificial immune system can be classified into clonal selection method, negative selection method, immune network method, and dendritic cell method (De Castro and Timmis, 2002).

In (Coelho et al., 2008), the authors used artificial immune network to search for multiple biclusters concurrently. In their proposed method, called multi-objective multi-population artificial immune network (MOM-aiNet), they

minimized the MSR and maximized the bicluster volume through multi-objective search. To detect multiple biclusters concurrently, the authors generated a subpopulation for each bicluster by randomly choosing one row and one column of the data matrix and running the multi-objective search on each subpopulation separately. In each iteration, each subpopulation underwent cloning and mutation. Then, all non-dominated biclusters was used to generate the new population for the next iteration. The method aimed to converge to distinct regions of the search space. To do this, MOM-aiNet compared the degree of overlap of the largest biclusters of each population. If the overlap value was greater than a threshold, two subpopulations were merged to a single subpopulation. The method also generated new random subpopulations from time to time.

Liu et al (Liu et al., 2011) proposed a dynamic multi-objective immune optimization biclustering method (DMOIOB). They detected maximized biclusters with minimized MSR and maximized row variance. A binary string encoded the antibodies of a fixed number of rows and a fixed number of columns. Their method started by generating antibodies population and antigen population. Then the size of the antibodies population increased to ensure a sufficient number of individuals and to explore unvisited areas of the search space. For each antibody, best local guide was selected using Sigma method and the best antibodies were used to produce the next generation. Non-dominated individuals were used to update the antigens population. Moreover, the size of the antibodies population was decreased to prevent excessive growth in population. In order to find the local best solution, they applied the basic idea of Sigma method and immune clonal selection among the archive individuals. The quality of the objective values and a biological analysis of the biclusters were

used to validate their method. DMOIOB achieved the diversity of solutions by using the concept of crowding distance and ε-distance.

However, using only these distances do not guarantee diversity among archive individuals. For example, if there are two biclusters in a data matrix with the same volume and the same values, then these biclusters have the same objective values in the objective space and the method ignores one of them.

2.2.5.2 GA based Biclustering

Genetic algorithm is a subfield of evolutionary algorithms inspired by natural evolution to generate reliable solutions to optimization problems and search problems by relying on mutation, crossover, and selection. In a genetic algorithm, a population of solutions evolve toward better solutions over several generations. Each solution has a set of properties encoded as a binary string or other encoding schemes.

Divina and Aguilar-Ruiz (Divina and Aguilar Ruiz, 2006) proposed an evolutionary computation method that combined the bicluster size, MSR, and row variance in a single objective cost function. In their method, the authors found biclusters with bigger size, higher row variance, smaller MSR and low level of overlapping among biclusters. They used Equation (2.3) to calculate MSR value and Equation (2.8) to calculate row variance of bicluster $Bic = (R,C)$.

$$var_{RC} = (\sum_{i \in R, j \in C} (e_{ij} - e_{iC})^2)/|R||C| \quad \textbf{(2.8)}$$

In order to avoid overlapping, the authors used a penalty value as the sum of weight matrix associated with the expression matrix ($penalty = \sum w_p(e_{ij})$). The weight of an element depended on the number of biclusters containing that

element and Equation (2.9) was used to update the weight matrix. In this equation, $|Cov(e_{ij})|$ is the number of biclusters containing e_{ij}.

$$w_P(e_{ij}) = \begin{cases} \dfrac{\sum_{n\in N, m\in M} e^{-|Cov(e_{nm})|}}{e^{-|Cov(e_{ij})|}} & if\ |Cov(e_{ij})| > 0 \\ 0 & if\ |Cov(e_{ij})| = 0 \end{cases} \quad \textbf{(2.9)}$$

The overall fitness function was $F_f(B) = MSR(B)/\delta + 1/var_B + w_d + penalty$, where $w_d = w_v(w_r(\delta/R) + w_c(\delta/C))$, and w_v, w_r and w_c are weights that were assigned to the bicluster. However, due to the conflicting nature of the fitness criteria, their single-objective function method does not produce optimal biclusters.

In (Roh and Park, 2008), the authors proposed ECOPSM based on evolutionary computation algorithm using the order preserving submatrix (OPSM) constraint based on a ranking matrix. According to them, a bicluster is a group of rows with strictly increasing values across a set of columns. Using evolutionary computation algorithm they searched for biclusters with a certain column length. The method evaluates the probability of a bicluster participating in the best OPSM by different permutation of OPSMs length from two to c such that it maximized this score. Here, each individual was encoded as an L-length permutation columns index. This encoding reduced the search space to $\sum_{L=2}^{c} cP_L$ where P stands for permutation. The fitness function ($F_f(L)$ = count(RCM,L)) was the number of rows stored in RCM, each of which had a subsequence equal to L. Single point crossover and single bit mutation were used to generate the next population. ECOPSM evaluated the results based on the size of the detected biclusters and their biological relation.

2.2.5.2.1 Multi-objective based Biclustering

Some of the objectives in a biclustering problem are conflicting and we cannot combine them into a single function. In fact, in many real-world situations optimization of two or more conflicting objectives is required. This leads to a multi-objective biclustering problem.

In (Mitra and Banka, 2006), the authors proposed a multi-objective non-dominated sorting genetic algorithm (NSGA) with local search strategy based on CC method. NSGA-II (Deb et al., 2002) was based on the use of a non-dominated crowding distance to retain the diversity among Pareto front, and a crowding selection operator. In their method, a binary vector encoded the biclusters with a fixed size equal to the number of rows plus the number of columns in the data matrix. A value of one indicates that corresponding row or column is present in the bicluster and zero otherwise. Homogeneity (MSR - Equation (2.3)) and size ($|R| \times |C|$) are the primary objectives. Single point crossover and single bit mutation are the genetic operators. As each individual only encoded one bicluster, this method only searched for one bicluster at a time, and it is not clear how the authors detected multiple biclusters in a data matrix.

In (Maulik et al., 2009), Maulik et al., proposed a multi-objective genetic biclustering method. In their method, they used variable length string that encoded the centre of M row clusters and the centre of N column clusters, thereby representing $M \times N$ biclusters in one individual. Their method optimized two conflicting objectives by minimizing MSR (Equation (2.3)) and maximizing the row variance of the bicluster (Equation (2.8)). The search strategy conducted NSGA-II with two points' crossover and single bit string mutation. Rows and columns underwent crossover and mutation separately. The average of the fitness of the biclusters encoded in the string gave the fitness of a string. Final

biclusters included every biclusters encoded in the individuals that constitute the Pareto front. They validated their results both biologically and statistically. It is not clear how the method handled similar biclusters or suboptimal biclusters in the final solution since Pareto optimality is only with respect to the individual and not with respect to the biclusters encoded in an individual.

In (Seridi et al., 2011), Seridi et al. used a combination of minimizing similarity (Equation (2.3)), maximizing size (number of elements $|R| \times |C|$), and maximizing row variance (Equation (2.8)) as three objectives in a multi-objective biclustering method. Maximizing row variance required significant fluctuation among set of columns, which was a property of additive pattern biclusters. They used the index of rows and columns as the encoding of the individuals, a single point crossover operator, and a heuristic search based on CC method as the mutation operator and NSGA-II/IBEA (Deb et al., 2002) as the multi-objective methods. Their method returned a set of solutions in the approximate Pareto front referring to one bicluster. Only statistical validation was performed on their results.

2.3 Evolutionary Optimization

Evolutionary algorithms are efficient optimization and powerful search methods that have the ability to find near optimal biclusters in data matrices. So far, we reviewed bicluster details i.e. what is a bicluster, different patterns in a bicluster, how to validate a detected bicluster, and some existing biclustering methods. In this section, we will discuss evolutionary algorithms in single objective and multi-objective methods.

2.3.1 Single Objective Methods

Among different optimization methods, evolutionary algorithms have become very popular in the last decades (Ursem, 1999). It has been used for global optimization and has been able to successfully solve difficult problems in many applications with great complexity. Generally, EA is a population-based trial and error metaheuristic optimization technique that incorporates the process of reproduction, mutation, crossover (recombination) and elitism (selection). Candidate solutions to the optimization problem are defined as individuals, which are generated and updated through an iterative process. The quality of each individual is determined by a fitness function. EA works in such a way that the fitness value of the population improves gradually in each iteration of the algorithm. The general steps of the EAs are explained in the remaining of this section.

2.3.1.1 Initial Population Generation

In EA, each candidate solution is encoded as an individual, which is a member of the set of possible solutions in the solution space of a given problem. In the biclustering problem, we try to find subsets of rows and columns from data matrices that satisfy our fitness function constraints. In this case, rows and columns belonging to a bicluster from a data matrix is encoded as a candidate solution. Concerning the big size of the real data matrices, random rows and columns are assigned to each individual.

Due to the big size of the solution space in the biclustering problem, pre-defined number of random individuals are generated in the population. In each iteration, the population is updated by removing the individuals with the less desired fitness value and introducing new individuals through the mutation and crossover process. The number of individuals in a population remains fixed

throughout the evolutionary process. Using a population of solutions helps the EA avoid becoming trap at a local optimum and it usually includes individuals in different regions of the solution space.

In our study, we consider two populations, a normal population and an external population, archive, which behaves as elitism in traditional GAs. The normal population includes both dominated and non-dominated individuals with a larger number of individuals while the archive includes only best individuals with the smaller number of individuals. The concept of domination will be explained in Section 2.3.2.

After adding all non-dominated individuals, due to the fix size of the archive, if the archive overflows, the individuals with the worse fitness value will be removed from the archive until the archive reaches the maximum number. Otherwise, dominated individuals from the previous population and the previous archive are added to the new archive based on their fitness value.

2.3.1.2 Mutation

Mutation is an operator that helps preserving the diversity from individuals of one population to individuals of the next population. Mutation alters one or more bit values in an individual. The new individual can become a better solution by using mutation. Mutation occurs based on a user-defined mutation probability. The most common way to do the mutation is through bit flips at a random position. The probability of mutation of a bit is $1/n$, where n is the number of bits.

2.3.1.3 Crossover

Crossover is an operator that helps to vary the programming of an individual from one population to the next population. In crossover, we need to take two

individuals (usually from the better individuals, which in our study are from the archive individuals) as parents and producing a child individual from them. The most common way to do the crossover operation is through a single-point crossover. A single point on both parents is selected. All bit values beyond that point is swapped between the two parents. The resulting individuals are two offspring or children.

2.3.1.4 Fitness Function

A fitness function or objective function is a function that assigns a real value to an individual in a way that summarizes how close an individual is to achieve a good solution. Using fitness value, individuals inside a population can be compared. In evolutionary algorithms, the fitness function is very important to guide and expedite the evolution towards an optimal solution.

In a single-objective minimization problem, we try to find $\min[F_f(x)]$ where F_f is a scalar fitness function, $x \in S_s$ and $S_s = \{x \in \mathbb{R}^m: h(x) = 0, g(x) \geq 0\}$. However, most common objective functions that are applicable in biclustering problem are the coherence of a bicluster and the size of a bicluster, which are in conflict with each other. Therefore, a single objective fitness function is not very useful for the biclustering problem. In recent biclustering methods, (Maulik et al., 2009, Maulik et al., 2013, Seridi et al., 2015, Coelho et al., 2008, Golchin et al., 2015, Liu et al., 2009b, Liu et al., 2012, Liu et al., 2011, Liu et al., 2008, Mitra and Banka, 2006, Seridi et al., 2012, Seridi et al., 2011), two or more conflicting objective functions are considered to optimize the biclusters simultaneously. Hence, the multi-objective evolutionary optimization methods (Mukhopadhyay et al., 2014b, Mukhopadhyay et al., 2014a) have combined with the evolutionary biclustering methods. In the following, we will have a general explanation of multi-objective fitness functions and the concept of domination.

2.3.2 Multi-objective Methods

Multi-objective search strategy handles multiple conflicting objectives, which is often encountered in biclustering problem. Similar to single objective methods, multi-objective methods also use many of the operations we discussed in the previous section. The only difference is the way fitness function is assigned to each individual. Multi-objective optimization problem is denoted as $\min[F_{f1}(x), \ldots, F_{fk}(x)]$, $x \in S_s$ and $k > 1$. Generally, in multi-objective optimization, a solution may be better in one objective value but worse in another objective value. In multi-objective optimization, there is no unique single optimum solution; instead, there exists a set of solutions to constitute the Pareto front solutions by employing the concept of domination. We can formally define the Pareto front by

Definition 1. Objective space (S_s): space of objective vectors.

Definition 2. Pareto optimal: $x^* \in S_s$ is Pareto optimal if $F_{fi}(x^*) \leq F_{fi}(x)$ for all $i \in \{1, \ldots, k\}$ (k is the number of objectives) and at least one $F_{fi}(x^*) < F_{fi}(x)$.

Definition 3. Domination: $x^* \in S_s$ dominates $x \in S_s$ ($x^* \prec x$) if x^* is Pareto optimal. x^* is called a non-dominated individual and x is called a dominated individual.

Definition 4. Pareto front solutions: $x^* \in S_s$ is Pareto front solutions if $x^* \prec x$ where $x \in S_s$.

Given the above, $x^* \prec x$ (x^* dominates x) when the entire objective values in the individual x^* are smaller or equals from the corresponding objective values in the individual x and there is at least one objective value in the individual x^* that is absolutely smaller than the corresponding objective value in the individual x in a minimization problem.

Mathematically, the different solutions in a Pareto front represent different trade-offs on the conflicting objective values and are therefore equally good. For solutions in the Pareto front, we cannot improve any objective value without degrading some of the other objective values. Figure 2.5 visualizes a two objective space f_1 and f_2 in a minimization problem. There are three non-dominated individuals as the hollow dot solutions in this figure. The line illustrates the Pareto front that contains all the three non-dominated individuals. Solid dot solutions represent the dominated individuals. Both objective values of individual *C* are larger in comparison to the corresponding objective values of individual *B*, so individual *B* dominates individual *C*. Individual *A* has larger f_1 value in comparison to individual *B* while the f_2 value of individual *A* is smaller than f_2 value of individual *B*. As the result, neither *A* nor *B* dominates each other.

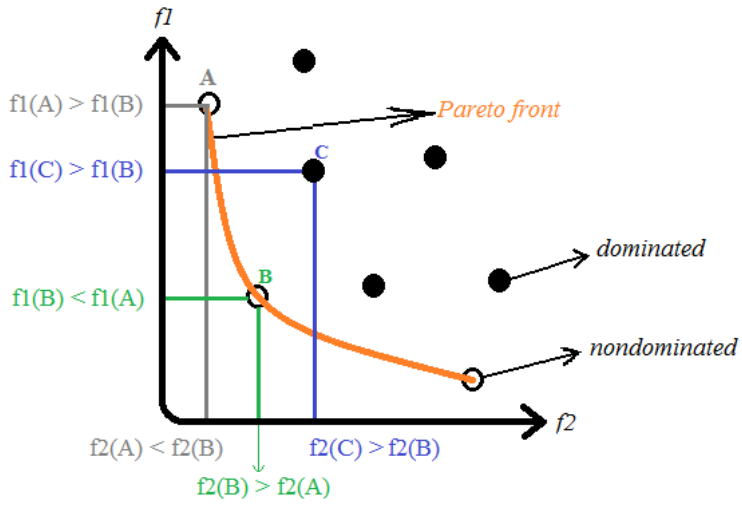


Figure 2.5. Visualization of a two objective space f_1 and f_2 for a minimization problem

If a problem requires a unique solution, then a single solution can be selected from among the Pareto front solutions based on some subjective preferences defined by the problem or the human decision maker.

2.4 Conclusion

In this chapter, we explored the background knowledge behind our biclustering methods. In addition, a wide range of methods that have been proposed to tackle the biclustering problem were reviewed and their merits and shortcomings highlighted. It is shown that among the different strategies, geometric based biclustering and evolutionary based biclustering are easy to implement and they achieved good results. Moreover, these two methods are able to address the main concerns in biclustering problems, which are the noise in the data, the overlapping of biclusters, and the high dimension of real data.

In the case of geometric based biclustering, almost all of the recent studies have used Hough transform to detect the hyperplane. Despite the accuracy of detected hyperplane, the space and memory requirement of HT is huge and it is not scalable for the high volume of real life data without using some heuristic procedures.

A good merit function is critical for evolutionary based biclustering methods. Although there are a number of EA methods being proposed, most of these methods extract biclusters sequentially rather than concurrently (Mitra and Banka, 2006, Seridi et al., 2011, Golchin et al., 2015, Maulik et al., 2013, Seridi et al., 2015). Thus, there is still a need for novel methods to search for better biclustering solution. By addressing these issues and proposing novel methods, we expect to contribute even further to solve the biclustering problem.

In the next chapter, we aim at introducing our first biclustering method called PBD-SPEA based on a multi-objective evolutionary algorithm and a new dynamic individual encoding scheme to tackle the biclustering problem.

3

Biclustering based on Strength Pareto Front Algorithm

In Chapter 2, we have discussed the biomedical data analysis, the biclustering problem, and the evolutionary optimization. We have also reviewed different algorithms that had been proposed to address the biclustering problem. We have shown that using multi-objective evolutionary algorithms had been one of the most important approaches used to tackle this problem. However, exploring different areas of the objective space and finding multiple biclusters concurrently instead of one by one had not been investigated adequately. In this chapter, we aim at finding multiple biclusters concurrently using evolutionary algorithm. To do this, we introduce a dynamic encoding scheme to encode all biclusters in an individual. We then propose three objectives fitness function to optimise the randomly generated biclusters. A new exploits and explore crossover operation and a heuristic mutation operation is proposed. Finally, we show the effectiveness of our proposed method. This chapter is based on our published paper (Golchin and Liew, 2017).

3.1 Introduction

Biclustering has become a popular method to analyse gene expression data and extract valuable information. Evolutionary based biclustering methods are considered the most popular methods in biclustering (Araújo et al., 2011, Divina and Aguilar Ruiz, 2006, Golchin et al., 2015, Golchin and Liew, 2016, Golchin and Liew, 2017, Liew, 2016, Mitra and Banka, 2006, Pontes et al., 2013, Roh

and Park, 2008, Seridi et al., 2011). Using a good merit function together with a suitable iterative search can lead to the detection of interesting biclusters.

In this study, a multi-objective evolutionary algorithm (Zitzler et al., 2001) with local search called PBD-SPEA (Parallel Biclustering Detection using Strength Pareto front Evolutionary Algorithm) is proposed. Unlike most existing multi-objective biclustering methods, in our method, a new dynamic encoding scheme is used to encode multiple biclusters in each individual. The new encoding scheme allows our method to search for multiple biclusters in a data matrix concurrently during each iteration of the evolutionary algorithm. Our multi-objective function consist of three objectives that simultaneously optimize the homogeneity of the elements in the bicluster, the size of the bicluster, and the variance of the column in the bicluster with respect to the entire data matrix.

We also introduced novel crossover and mutation operations to ensure individuals in the next generation are fitter than their parents. Crossover is done by selecting and combining the best biclusters among the encoded biclusters from both parents through a strategy of exploration and exploitation. Finally, a sequential selection procedure is used to select the final set of biclusters from the individuals that constitute the Pareto front.

In this chapter, in addition to the synthetic and real gene expression data, we will investigate the impact and generality of our proposed method on two non-medical dataset including an image dataset and a Facebook dataset.

3.2 The Proposed Method

We propose a method, denoted as PBD-SPEA that uses the strength Pareto front algorithm (SPEA2) as our multi-objective optimization method. One advantage

of SPEA2 is that the distribution of Pareto front solutions in SPEA2 is wider and more uniform than NSGA-II, especially with a larger number of objectives (Konak et al., 2006). In addition, SPEA2 has faster convergence rate in higher dimensional objective space (Zitzler et al., 2001).

3.2.1 Initial Population Generation

In our method, each individual encodes a number of biclusters using a numeric coding scheme as shown in Figure 3.1. The first number in the coding scheme refers to the number of rows in a bicluster, the second number refers to the number of columns in a bicluster, the first set of numbers indicates the row indices of the bicluster, the second set of numbers indicate the column indices of the bicluster, and a zero separates the biclusters. For example, an individual as shown in Figure 3.1 contains two biclusters of size 4×3 and 2×3, where the first bicluster has row indices of {1, 3, 6, 8}, and column indices of {2, 4, 9}, and the second bicluster has row indices of {3, 6} and column indices of {7, 2, 3}. Our encoding of each bicluster is similar to (Seridi et al., 2012, Seridi et al., 2011, Seridi et al., 2015), but in their methods, each individual only encodes one bicluster.

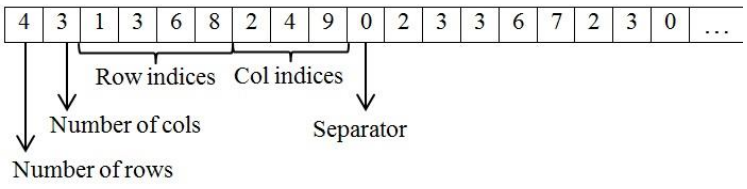


Figure 3.1. The representation of individuals

PBD-SPEA uses two populations as described in Section 2.3.1.1. The initial population of individuals, where each individual encodes all biclusters and an empty archive are generated randomly with the user-defined number of biclusters.

3.2.2 Mutation

In order to generate the next generation, mutation and crossover operations are applied to each individual. Mutation is done based on an improved heuristic search based on the local search of (Cheng and Church, 2000) to remove unwanted rows or columns of a bicluster or to grow the bicluster. For node deletion, when the MSR of a bicluster is bigger than a user defined threshold, a row or column is randomly selected from the bicluster and its row or column residue is calculated. If the row or column residue is bigger than the mean square residue, the selected row or column is removed from the bicluster. For node addition, if the MSR of a bicluster is smaller than a user-defined threshold, rows or columns are added to the bicluster. First, a row or column not in the bicluster is randomly chosen. If the row or column residue is smaller than the mean square residue, the row or column is added to the bicluster. The steps of the local search are summarized in Algorithm 3.1. α and β determine the rates of deleting and adding rows and columns respectively. In (Mitra and Banka, 2006), α is set to 1.4, and β is set to one. The same parameter values are used in this work as well. However, a higher value for α is selected that decreases in each step of the loop to increase the efficiency of the method by removing the rows and columns with higher MSR values first. This makes the MSR of the bicluster to reach the desired value faster. MSR is calculated based on Equation (2.3).

Algorithm 3.1: Mutation

Input: A random individual that is selected for mutation

Output: More coherent individual

For each bicluster *Bic* in the individual:

// Nodes deletion

While the MSR ≥ user defined threshold

  Calculate the row residue for a random row *i* selected from *Bic* by $\frac{1}{|C|}\sum_{i\in R}(e_{ij} - e_{iC} - e_{Rj} + e_{RC})^2$

  If row residue ≥ $\propto\times MSR(R, C)$, remove the row from the bicluster

  Recalculate the values of $e_{ij}, e_{iC}, e_{Rj}, e_{RC}$ and $MSR(R, C)$

While the MSR ≥ user defined threshold

  Calculate the column residue for a random column *j* selected from *Bic* by $\frac{1}{|R|}\sum_{j\in C}(e_{ij} - e_{iC} - e_{Rj} + e_{RC})^2$

  If column residue ≥ $\propto\times MSR(R, C)$, remove the column from the bicluster

  Recalculate the values of $e_{ij}, e_{iC}, e_{Rj}, e_{RC}$ and $MSR(R, C)$

// Nodes addition

While the MSR ≤ user defined threshold

  If the row residue for a random row not in *Bic* is < $\beta \times MSR(R, C)$ then add the row to the bicluster

  Recalculate the values of $e_{ij}, e_{iC}, e_{Rj}, e_{RC}$ and $MSR(R, C)$

While the MSR ≤ user defined threshold

  If the column residue for a random column not in *Bic* is < $\beta \times MSR(R, C)$ then add the column to the bicluster

  Recalculate the values of $e_{ij}, e_{iC}, e_{Rj}, e_{RC}$ and $MSR(R, C)$

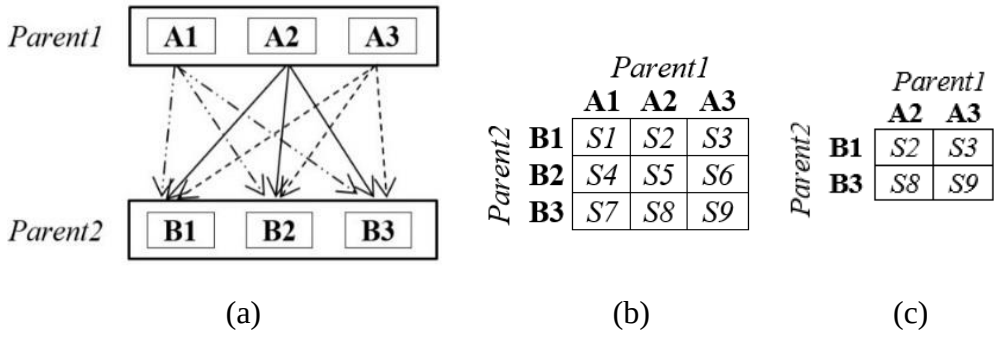
3.2.3 Crossover

For crossover operation, two parents are selected from the archive using binary tournament selection by replacement from the archive. In our method, crossover of two parents is done through crossover between pairs of similar biclusters from

the two parents. The similarity value between two biclusters of two parents is given by

$$Similarity\ value = \frac{\left|x_{P_1} \cap x_{P_2}\right|}{\min(\left|x_{P_1}\right|, \left|x_{P_2}\right|)} \quad \textbf{(3.1)}$$

where x_{P1} is the row or column indices of the first parent and x_{P2} is the row or column indices of the second parent. |.| denotes the size of a set. The similarity value measures the degree of overlap of two sets. An illustration of similarity search and the corresponding similarity table are shown in Figure 3.2. The similarity value is calculated for each pair of biclusters from the parents and a similarity table is constructed based on these values. The pair of biclusters with the largest similarity value is first obtained from the table. This similarity value is then compared to a present threshold τ. If the value is larger than the threshold τ, the best bicluster in the pair is copied into the offspring. If the value is less than τ, a single point crossover is applied to the pair to generate the bicluster in the offspring. Once the pair of biclusters is processed, the corresponding row and column are removed from the table. The process is repeated until the table has only a single item.



(a) (b) (c)

Figure 3.2. An example of similarity search in parents and the resulting similarity table. (a) Search procedure, (b) similarity table, (c) the updated similarity table after *S4* is found to have the largest similarity value.

When the pair of biclusters is not similar enough, i.e. its similarity value is less than τ, a single point crossover is applied to both the row and column of the biclusters as illustrated in Figure 3.3 to generate the bicluster in the offspring. In this figure, row index 6 from the first parent is randomly selected. Then in parent two all the row indices bigger than 6 are selected, and the row indices of the first bicluster in the offspring would consist of {1, 3, 6} from parent one and {7, 9} from parent two. The same operation goes for the columns of the bicluster as well. Algorithm 3.2 shows the steps of crossover operation.

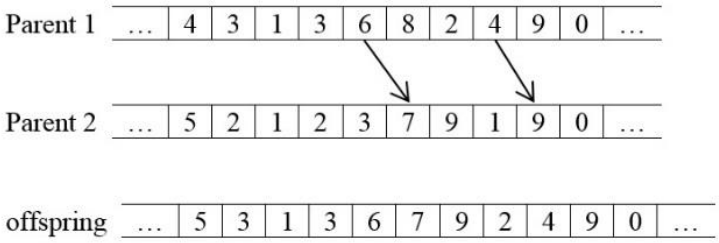


Figure 3.3. Single-point crossover

As τ determines whether the offspring inherits the best bicluster from the pair of similar biclusters or a new bicluster should be generated using random crossover of the bicluster pair, it can be viewed as a parameter that balances exploitation vs exploration. When similarity is low, we would like to encourage exploration to find better solution. On the other hand, when similarity is high, we want to exploit the good solution that we already have. During the early stage of iterative search, the similarity between pair of biclusters is generally low, and the crossover operation is mostly about exploration. As iteration proceeds and the pair of biclusters becomes more similar, exploitation becomes more frequent. This procedure ensures that the generated offspring contains biclusters as good as the pair of similar biclusters from the parents.

Algorithm 3.2: Crossover	
Input:	Two parents selected based on binary tournament selection by replacement
Output:	The resulting offspring
1	Construct the similarity table for each pair of biclusters from the parents
2	For all the values in the similarity table do
3	Find the largest similarity value

4	If the similarity value is larger than a similarity threshold τ
5	Copy the best bicluster into the offspring
6	Else
7	Perform single point crossover on the two biclusters to generate a bicluster in the offspring
8	Remove the corresponding row and column from the table

3.2.4 Fitness Function

In our method three objective functions are used, namely MSR score, bicluster volume score, and a variance score. The MSR of a bicluster is computed using Equation (2.3). The second objective is calculated by the size equation as introduced in (Divina and Aguilar Ruiz, 2006).

$$w_d = w_r \times \delta/m + w_c \times \delta/n \quad \textbf{(3.2)}$$

where w_r and w_c are weights used to balance the number of rows and columns. δ is the predefined MSR threshold that we try to keep the MSR value of the biclusters below this value. The value of w_r is set to one as in (Divina and Aguilar Ruiz, 2006) since in almost every real data matrix the number of rows is bigger than the number of columns. w_c is the ratio of the number of rows to the number of columns in the data matrix, and it varies from one to 10. w_d is inversely related to the number of elements in the detected bicluster. The variance score is calculated by Equation (3.3) which is based on the relevance index (Yip et al., 2004). Variance score is calculated for the columns of the bicluster and it is the sum of variances of each column in the bicluster over the variance of that column in the database. The smaller the score is, the more identical the elements of a bicluster are in comparison to the data matrix. In this equation, σ_{ij} is the local variance of bicluster *i* under column *j* and it is calculated based on the variance

of the columns in the bicluster. σ_j is the global variance of column j which is calculated based on the variance of the columns in the data matrix.

$$VarianceScore = \frac{1}{n}\left(\sum_{j=1}^{n} \frac{\sigma_{ij}}{\sigma_j}\right) \quad \textbf{(3.3)}$$

Variance score shows the closeness between the expression values of a column among the selected rows. The score is small when the local variance is small in comparison to the global variance. The multi-objective cost function is calculated using Equation (3.4). The SPEA2 method tries to minimize all the objectives, so the smaller the cost value is, the better the detected bicluster is.

$$ObjVal = \begin{cases} f_1 = MSR(R, C) \\ f_2 = w_d \\ f_3 = VarianceScore \end{cases} \quad \textbf{(3.4)}$$

The objective value is calculated for each bicluster of each individual using Equation (3.4). In order to calculate the overall cost function of an individual, Equation (3.5) is used to combine the cost functions of all biclusters in the individual. In this equation C_j^i is the cost function of the i^{th} bicluster of the j^{th} individual. $\#B_C$ is the number of biclusters. The square root suppresses noise in the biclusters by de-emphasizing its effect in the overall cost function, and our experiments show that it generally resulted in biclusters of bigger size.

$$Cost_j = \sqrt{\frac{1}{\#B_C}\sum_{i=1}^{\#B_C} C_j^i} \quad \textbf{(3.5)}$$

3.2.5 Fitness Value Assignment

Generally, in a multi-objective method, two types of cost function are defined, a vector-valued objective function and a fitness function. Objective values determine the fitness value. In order to calculate the Pareto front individuals, For each individual i in the archive population, a_p, and in the normal population, n_p, a strength value $S(i)$ is assigned. The strength value represents the number of individuals that individual i dominates as in Equation (3.6):

$$S(i) = |\{ j \mid j \in a_p \cup n_p \wedge i \prec j\}| \quad \textbf{(3.6)}$$

where $|\bullet|$ shows the number of individuals, $\prec$ stands for the domination. i and j are individuals in the population and archive. The raw fitness value for individual i is determined by summing the strength values of individuals that dominate the individual i as in Equation (3.7).

$$R(i) = \Sigma_{j \in ap \cup np \wedge j \prec i} S(j) \quad \textbf{(3.7)}$$

Note that we are going to minimize the fitness value so $R(i) = 0$ corresponds to a non-dominated individual. Larger values of $R(i)$ shows that individual i is dominated by many individuals. If there are only a few individuals that dominate each other then there exist many individuals with the same $R(i)$ value. In this case, Zitzler et al., (Zitzler et al., 2001) use a density value $D(i)$ based on the inverse of the k^{th} nearest neighbor method as in Equation (3.8).

$$D(i) = 1 / (d_{ik} + 2) \quad \textbf{(3.8)}$$

d_{ik} is the Euclidian distance of individual i to the k^{th} nearest neighbor. The value of k calculates as the square roots of the sample size ($N = |a_p| + |n_p|$), which is $k = \sqrt{N}$. $|\bullet|$ indicates the number of individuals in the population. $D(i)$ is always smaller than one and this value keeps the diversity among the individuals. When

the raw fitness of individuals i and j are the same and the d_{ik} distance of individual i is larger than the d_{jk} distance of individual j, the density value of the individual i is smaller than the density value of the individual j. As a result, the method selects individual i. Finally summing up the values of $D(i)$ and $R(i)$, the final fitness value is calculated as in Equation (3.9).

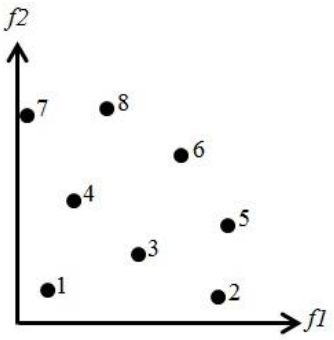
$$F(i) = R(i) + D(i) \quad \textbf{(3.9)}$$

Algorithm 3.3 summarizes the steps of the fitness value assignment to individuals. The worst case computational time of this algorithm is $O(N^2 \log N)$.

Algorithm 3.3: Fitness value assignment

1 For each individual i in the population and the archive do
2 $S(i) = |\{ j \mid j \in a_p \cup n_p \wedge i \prec j\}|$
3 $R(i) = \Sigma_{j \in ap \cup np \wedge j \prec i} S(j)$
4 $D(i) = 1 / (d_{ik} + 2)$
5 $F(i) = R(i) + D(i)$

For instance, in Figure 3.4, we show how to calculate strength value and raw fitness for each individual in the objective space. In this figure, nodes {1, 2, 7} are the Pareto front individuals.



Node No	Strength Value	Raw Fitness
1	5	0
2	1	0
3	2	5
4	2	5
5	0	8
6	0	9
7	1	0
8	0	8

Figure 3.4. The fitness assignment scheme (the strength value and the raw fitness value) for a minimization problem with two objectives f_1 and f_2

3.2.6 Final Biclusters Selection

After the main loop terminated, a post-processing step selects the final biclusters from the set of Pareto front individuals. The Pareto front consists of a set of individuals and not all biclusters in an individual on the Pareto front are optimum. To obtain the final set of biclusters, sequential selection of a set of best biclusters from individuals in the Pareto front is performed. This is done as follows. First, two individuals are randomly select from the Pareto front and pairs of similar biclusters from the two individuals are identified based on Equation (3.1). Then, in each pair, only the best bicluster is retained. The list of best biclusters obtained is then compared with the next individual randomly selected from the Pareto front to obtain a new set of best biclusters. This procedure is repeated until all individuals in the Pareto front are examined. Although sequential selection is only locally optimum, the set of biclusters obtained from the Pareto front using this procedure has been found experimentally to be of good quality.

3.2.7 The Overall Method

The flowchart of our method is summarized in Figure 3.5.

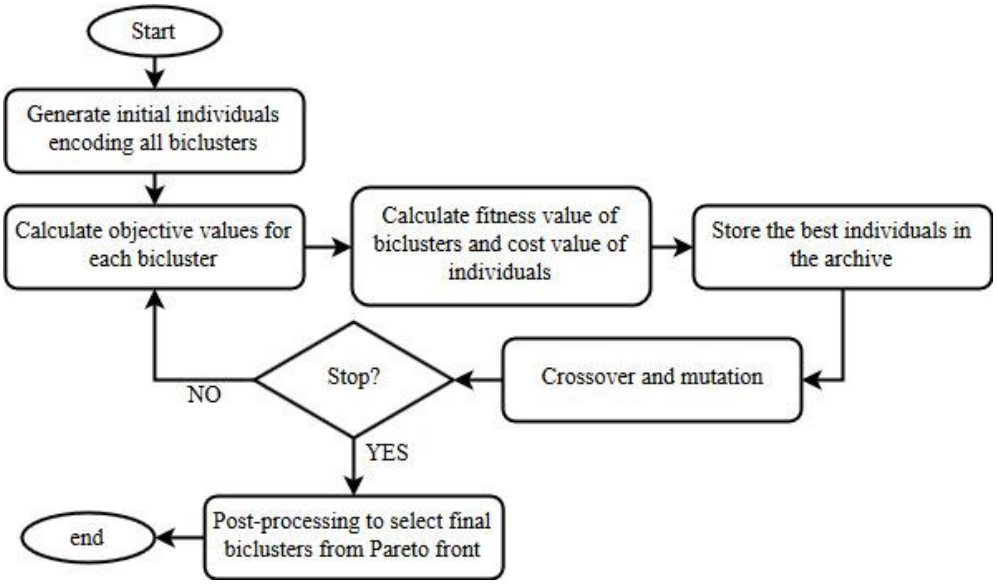


Figure 3.5. The overall flow diagram of the PBD-SPEA method

The steps of the proposed PBD-SPEA are summarized in Algorithm 3.4.

Algorithm 3.4: PBD-SPEA

Input:	Data matrix; δ; # biclusters; Similarity threshold; Population size; Archive size; Maximum iteration
Output:	Set of final biclusters
1	Generate initial population (*pop*) randomly where each individual contains a set of biclusters, and create the empty archive (*arch*)
2	Calculate objective values of each bicluster in each individual based on Equation (3.4)
3	For i = 1 : maximum number of iteration do

4	Compute cost function of each individual using Equation (3.5).
5	Evaluate each individual by calculating the strength value, raw fitness and density value based on Section 3.2.5.
6	Copy all non-dominated individuals from $arch_i$ and pop_i to $arch_{i+1}$. If the size of $arch_{i+1}$ exceeds the archive size, reduce the size $arch_{i+1}$ by the truncation operator, otherwise, if the size of the archive is less than archive size, fill $arch_{i+1}$ with dominated individuals from $arch_i$ and pop_i as described in Section 2.3.1.1.
7	Copy non-dominated individuals to Pareto front
8	Perform the crossover (Algorithm 3.2) and mutation (Algorithm 3.1) to generate the next population
9	Add archive population to the next population
10	Find the set of final biclusters from individuals in the Pareto front by sequential selection
11	Return the set of biclusters

3.3 Results and Discussion

We apply PBD-SPEA on three different real world datasets, which consists of a gene expression dataset, and a multimodal image dataset, and a big Facebook dataset.

Multimodal data are data that includes different modality such as audio, image, and text. The different modality in the data can potentially review different aspect of the underlying concepts in the data, but the feature vectors derived from multimodal data are usually of high dimension and therefore suffer from the curse of dimensionality (Zhu et al., 2017). For the image dataset, a set of keywords (annotation words) is extracted and is used to form the feature vector that describes the image. PBD-SPEA is then used to cluster the images so that images that share the same concepts are grouped together.

In recent years, social networks such as Facebook, twitter and so on have attracting millions of users. This leads to the creation of web based applications to be offered to social network users. Data analysis in social network is concerned with studying the users' behaviour and the usage patterns to design new tools and applications (Bozkir et al., 2010). We run PBD-SPEA on the Facebook dataset to discover how users are grouped together based on different subsets of features.

There are a number of parameter settings to control the EA search. These values are determined experimentally and are set to: Maximum Iteration =150, Population Size =100, Archive Size =40, Mutation Probability =0.2, Crossover Probability =0.8.

The effects of these parameters are discussed in Table 3.1. In this table, the performance metric is the mean value of Jaccard index of the detected biclusters. The dataset used in this experiment is SD2. If these values are too small, then we will have a fast and premature convergence. In order to have the right balance between exploitation and exploration, the parameters are selected as above.

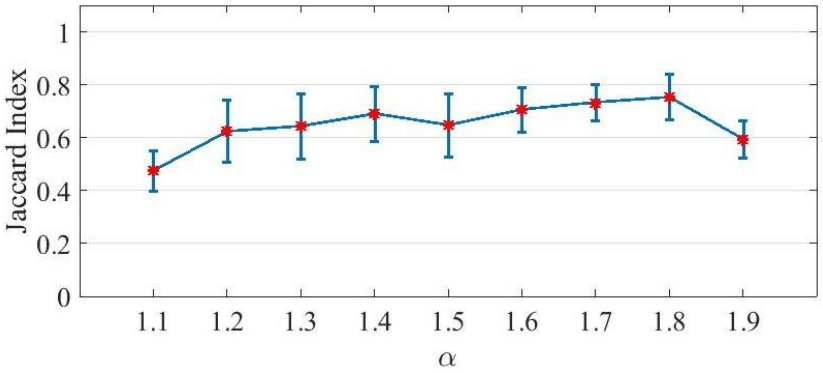
Table 3.1. The effects of the parameters on the method's performance

Parameter	**Value**	**Performance**
Maximum iteration	50	0.75
	100	0.85
	150	0.91
	200	1
Population size	50	0.7
	100	0.77
	150	0.8

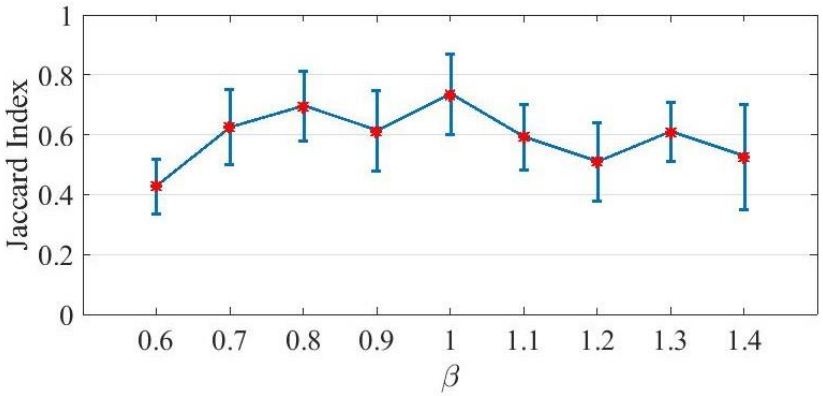
Parameter	Value	Performance
	200	0.93
Archive size	10	0.95
	20	0.98
	30	0.97
	40	0.99
Crossover probability	0.6	0.87
	0.7	0.92
	0.8	0.98
	0.9	0.82

3.3.1 Parameter Setting

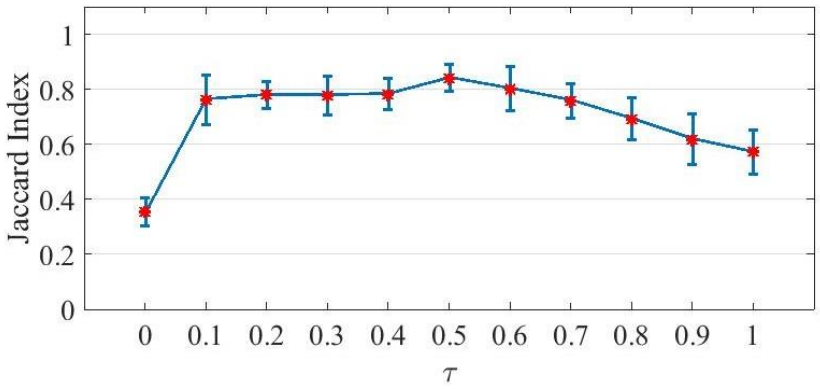
The three parameters that control the quality of the biclusters are α, β, and the similarity threshold τ. Figure 3.6 examines their effects on the quality of the biclusters using the Jaccard index. The data matrix used to generate Figure 3.6 has 20 rows and 15 columns with two additive pattern biclusters without noise (see Figure 3.7 (b)). The mean value of the Jaccard index for these two biclusters over 10 runs is used to plot the graphs. For α = 1.7, the Jaccard index has the smallest variance. Figure 3.6 (b) shows the effect of parameter β on the Jaccard index. The highest mean value is achieved when β = 1. Figure 3.6 (c) shows the effect of different similarity threshold on the Jaccard index. A similarity threshold of τ = 0.5 gives the highest Jaccard index with the smallest variance. It can be observed from Figure 3.6 that the quality of the biclusters is not dramatically affected by the choice of parameter values if they are kept within a certain range.


Jaccard Index
1
0.8
0.6
0.4
0.2
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
α

(a)


Jaccard Index
1
0.8
0.6
0.4
0.2
0
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
β

(b)



(c)

Figure 3.6. Biclustering accuracy for different values of (a) α, (b) β, and (c) τ. Vertical lines are the standard deviation error bars.

3.3.2 Synthetic Data

Four synthetic datasets with different characteristics are considered in our experiment. The first synthetic data matrix, SD1, has 200 rows by 40 columns. The background matrix is generated by uniformly distributed random values between -5 and 5. Four biclusters are embedded in the data matrix with the following details. A constant row pattern bicluster of size 40 × 7; a constant column pattern bicluster of size 25 × 10; a constant column pattern bicluster of size 35 × 8 with three rows and three columns in common with the previous bicluster; a constant value pattern bicluster of size 40 × 8. Gaussian noise with standard deviation of 0.3 is added to the data matrix. The second data matrix, SD2, and the third data matrix, SD3, consider additive pattern and constant pattern, respectively. These data matrices consist of 20 rows and 15 columns with a noisy background generated by uniform distribution U(-5,5). There are two biclusters in the data matrices with sizes 10 × 6 and 11 × 6 and 1 row and

no column in common. The biclusters are degraded by a Gaussian noise with the variance set between zero and one. The last data matrix, SD4, consists of two constant biclusters with no noise and six rows in common. The sizes of biclusters are 10 × 6 and 11 × 6. Due to the overlapping rows between the two biclusters, SD4 actually has a third bicluster in it. Figure 3.7 shows the synthetic data matrices before noise is added. For the synthetic data matrices, δ is set to three in the experiments.

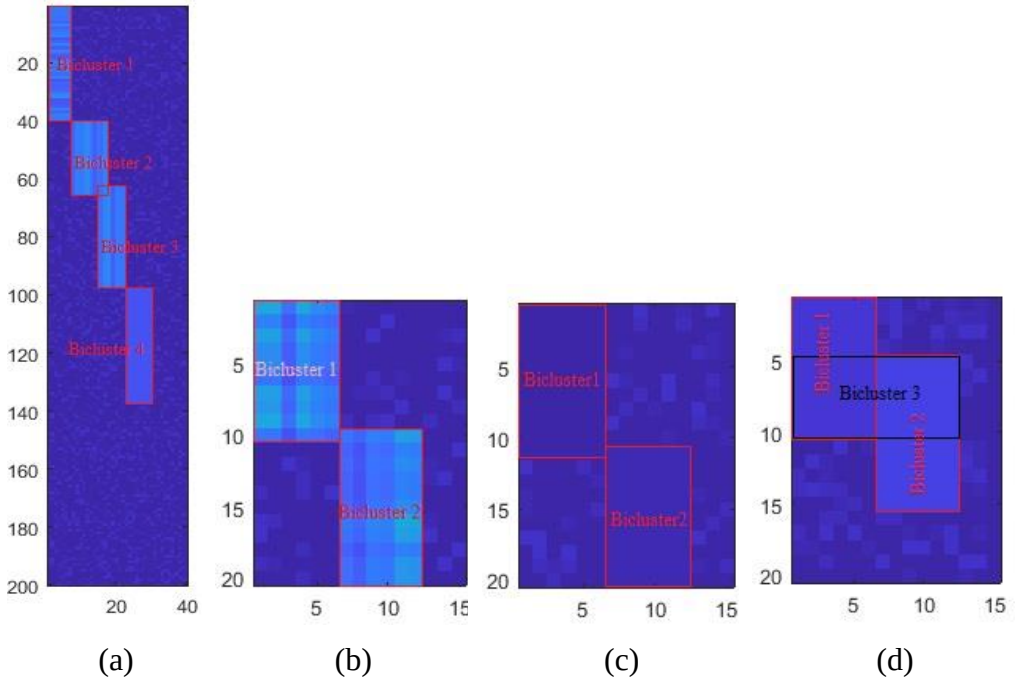


Figure 3.7. Visualisation of synthetic data matrices before noise is added. (a) SD1, (b) SD2, (c) SD3, (d) SD4

In order to validate and compare the results, the Jaccard index as defined in Equation (2.1) is used. The comparison results are shown in Figure 3.8 to Figure

3.11. We compared our method with BiMax (Prelić et al., 2006), FABIA (Hochreiter et al., 2010), ISA (Bergmann et al., 2003), CC (Cheng and Church, 2000), xMotif (Murali and Kasif, 2003), OPSM (Ben-Dor et al., 2003), and LAS (Shabalin et al., 2009).

The results of BiMax, CC, ISA, xMotif and OPSM are calculated using BicAT tool (Barkow et al., 2006). The LAS method is publicly available at https://genome.unc.edu/las/. FABIA method is also publicly available at http://www.bioinf.jku.at/software/fabia/fabia.html. Based on the discussion in (Pontes et al., 2013) we use the default parameters set as given in those methods.

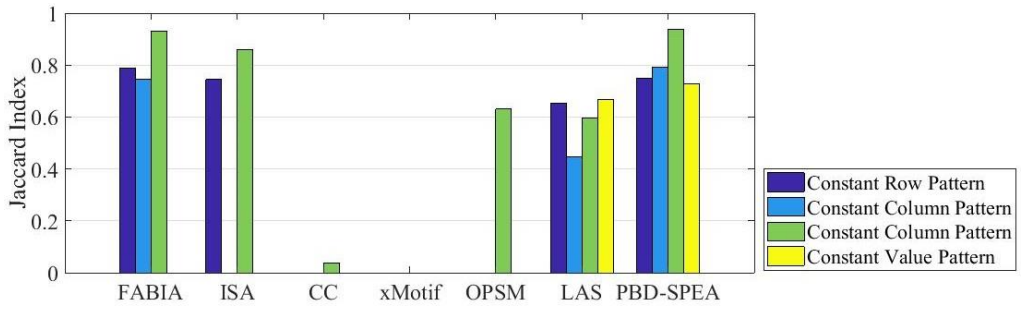


Figure 3.8. Biclustering accuracy in detecting different biclusters for SD1 data matrix

The biclustering accuracy in detecting different biclusters for SD1 data matrix is shown in Figure 3.8. As can be seen, LAS and PBD-SPEA are the two methods that are able to detect all four biclusters. PBD-SPEA has the higher accuracy than LAS in detecting all four biclusters. The xMotif method is not able to discover any bicluster because of the noise added to the data matrix. Interestingly, regardless of its high accuracy in detecting constant row and

constant column pattern biclusters, FABIA is not able to detect constant value pattern bicluster as the method is based on the outer product of two vectors.

From the results that are presented in Figure 3.9 and Figure 3.10, it is interesting to note that ISA could not detect any bicluster at all with the default parameter values and xMotif only detects biclusters when there is no noise in the data. Compare to other methods, the Jaccard index of BiMax remains stable when the noise level increases. It is clear that PBD-SPEA outperformed all the compared methods.

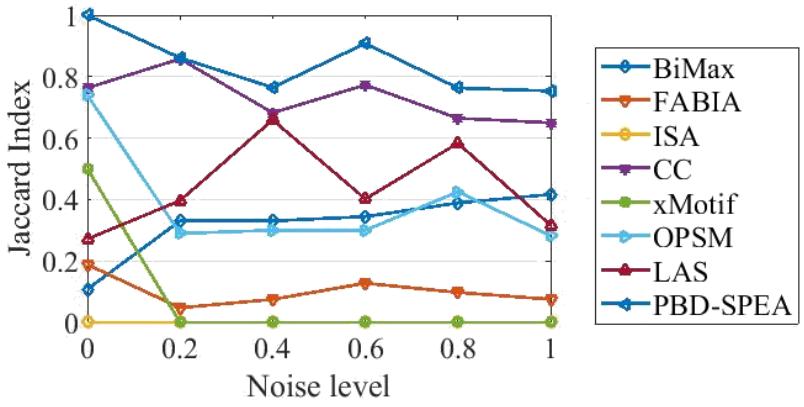


Figure 3.9. Biclustering accuracy against different noise level for SD2 data matrix

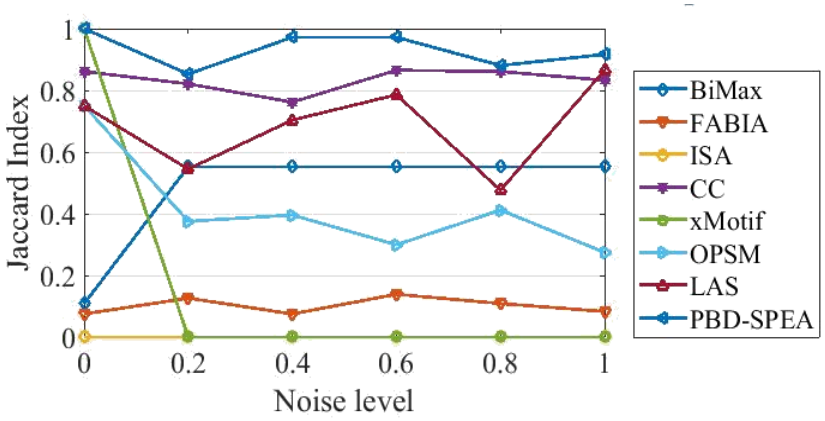


Figure 3.10. Biclustering accuracy against different noise level for SD3 data matrix

For the SD4 data matrix, our method is able to detect the two biclusters plus the overlapped part as a separate bicluster (Bicluster 3 in Figure 3.7 (d)). Note that technically the overlap part can be considered as a separate, third bicluster. xMotif is able to detect 100% of bicluster 1 and 2 but fails to detect the overlapped part. All other methods fail to either detect a bicluster completely or detect biclusters that are not perfect. The comparison results are shown in Figure 3.11.

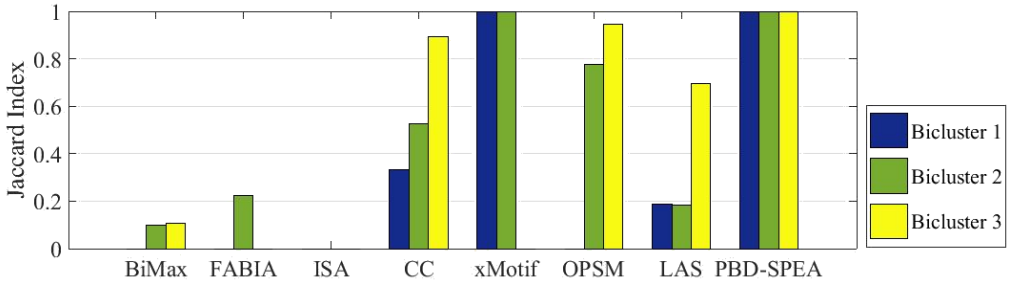


Figure 3.11. Biclustering accuracy in detecting different biclusters for SD4 data matrix

3.3.3 Gene Expression Data

In order to assess the biological relevance of the proposed method, we performed experiments on the yeast *Saccharomyces cerevisiae* gene expression data (Cho et al., 1998) containing 2884 genes and 17 conditions, and Human B-cell lymphoma data matrix (Alizadeh et al., 2000) that has 4026 genes and 96 conditions. Although there are many advanced methods to impute the missing values in the data matrices, in our experiments we simply replaced them with a randomly selected value within the corresponding conditions as in (Cheng and Church, 2000). The value of δ is set to 300 for the Yeast data matrix and 1200 for the Human Lymphoma data matrix as in (Cheng and Church, 2000).

Table 3.2 presents the comparison results of average MSR values and the average size of the detected biclusters in a number of methods. We run our method using the data matrices, and the results of other methods are taken from their original papers. The proposed method achieved smaller average MSR values compared to the other methods for both data matrices. Although our method did not detect the biggest biclusters among all the methods tested, our detected biclusters exhibit high biological coherence between genes during biological validation. As it is evident from the results, our method can detect large biclusters with small MSR value that are biologically meaningful.

Table 3.2. The comparison of biclusters of different methods for Yeast and human b-cell data matrices

Method	**Average MSR Value**		**Average Size Value**	
	Yeast	**Human B-Cell**	**Yeast**	**Human B-Cell**
PBD-SPEA	122.67	635.667	1023.65	1309.68
NSGA2B (Mitra and Banka, 2006)	234.87	987.5	10301.7	33463.74

Method	Average MSR Value		Average Size Value	
	Yeast	Human B-Cell	Yeast	Human B-Cell
MOPSOB (Liu et al., 2008)	218.54	34012.2	10510.8	927.4
HMOBI (Seridi et al., 2015)	299.6	1199.9	7665.59	47442.87
DMOPSOB (Liu and Chen, 2010)	216.13	905.23	11213.5	35442.98
MODPSFLB (Liu et al., 2012)	212.8	904.9	11220.7	35601.83
DMOIOB (Liu et al., 2011)	201.86	832.79	2841.08	7106.51
MOIB (Liu et al., 2009a)	202.32	839.74	2638.74	6918.29

In order to verify the functional enrichment of detected genes in biclusters for the Yeast data matrix, GENECODIS (Tabas Madrid et al., 2012, Nogales Cadenas et al., 2009, Carmona Saez et al., 2007) is used. Table 3.3 to Table 3.6 show the GO term and KEGG evaluation of one of the biclusters with the smallest p-value. This bicluster contains 11 genes and 17 conditions. The MSR value for this bicluster is 77.5262 while the size is 187. It is clear that the GO terms listed in the tables are significantly enriched compared to background occurrence frequency.

Table 3.3. Biological process ontology of GOTermFinder

GO term	Cluster frequency	Background frequency	P-value	FDR	Expected False Positives	Genes annotated to the term
cytoplasmic translation | AmiGO	6 out of 11 genes, 54.5%	171 out of 7163 background genes, 2.4%	3.83e-06	0.00%	0.00	RPL13A/YDL082W, RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, RPL25/YOL127W, RPL33B/YOR234C

GO term	Cluster frequency	Background frequency	P-value	FDR	Expected False Positives	Genes annotated to the term
ribosome assembly | AmiGO	4 out of 11 genes, 36.4%	57 out of 7163 background genes, 0.8%	6.16e-05	0.00%	0.00	RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, RPL25/YOL127W
ribosomal large subunit assembly | AmiGO	3 out of 11 genes, 27.3%	35 out of 7163 background genes, 0.5%	0.00093	0.00%	0.00	RPL6B/YLR448W, RPL6A/YML073C, RPL25/YOL127W
translation | AmiGO	7 out of 11 genes, 63.6%	709 out of 7163 background genes, 9.9%	0.00113	0.00%	0.00	RPL13A/YDL082W, RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, GIS2/YNL255C, RPL25/YOL127W, RPL33B/YOR234C
organelle assembly | AmiGO	4 out of 11 genes, 36.4%	122 out of 7163 background genes, 1.7%	0.00130	0.00%	0.02	RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, RPL25/YOL127W
ribonucleoprotein complex assembly | AmiGO	4 out of 11 genes, 36.4%	150 out of 7163 background genes, 2.1%	0.00294	0.01%	0.08	RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, RPL25/YOL127W
ribonucleoprotein complex subunit organization | AmiGO	4 out of 11 genes, 36.4%	160 out of 7163 background genes, 2.2%	0.00378	0.01%	0.08	RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, RPL25/YOL127W

Table 3.4. Molecular function ontology of GOTermFinder

Gene Ontology term	Cluster frequency	Background frequency	P-value	FDR	Expected False Positives	Genes annotated to the term
structural constituent of ribosome | AmiGO	6 out of 11 genes, 54.5%	224 out of 7163 background genes, 3.1%	2.48e-06	0.00%	0.00	RPL13A/YDL082W, RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, RPL25/YOL127W, RPL33B/YOR234C

Gene Ontology term	Cluster frequency	Background frequency	P-value	FDR	Expected False Positives	Genes annotated to the term
structural molecule activity | AmiGO	6 out of 11 genes, 54.5%	344 out of 7163 background genes, 4.8%	3.09e-05	0.00%	0.00	RPL13A/YDL082W, RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, RPL25/YOL127W, RPL33B/YOR234C

Table 3.5. Cellular component ontology of GOTermFinder

Gene Ontology term	Cluster frequency	Background frequency	P-value	FDR	Expected False Positives	Genes annotated to the term
cytosolic ribosome | AmiGO	6 out of 11 genes, 54.5%	175 out of 7163 background genes, 2.4%	2.03e-06	0.00%	0.00	RPL13A/YDL082W, RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, RPL25/YOL127W, RPL33B/YOR234C
cytosolic large ribosomal subunit | AmiGO	5 out of 11 genes, 45.5%	94 out of 7163 background genes, 1.3%	3.79e-06	0.00%	0.00	RPL13A/YDL082W, RPL6B/YLR448W, RPL6A/YML073C, RPL25/YOL127W, RPL33B/YOR234C
ribosome | AmiGO	7 out of 11 genes, 63.6%	352 out of 7163 background genes, 4.9%	4.53e-06	0.00%	0.00	RPL13A/YDL082W, RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, GIS2/YNL255C, RPL25/YOL127W, RPL33B/YOR234C
cytosolic part | AmiGO	6 out of 11 genes, 54.5%	232 out of 7163 background genes, 3.2%	1.09e-05	0.00%	0.00	RPL13A/YDL082W, RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, RPL25/YOL127W, RPL33B/YOR234C
ribosomal subunit | AmiGO	6 out of 11 genes, 54.5%	241 out of 7163	1.36e-05	0.00%	0.00	RPL13A/YDL082W, RPS0A/YGR214W, RPL6B/YLR448W,

Gene Ontology term	Cluster frequency	Background frequency	P-value	FDR	Expected False Positives	Genes annotated to the term
		background genes, 3.4%				RPL6A/YML073C, RPL25/YOL127W, RPL33B/YOR234C
large ribosomal subunit | AmiGO	5 out of 11 genes, 45.5%	141 out of 7163 background genes, 2.0%	2.89e-05	0.00%	0.00	RPL13A/YDL082W, RPL6B/YLR448W, RPL6A/YML073C, RPL25/YOL127W, RPL33B/YOR234C
ribonucleoprotein complex | AmiGO	8 out of 11 genes, 72.7%	766 out of 7163 background genes, 10.7%	5.07e-05	0.00%	0.00	RPL13A/YDL082W, RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, NOP13/YNL175C, GIS2/YNL255C, RPL25/YOL127W, RPL33B/YOR234C
non-membrane-bounded organelle | AmiGO	8 out of 11 genes, 72.7%	1308 out of 7163 background genes, 18.3%	0.00296	0.00%	0.00	RPL13A/YDL082W, RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, NOP13/YNL175C, GIS2/YNL255C, RPL25/YOL127W, RPL33B/YOR234C
intracellular non-membrane-bounded organelle | AmiGO	8 out of 11 genes, 72.7%	1308 out of 7163 background genes, 18.3%	0.00296	0.00%	0.00	RPL13A/YDL082W, RPS0A/YGR214W, RPL6B/YLR448W, RPL6A/YML073C, NOP13/YNL175C, GIS2/YNL255C, RPL25/YOL127W, RPL33B/YOR234C

In these tables, the number of genes with the annotated GO terms in the detected bicluster and in the data matrix is shown as cluster frequency and background frequency, respectively. False discovery rate (FDR) is the ratio of the average number of genes that have a p-value as good as the real gene's p-

value by the number of genes in the real data that have a p-value as good as that p-value.

Table 3.6 shows the KEGG pathway analysis results of the same bicluster. In this table, NGR represents the number of annotated genes in the reference list and TNGR represents the total number of genes in the reference list with the KEGG item terms. The input list includes all genes in the bicluster that are annotated with the item terms for the selected category and organism. NG stands for the number of annotated genes in the input list and TNG is the total number of genes in the input list. Hyp is the hypergeometric p-value and Hyp* is the corrected hypergeometric p-value. As can be seen, most of the KEGG item terms are significantly enriched in the bicluster.

Table 3.6. Singular enrichment analysis of KEGG pathway

Genes	**NGR (TNGR)**	**NG (TNG)**	**Hyp (Hyp*)**	**Items**	**Items Details**
YDR156W, YER099C, YDR305C, YNL113W, YDL150W, YOR341W	94 (7109)	6 (85)	0.000859698 (0.0017194)	KEGG:00230	Purine metabolism
YDR156W, YNL113W, YDL150W, YOR341W	30 (7109)	4 (85)	0.000411578 (0.00164631)	KEGG:00230, KEGG:00240, KEGG:03020	Purine metabolism, Pyrimidine metabolism, RNA polymerase
YOR056C, YLR186W, YNR053C, YPL204W	91 (7109)	4 (85)	0.0231683 (0.0231683)	KEGG:03008	Ribosome biogenesis in eukaryotes
YDR156W, YNL113W, YDL150W,	70 (7109)	5 (85)	0.00142817 (0.00190423)	KEGG:00240	Pyrimidine metabolism

Genes	NGR (TNGR)	NG (TNG)	Hyp (Hyp*)	Items	Items Details
YEL021W, YOR341W					

3.3.4 Image Data

We apply PBD-SPEA to cluster the 15-scene categories dataset (Fei-Fei and Perona, 2005, Oliva and Torralba, 2001, Lazebnik et al., 2006).

In order to apply PBD-SPEA, we generate a feature vector that describes each image using the bag-of-words (BoW) model. Feature detection, feature description and codebook generation are the three steps in the BoW model (Fei-Fei and Perona, 2005). The features in the BoW model are keywords that characterise the image for each category. For example, for the kitchen category, appliances, cup, cupboard, drawer, plate set, cutlery, pots and pans, kitchen bench, dining table, and chair are the set of keywords in this category. For the living room category, sofa, armchair, chair, cushion, lamp, coffee table, side table, rug, ceiling fan, fireplace, photo frame and curtain are the keywords. We use 10 images from the first 10 categories and 5 individual images for each keyword from Google search images. Then, a sparse binary vector is used as feature descriptor to represent each image.

We apply PBD-SPEA to group the images into meaningful clusters. The number of biclusters is set to two, and Figure 3.12 shows the two detected biclusters. The first detected bicluster (Bicluster1) includes images with features such as cupboard, appliances, pots and pans, dining table and chair, plate set and cutlery, and cup. The second bicluster (Bicluster2) includes images with features such as sofa, coffee table, cup, plate set and cutlery, dining table and chair. Interestingly, there is an overlap between the detected images in the two

biclusters (cup, plate set and cutlery, dining table and chair). From visual inspection, we can see that Bicluster1 images correspond to the kitchen category while Bicluster2 points to the living room category.

This experiment shows that biclustering can be used to uncover higher-level semantic information within images. Compare to the results reported in the paper of (Fei-Fei and Perona, 2005), our results show more comprehensibility. Here, we can see why images appeared in the same bicluster while the relationship of the 13 category in Figure 9 in (Fei-Fei and Perona, 2005) is not clear. PBD-SPEA is able to uncover higher-level concepts (i.e. kitchen, living room) by recognizing a group of features that clusters a group of images together.

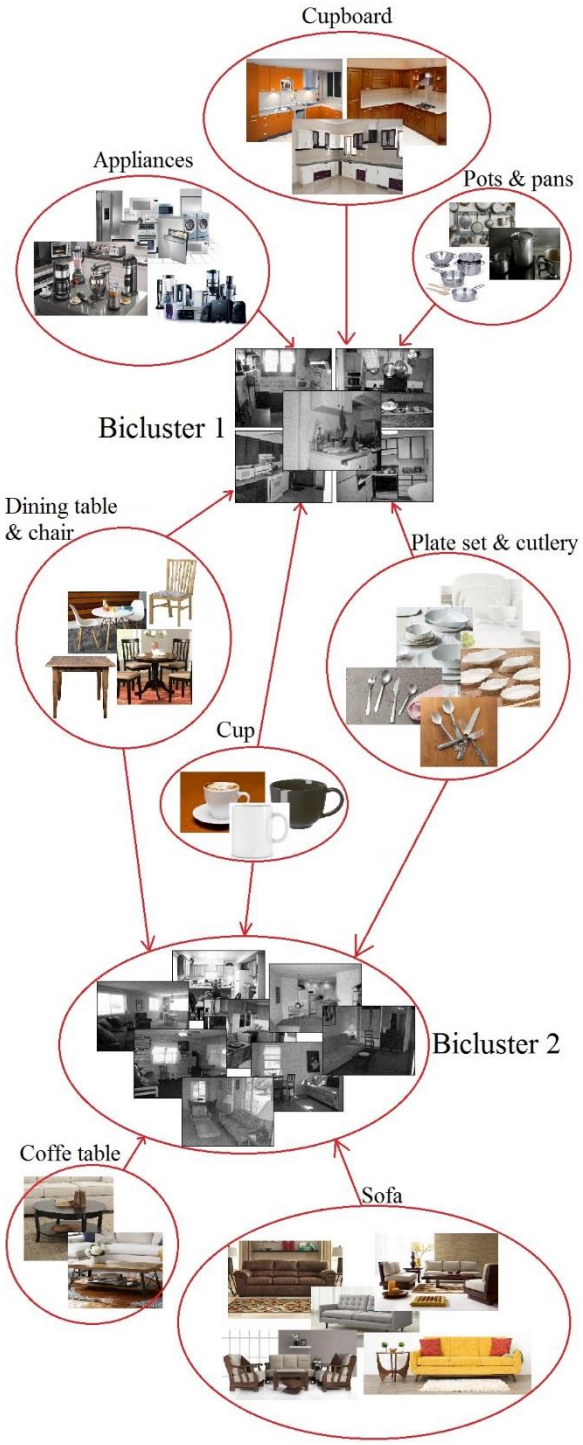


Figure 3.12. Detected biclusters group images with similar concepts

3.3.5 Facebook Data

We apply PBD-SPEA to the Social Circles Facebook dataset (Leskovec and Mcauley, 2012), which consists of circles (friends' lists) from Facebook. This dataset includes 4039 nodes and 88234 edges and is publicly available at https://snap.stanford.edu/data/egonets-Facebook.html. In this dataset, there are 10 networks where each user is represented by a set of features including birthday, education, first name, last name, gender, hometown, languages, location, work, and locale. We run PBD-SPEA for each network separately.

In Table 3.7, we report the number of IDs and features in a network, and the number of IDs and features in the detected bicluster. We also summarise the mean value of the pairwise cosine distance of the detected bicluster μ_{cd}; the mean value of the pairwise cosine distance of the samples μ; the standard deviation of the pairwise cosine distance of the samples σ; and the common detected features.

From the results in Table 3.7, we can conclude that the friend circles are mostly formed by the common educational activities, work place and/or biological relationships. These biclusters are smaller and more coherent in comparison to the original network (in network #2, the detected bicluster is almost 84% smaller than the original network). Furthermore, the biclusters can be used for tagging groups of interest and as an input data for recommendation system, search relevant, user profiling (Mislove et al., 2010), and targeted marketing (Bolotaeva and Cata, 2011) applications.

For example, in network #2, the detected bicluster groups users with the same educational background and work history. Most probably, this group of users go through educational events such as graduation ceremony at the same time and they are much more likely to have similar needs. It is easier to create a promotion

post or an advertisement to target these core users rather than targeting the whole network with diverse needs.

In order to disclose the correlation of our results that correspond to the users' interests, we generate 10,000 submatrices from each network by random sampling of rows and columns based on the number of rows and columns in the detected bicluster. For each sampled submatrix, we calculate the mean value of the pairwise cosine distance μ_{cd} and plot their histogram in Figure 3.13.

In Figure 3.13, the x-axis refers to the mean values of pairwise cosine distance, where $1 \in [0\ 0.1)$; $2 \in [0.1\ 0.2)$; $3 \in [0.2\ 0.3)$; $4 \in [0.3\ 0.4)$; $5 \in [0.4\ 0.5)$; $6 \in [0.5\ 0.6)$; $7 \in [0.6\ 0.7)$; $8 \in [0.7\ 0.8)$; $9 \in [0.8\ 0.9)$; $10 \in [0.9\ 1)$. In these figures, the asterisk on the x-axis shows where μ_{cd} is located. These empirical distributions provide the baseline statistical distributions for us to assess the significant of the detected biclusters. We calculated the mean value μ and standard deviation σ of these distributions.

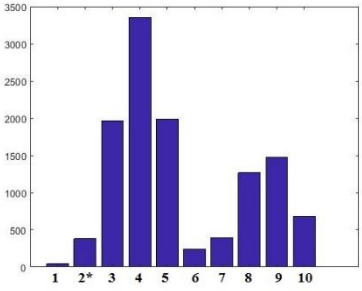
The μ_{cd} of the detected bicluster are mostly smaller than $\mu - 3\sigma$. Based on the Chebyshev's inequality for general probability distribution only 5.5% of the values are smaller than $\mu - 3\sigma$. Please note that this is a much weaker bound than the bound obtained under normality assumption. This suggests that random selection of a submatrix with a similar μ_{cd} to the detected bicluster is unlikely and the probability of obtaining the detected bicluster by chance is very low. Therefore, PBD-SPEA is able to detect biclusters that show significant semantic enrichment.

We also compare our method with the method we proposed in Chapter 4. The results are summarised in Table 4.11. Based on the reported results, LBDP

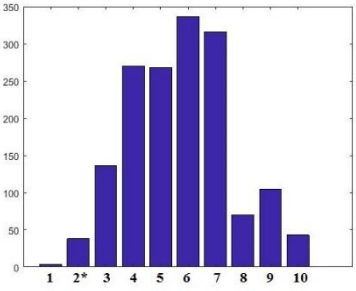
(Golchin and Liew, 2018) outperforms PBD-SPEA by having smaller mean values in their detected biclusters.

Table 3.7. Biclustering results on 10 different Facebook network

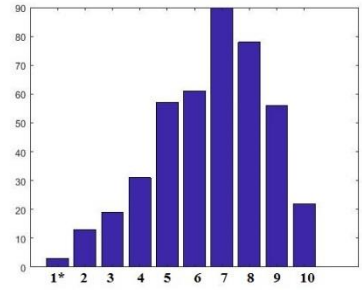
Net. NO.	Network		Detected bicluster		μ_{cd} detected bicluster	μ	σ	Common detected features
	# IDs	# features	# IDs	# features				
1	348	224	180	30	0.2267	0.45	0.11	Education, last name, work
2	1046	576	496	200	0.3577	0.66	0.05	Education, work
3	228	161	91	20	0.0986	0.51	0.05	Education, work, birthday
4	160	105	78	10	0.0986	0.64	0.05	Education, work, birthday
5	171	63	63	29	0.2574	0.63	0.03	Education, last name
6	67	48	24	8	0.1622	0.55	0.04	Education, work
7	793	319	416	97	0.1984	0.49	0.04	Education, hometown, last name, work
8	756	480	449	106	0.2043	0.42	0.08	Education, work, birthday
9	548	262	221	26	0.0893	0.38	0.03	Education, last name, work
10	60	42	20	10	0.1165	0.58	0.06	Education, work


3500
3000
2500
2000
1500
1000
500
0
1 2* 3 4 5 6 7 8 9 10

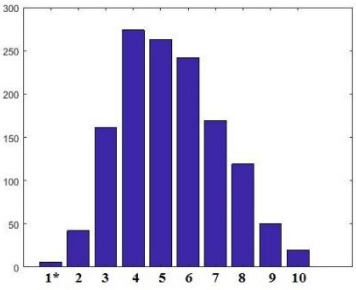
(a)


350
300
250
200
150
100
50
0
1 2* 3 4 5 6 7 8 9 10

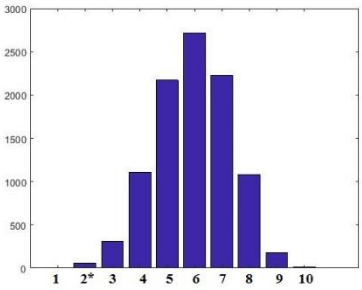
(b)


90
80
70
60
50
40
30
20
10
0
1* 2 3 4 5 6 7 8 9 10

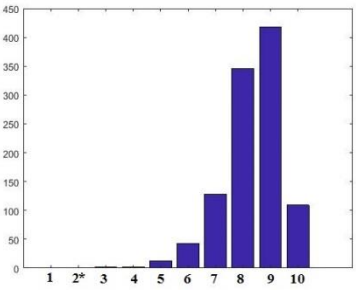
(c)


300
250
200
150
100
50
0
1* 2 3 4 5 6 7 8 9 10

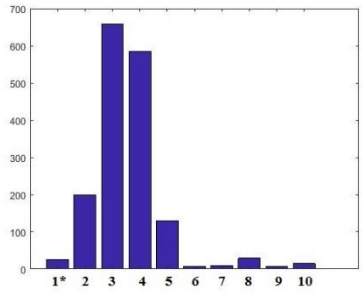
(d)


3000
2500
2000
1500
1000
500
0
1 2* 3 4 5 6 7 8 9 10

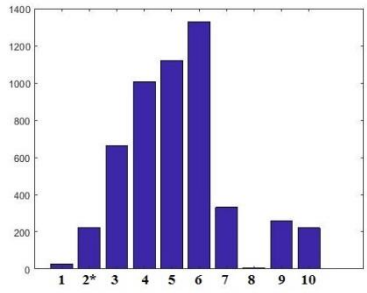
(e)


450
400
350
300
250
200
150
100
50
0
1 2* 3 4 5 6 7 8 9 10

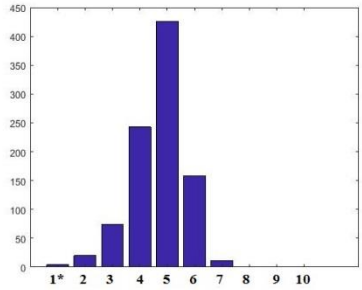
(f)



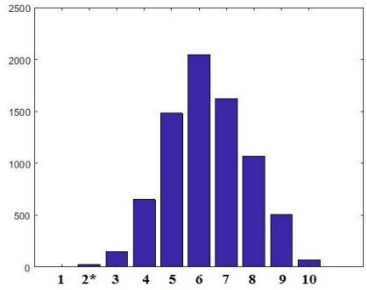
(g)



(h)



(i)



(j)

Figure 3.13. The histogram of the mean values of pairwise cosine distance for randomly generated biclusters (a) network ID 1, (b) network ID 2, (c) network ID 3, (d) network ID 4, (e) network ID 5, (f) network ID 6, (g) network ID 7, (h) network ID 8, (i) network ID 9, (j) network ID 10

3.4 Conclusion

In this Chapter, we proposed an efficient evolutionary algorithm to find the biclusters in a data matrix. To do this, a multi-objective evolutionary algorithm called PBD-SPEA is proposed to detect multiple biclusters concurrently from a

data matrix. The multi-objective search strategy is used to handle conflicting objectives in the cost function. PBD-SPEA is able to detect multiple biclusters concurrently through exploration and exploitation of the solution space, and returns a set of solution in the Pareto front. A post-processing step is also proposed to select the set of final biclusters among Pareto front individuals.

We can obtain three main conclusions based on our experiments as follows. First, our method achieves promising results for all experiments in this study for both synthetic and real gene expression data. Second, our method uncovered higher-level semantic information among images by recognizing a group of features that clusters a group of images. Finally, we justified the detected biclusters in this study are rarely achieved by chance in a social media dataset.

Despite achieving good results compared to (Barkow et al., 2006, Hochreiter et al., 2010, Bergmann et al., 2003, Cheng and Church, 2000, Murali and Kasif, 2003, Ben-Dor et al., 2003) our method is not able to detect all types of bicluster patterns. We plan to achieve this goal by introducing multi-objective evolutionary geometrical biclustering. This will be described in the next chapter.

4

Geometric Biclustering based on Multi-objective Evolutionary Algorithm

In previous chapter, we have shown that by proposing a multi-objective evolutionary algorithm and combining the encoding of multiple biclusters in an individual, we are able to detect multiple biclusters concurrently. Furthermore, using the right objective values significantly improve the accuracy of the biclustering result. We have also shown the effectiveness of our method to deal with overlapped biclusters. However, we were not able to discover biclusters having linear pattern. In order to achieve this goal, we need to solve the problem of biclustering from the geometrical point of view. This chapter is based on our submitted work (Golchin and Liew, 2018).

In this chapter, we consider bicluster as rows that appear in the column space as hyperplane. One of the challenges in the proposed method, denoted as LBDP (Linear Bicluster Detection by Projection), is to estimate the parameters of the hyperplane. The dimension of the hyperplane is hypothesized during the evolutionary search of the bicluster. The normal vector of the hyperplane is determined by singular value decomposition (SVD). We also introduced two novel objective values to keep the diversity among the objective space and to converge into several local and/or global optima. In our new proposed method, we are able to detect different patterns in biclusters without pre-processing the data matrices. We will show the effectiveness and reliability of our proposed geometrical method on both synthetic and real datasets.

4.1 Introduction

As it was highlight in Chapter 2, we have shown that using the right merit function and evolutionary algorithm can improve the quality of the biclustering solution. However, using evolutionary algorithm, only one bicluster is detected at a time. Furthermore, by using sequential detection, most methods replace the detected bicluster with the background noise, which avoids the detection of overlapped biclusters. In Chapter 3, we addressed this problem by introducing a new encoding scheme for individuals that encode all biclusters in an individual.

In this Chapter, we try to find linear pattern biclusters through geometric biclustering methods and hyperplane detection. (Gan et al., 2005) introduced geometric biclustering in 2005. The idea is based on identifying linear patterns in biclusters. In a high dimensional space, rows in biclusters can be considered as points distributed on a geometric structure. In order to identify this geometric structure, Hough transform (HT) is used in the literature (Gan et al., 2005, Liu et al., 2014, Wang and Yan, 2010, Wang and Yan, 2013, Wang et al., 2012, Zhao et al., 2008). However, the memory and computation cost of HT in high dimensional space is high. In order to overcome this issue, (Wang and Yan, 2010, Wang and Yan, 2013, Wang et al., 2012, Zhao et al., 2008) attempt to apply HT in column-pair spaces. The challenge now is how to combine the sub-biclusters to generate the final biclusters. Nevertheless, such heuristic combination may converge into a local solution.

In this chapter, in order to solve the problem of geometrical biclustering and hyperplane detection, we use evolutionary algorithms and singular value decomposition (SVD). The proposed method, LBDP, reduces the space usage and complexity of geometric biclustering by utilizing the power of EA and SVD. LBDP can be applied without pre-processing the data matrix and is able to

converge into several local or global optima concurrently to find several biclusters with different patterns. Using the encoding in EA, LBDP does not need to estimate the dimension of the hyperplane, which makes the parameter estimation of the hyperplane much easier. These changes lead to improved performance of the proposed method.

4.2 The Proposed Method

The main idea of our proposed method is to project rows into the column space to form a hyperplane. One of the challenges in LBDP is to estimate the parameters of the hyperplane. The dimension of the hyperplane is identified during the evolutionary search of the solution. The normal vector of the hyperplane is determined by SVD. We also introduced two novel objective values to keep the diversity among the objective space and to converge into several local and/or global optima. In addition, to select the final bicluster from the Pareto front individuals we applied the k-means algorithm. One of the advantages of LBDP compared to the PBD-SPEA that proposed in Chapter 3 is that LBDP is able to detect different patterns in biclusters without pre-processing the data matrix.

4.2.1 Initial Population Generation

In LBDP, each bicluster is encoded in an individual as a fixed size $M+N$ binary string as in Figure 4.1. The first M bits represent rows and the remaining N bits represent the columns where one indicates a row that belongs to the bicluster and zero otherwise. In Figure 4.1, rows {2, 3, 7, 8, 9} and columns {2, 3} from the data matrix belong to the bicluster. In order to generate the initial population, random rows and columns are assigned to each bicluster.

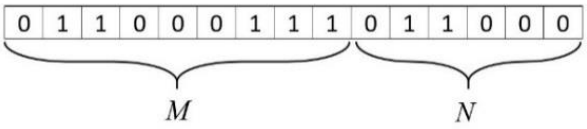


Figure 4.1. The individual representation

4.2.2 Local Search

Due to the nature of evolutionary algorithms, the probability of having unwanted rows and columns in a bicluster is high. In order to improve the quality of the generated biclusters, a local search removes the uncorrelated rows and columns and adds correlated rows and columns to the hyperplane. In order to remove rows or columns, a row or column is first selected randomly. Then, the Pearson correlation coefficient (PCC) r_{xy} based on Equation (4.1) is used to calculate the correlation between the selected row or column and the remaining rows or columns in the bicluster. If r_{xy} is smaller than a user-defined threshold in 75% of rows or columns, the selected row or column is removed from the bicluster.

On the other hand, in order to add rows or columns, a random row or column is selected from the rows or columns in the data matrix excluding the bicluster. PCC calculates the correlation between the selected row or column and the rows or columns in the bicluster. If r_{xy} is greater than a user-defined threshold in 75% of the rows or columns, then the selected row or column is added to the bicluster. Algorithm 4.1 summarizes the steps of the local search. r_{xy} is defined by

$$r_{xy} = \frac{n\sum x_i y_i - n\bar{x}\bar{y}}{\sqrt{(\sum x_i^2 - n\bar{x}^2)}\sqrt{(\sum y_i^2 - n\bar{y}^2)}} \quad \textbf{(4.1)}$$

where $x = \{x_1, x_2, \ldots, x_n\}$ and $y = \{y_1, y_2, \ldots, y_n\}$ are the two rows or columns in a bicluster, $\bar{x} = 1/n(\sum_{i=1}^{n} x_i)$ and $\bar{y} = 1/n(\sum_{i=1}^{n} y_i)$ are the mean values of

those rows or columns, x_i and y_i are the expression values of the i^{th} row or column. The value of r_{xy} is between -1 to 1 where -1 is total negative correlation, 1 is total positive correlation and 0 is no correlation.

Algorithm 4.1: Local search

Input:	A random individual that is selected for mutation
Output:	More coherent individual
Repeat for all rows in a bicluster // Rows deletion	
1	Select a row r_i randomly
2	Calculate r_{xy} between r_i and the remaining rows in the bicluster using Equation (4.1)
3	If $r_{xy} \leq \alpha_{rr}$ for more than 75% of the remaining rows in the bicluster then remove r_i
Repeat for all columns in a bicluster // Columns deletion	
4	Select a column c_j randomly
5	Calculate r_{xy} between c_j and the remaining columns in the bicluster using Equation (4.1)
6	If $r_{xy} \leq \alpha_{rc}$ for more than 75% of the remaining columns in the bicluster then remove c_j
Repeat for selected rows in the data matrix S // Rows addition	
7	Select a row $r_i \in S$ randomly
8	Calculate r_{xy} between r_i and the rows in the bicluster using Equation (4.1)
9	If $r_{xy} \geq \alpha_{ar}$ for more than 75% of the rows then add r_i
Repeat for selected columns in the data matrix F // columns addition	
10	Select a column $c_j \in F$ randomly
11	Calculate r_{xy} between c_j and the columns in the bicluster using Equation (4.1)
12	If $r_{xy} \geq \alpha_{ac}$ for more than 75% of the columns then add c_j

In this algorithm α_{rr}, α_{rc}, α_{ar}, and α_{ac} are user-defined thresholds to control the rate of rows deletion, columns deletion, rows addition, and columns addition, respectively. We will study the effects of parameter changes in the accuracy of LBDP in Section 4.3.1. We also discuss the effects of the number of rows and columns to be added or removed from a bicluster and why it is set at 75% in Section 4.3.1. Based on that, the value of the parameters are set as follows: α_{rr} = 0.85, α_{rc} = 0.85, α_{ar} = 0.9 and α_{ac} = 0.95.

4.2.3 Fitness Function

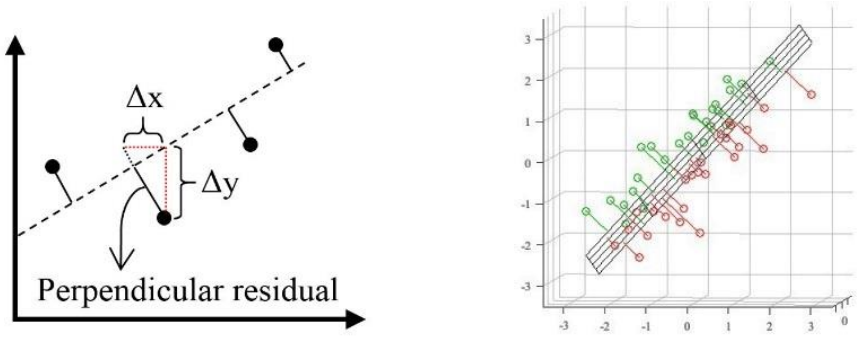
The objectives in our proposed method include the coherence of the bicluster that is calculated by the root mean square errors (e_r) which we try to minimize, and the size of the bicluster (S_b) which we try to maximize. These two objectives are in conflict since maximizing the size would increase the incoherence of the bicluster. In addition, in order to increase the diversity of the population and group the population into a number of biclusters, we introduce two additional novel objectives: (i) the distance between the individuals in each group (Ind_J), which we try to minimize, and (ii) the distance between individuals of each group to another group ($Outd_J$), which we try to maximize. Equation (4.2) defines the objective function $ObjVal$. The better the objective values are, the higher the quality of the detected bicluster.

$$ObjVal = \begin{cases} e_r \\ S_b \\ Ind_J \\ Outd_J \end{cases} \quad \textbf{(4.2)}$$

4.2.3.1 Root Mean Square Errors

As we consider each row in the data matrix as a data point in a high dimensional space, where each column defines a dimension and each element specifies a coordinate value, the linear pattern among the elements of the data matrix makes a hyperplane in the high dimensional space. In LBDP, we try to fit data to a hyperplane. The problem of finding a hyperplane is now formulated as a regression problem. In general, if we have n predictor variables, then $y = c_0 + c_1x_1 + \ldots + c_nx_n + \varepsilon$ is our response variable. We assume that y is normal with mean $E[y|x_1, \ldots, x_n] = c_0 + c_1x_1 + \ldots + c_nx_n$ and standard deviation σ. We need to estimate the values of $c_0, c_1, \ldots, c_n$, and σ. In this model y is the random outcome of the dependent variables (observed value), c_0 is the regression constant ($E(y|x_1 = \ldots = x_n = 0)$), c_is are partial regression coefficient for variable x_i, and ε is the random error term, which we assume $\varepsilon \sim N(0,\sigma)$.

The best fit to a given set of points minimizes the sum of the squares of the residuals of the points to the hyperplane. Residuals or offsets are the difference between the observed value and the fitted value, which can be calculated as the perpendicular residuals as in Figure 4.2.



(a) (b)

Figure 4.2. Least square problem; (a) Fitting 2 dimension points to a line, (b) Fitting 3 dimension points to a plane

LBDP tries to find a hyperplane that is as close as possible to the set of rows in the bicluster. The summation of the orthogonal distances between points to the hyperplane indicates the quality of the hyperplane. Let h denote a point on the hyperplane and n_v denotes the normal vector of the hyperplane, then the orthogonal distance between a point p_i and the hyperplane is given by $(p_i - h)^T n_v$. Thus, Equation (4.3) solves the problem of finding a hyperplane.

$$\min_{h,\|n_v\|=1} \sum_{i=1}^{n} ((p_i - h)^T n_v)^2 \qquad \textbf{(4.3)}$$

Introducing a new $m \times n$ matrix $A = [p_1\text{-}h, p_2\text{-}h, \ldots, p_n\text{-}h]$, we can rewrite Equation (4.3) as

$$\min_{\|n_v\|=1} \|A^T n_v\|_2^2$$

The SVD of A is used ($A = U\,S\,V^T$) to replace A in the equation, where $U \in \mathbb{R}^{m\times m}$ is the left singular matrix satisfies $U^TU = I$; $V \in \mathbb{R}^{n\times n}$ is the right singular matrix satisfies $V^TV = I$; and $S \in \mathbb{R}^{m\times n}$ is the diagonal matrix of singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$, $r = \min(m, n)$ such that $AV = US$. Then we have

$$\|A^T n_v\|_2^2 = \|VS^TU^T n_v\|_2^2 = \|S^TU^T n_v\|_2^2$$
$$= (\sigma_1 y_1)^2 + (\sigma_2 y_2)^2 + \cdots + (\sigma_r y_r)^2$$

where y is the vector $y = U^T n_v$. Thus, $\|A^T n_v\|_2^2$ is minimized for $y = (0, \ldots, 0, 1, 0, \ldots, 0)^T$ where 1 is at the r^{th} position of vector y. r shows the rank of the matrix A and the rank of the matrix is the number of non-zero elements of a singular value matrix. Please refer to (Gander and Hrebicek, 2011) for more details. Figure 4.3 shows the visualization of the above statement. In this figure, a 3 × 2 matrix of rank two maps a circle on the hyperplane to an eclipse in the singular space. The main finding resulted by SVD of a matrix is that "*an m × n matrix M of rank r maps the r-dimensional unit hypersphere in row space of M into an r-dimensional hyper-ellipse in the range of M"* (Tomasi, 2013). According to Tomasi (Tomasi, 2013) the r^{th} column of the left singular vector U represents the normal vector of the hyperplane (n_v) (Söderkvist, 1993, Arun et al., 1987). The worst time complexity of the SVD for an $m \times n$ matrix is $O(min\{mn^2, m^2n\})$ (Holmes et al., 2007).

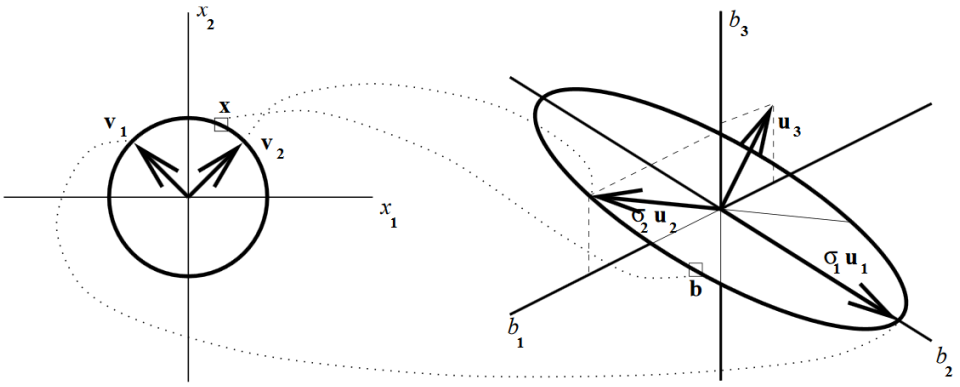


Figure 4.3. Visualisation of a 3×2 matrix transformation of rank two, using SVD (Tomasi, 2013)

The coherence of a bicluster is calculated by how close the rows are to the detected hyperplane. To find this value, the root mean square error (RMSE) of the rows are calculated by Equation (4.4).

$$e_r = \sqrt{(\sum_{i=1}^{m} \frac{\langle n_v, p_i \rangle + c}{\|n_v\|})/m} \quad \textbf{(4.4)}$$

Here, n_v is the normal vector of the hyperplane, c is the constant value of the hyperplane, p_i represents each row in the bicluster, $\|\bullet\|$ denotes the norm value of a vector and $\langle\bullet\rangle$ denotes the inner product of two vectors.

SVD should be applied on the normalized data especially when the variance of data is very different (Tomasi, 2013). Subtracting the mean value from the data centralizes the data such that the mean of the new data becomes zero. Algorithm 4.2 summarizes the steps of hyperplane detection. Let $\bar{e_j}$ denotes the mean value of each column in the bicluster. Matrix A is an $n \times m$ normalized matrix of the bicluster. SVD(A) computes the singular value decomposition of

the normalized matrix *A*. rank(*S*) calculates the number of non-zero elements in the singular value matrix *S*.

Algorithm 4.2: Hyperplane detection

1 $\bar{e_j} = \frac{1}{m}\sum_{j=1}^{m} e_{ij}$

2 $A = [e_{i1} - \bar{e_j}, \ldots, e_{im} - \bar{e_j}],\ i = \{1, \ldots, n\}$

3 $[U, S, V]$ = SVD(A)

4 r = rank(S)

5 $n_v = U(:,r)$

4.2.3.2 Size of a Bicluster

Equation (4.5) calculates the size of the bicluster where *m* and *n* are the numbers of rows and columns in a bicluster, respectively. w_g and w_c are the weights for the number of rows and columns in the data matrix to balance the number of selected rows and columns. The value of w_g is set to one, and w_c varies from one to ten and is the ratio of the number of rows and the number of columns in the data matrix. S_b defines the number of detected elements in the bicluster.

$$S_b = w_g \times M/m + w_c \times N/n \quad \textbf{(4.5)}$$

4.2.3.3 Diversity of the Population

Generally, EAs would converge to a single optimal solution in the search space. However, in this research we aim to converge into several local or global optima to be able to detect multiple biclusters concurrently. Figure 4.4 visualizes a multimodal function with more than one local or global optima. In order to identify several optima, niching method (Horn et al., 1994) maintain population

diversity by forming sub-populations in the neighbourhood of the local or global optima solutions. This is all possible by choosing the right objective function to reward individuals that exploit less dense areas of the search space. This causes population diversity and maintains individuals at local or global optima. We use the same idea in LBDP. The main population is always divided into several subgroups so that multiple optimal solutions related to different biclusters can be found concurrently. Note that the number of subgroups is fixed and it is equal to the number of biclusters in the data matrix, whereas the number of individuals in the subgroups is dynamic such that individuals can move from one subgroup to another.

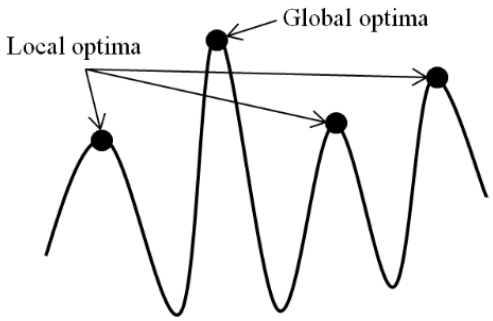


Figure 4.4. Local and global optima solutions in a multimodal function

In order to maintain the diversity of the population and to allow for searching for multiple biclusters concurrently we introduce a new objective based on the Jaccard distance between the individuals. In our method, a binary encoding scheme is used to represent each bicluster (i.e. each individual). The Jaccard distance between two individuals is the number of nonzero bits at which the bits are different. Equation (4.6) calculates the Jaccard distance between two individuals.

$$Ind_J = (\sum_{i=1}^{ind} \frac{\sum P_i \oplus B}{\sum P_i \vee B})/ind \quad \textbf{(4.6)}$$

where $\oplus$ is the logical exclusive OR (XOR) that outputs 1 when bits are different and $\vee$ is the logical OR that outputs 1 when one of the bits is 1. P_i are the biclusters in the population excluding the subgroup that B belongs. B is the newly generated bicluster, and we want to calculate the distance between B and the rest of the population. *ind* is the number of individuals in the population excluding B. The goal is to try to maximize Ind_J.

4.2.3.4 The Proximity of each Group

The Ind_J keeps individuals apart from each other and divides the population into subgroups in order to find multiple biclusters concurrently. In order to converge each group to the local optima, the proximity of each group members is considered. To do so, Equation (4.7) is used. A constant value of 2 is added to make sure that $Outd_J$ is bigger than zero and smaller than 1.

$$Outd_J = 1/(Ind_J + 2) \quad \textbf{(4.7)}$$

4.2.4 Final Bicluster Selection

The k-means algorithm selects the final bicluster among the Pareto front individuals from each subgroup. However, finding the optimal number of clusters *k* is a challenge. In this study, silhouette plot is used to determine the number of clusters as in (Chaudhari et al., 2010). Silhouette width for *k* values between 3 and 10 is calculated. The highest average silhouette width would indicate the optimal number of clusters. The silhouette width indicates the dependency of a solution to its own cluster (De Amorim and Hennig, 2015). The

higher the silhoutte width is, the stronger the probability of the solution belongs to its own cluster and less to its neighboring clusters. The best bicluster is the closest bicluster to the centroid of the cluster.

Figure 4.5 shows the Pareto front individuals when RMSE and the size of the bicluster are chosen as the objective values. The k-means algorithm is used to group the individuals and the number of k is set to three in this figure. Among the three regions, region 3 is selected to obtain the final individual since it provides a nice compromise of both objectives among the Pareto front. Note that based on Equation (4.5), the larger m and n are, the smaller the S_b (size) is.

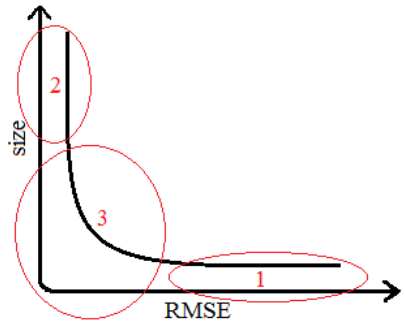


Figure 4.5. The division of Pareto front into three regions after applying the k-means algorithm when k = 3

4.2.5 The Overall Method

Multi-objective strength Pareto front evolutionary algorithm (Zitzler et al., 2001) is the basis of the LBDP method. Single point crossover and bit string mutation are used to generate the next population by applying to the rows and columns separately. The objective values are calculated based on Equation (4.2) and the fitness value is calculated as described in Section 3.2.5. After the stopping

criteria is satisfied, LBDP selects the final bicluster from the Pareto front individuals using the procedures describe in Section 4.2.4. Figure 4.6 shows a flowchart of the proposed method.

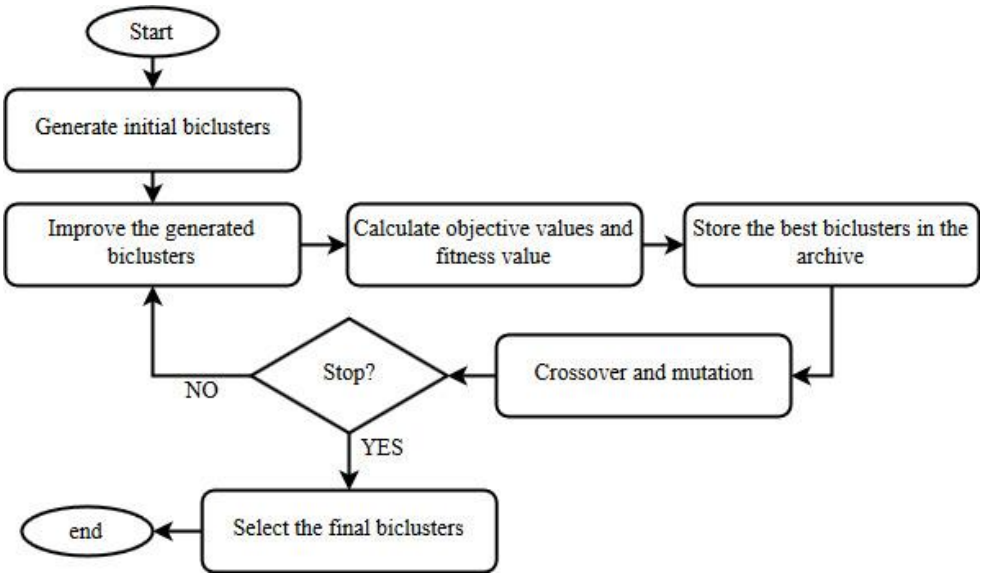


Figure 4.6. The overall flow diagram of the LBDP method

We can also consider our method as a probabilistic model as well. The maximum likelihood solution is equivalent to the least square solution when the noise is assumed to be Gaussian. The following shows the proof of the above statement.

In our model, $y_i = Ax_i + B$, let consider B as a noise term that follows a normal Gaussian distribution $B \sim G(0,\sigma^2)$. The probability density function of B can be written as $p(B) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{B^2}{2\sigma^2}}$. Therefore, the probability of y_i given x_i and parametrized by A is $p(y_i|x_i; A) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - Ax_i)^2}{2\sigma^2}}$. Therefore, the likelihood

of A given all X and Y is $L(A) = p(Y|X;B)$. Given that all m observations are independent then the likelihood of A is $L(A) = \prod_{i=1}^{m} p(y_i|x_i;B) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - Ax_i)^2}{2\sigma^2}}$. The log-likelihood function obtained by taking the logarithm on both sides gives $\log L(A) = log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - Ax_i)^2}{2\sigma^2}} = \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - Ax_i)^2}{2\sigma^2}} = m \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y_i - Ax_i)^2$. From this, we can see that maximization the likelihood function $L(A)$ is the same as minimizing $\sum_{i=1}^{m} (y_i - Ax_i)^2$ which is the same function that is used in the least square formulation.

4.3 Results and Discussion

In order to evaluate the performance of the proposed method, both synthetic and real datasets are considered. There are a number of parameters to control the evolutionary search. These parameters are obtained experimentally and are set as maximum number of iterations = 150; population size = 100; archive size = 20; crossover probability = 0.8; mutation probability = 0.2. The effects of these parameters are discussed in Table 4.1. The dataset used in this experiment is the same as in Experiment 2 of Section 4.3.2 below. Generally, these values decide the tradeoff between exploration and exploitation of individuals during the evolutionary process. If these values are too small, then we will have a fast and premature convergence which do not represent a good bicluster. The number of individuals in the archive population does not affect the performance of the method, however, with the bigger number of individuals in the population and the number of iterations, the performance of the proposed method increases. In

order to have the right balance between exploitation and exploration, the parameters are selected as above.

Table 4.1. The effects of the parameters on the method's performance

Parameter	Value	Performance
Maximum iteration	50	0.77
	100	0.82
	150	0.86
	200	0.92
Population size	50	0.73
	100	0.77
	150	0.62
	200	0.94
Archive size	10	0.92
	20	1
	30	1
	40	0.99
Crossover probability	0.6	0.92
	0.7	0.94
	0.8	1
	0.9	0.82

4.3.1 Parameter Setting

Figure 4.7 examines the effects of the local search thresholds α_{rr}, α_{rc}, α_{ar} and α_{ac} on the accuracy of the biclustering using the Jaccard index defined in Equation (2.1). The data matrix in this experiment is the same as in Experiment 2 of Section 4.3.2 below. There are five biclusters in this data matrix so the mean of

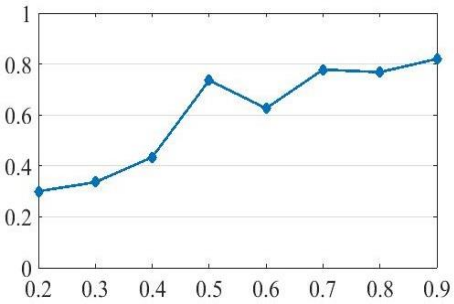
the Jaccard indexes of the detected biclusters indicates the accuracy of LBDP against different parameter values.

When all the parameter values are below 0.5, the accuracy of the method degrades dramatically. However, when all the parameter values are greater than 0.7, the accuracy is about 80% (please refer to Figure 4.7 (a)). Again, the accuracy drops when the parameter values are all close to one. Figure 4.7 (a) shows the accuracy of LBDP (y-axis) when all the parameter values (x-axis) are set to be the same number. It is clear that when the parameter values are smaller than 0.5 the accuracy is poor. By increasing the parameter values, the accuracy of LBDP increases by 30%. Figure 4.7 (b) and (c) show the accuracy of LBDP against different parameter values of α_{rr} and α_{rc} respectively. The highest value is achieved when α_{rr} = 0.85 and α_{rc} = 0.85. Figure 4.7 (d) and (e) show the accuracy of LBDP against different addition parameters. α_{ar} = 0.9 and α_{ac} = 0.95 achieve the highest accuracy.

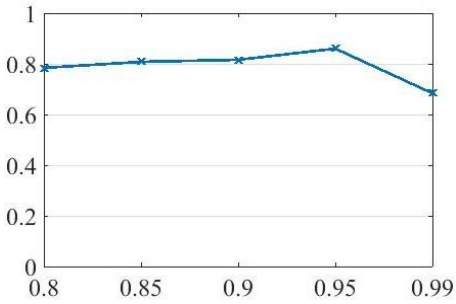
The effect of the number of rows and columns to be correlated and added and/or removed to/from a bicluster and the reason for choosing the threshold to be 75% is shown in Figure 4.7 (f). As can be seen we have the highest Jaccard index value when 75% of rows and columns are selected to find the correlation. The performance is not effected heavily when less number of rows and columns are used, only the running time increases. When we chose values bigger than 75%, the performance decreases since the size of a bicluster gets smaller.

It can be observed from Figure 4.7 that the accuracy of the biclusters is not dramatically affected by the choice of parameter values when they are greater than 0.8. When we select very small parameter values for all four thresholds, the accuracy of the method degrades because this allows any correlated rows or columns to be deleted and any uncorrelated rows or columns to be added. Better

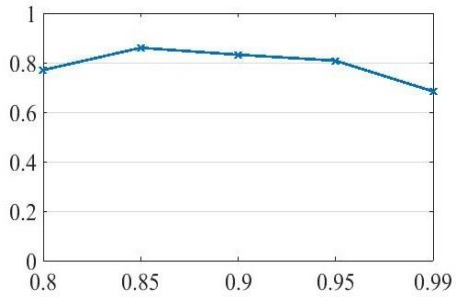
accuracy is obtained when all parameter values are greater than 0.5, and the deletion parameters are smaller than the addition parameters.



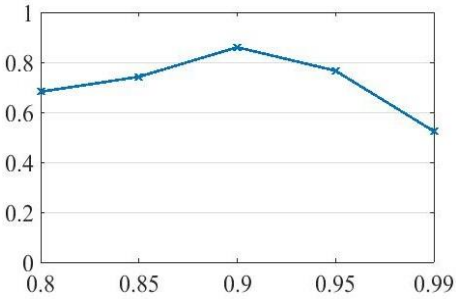
(a)



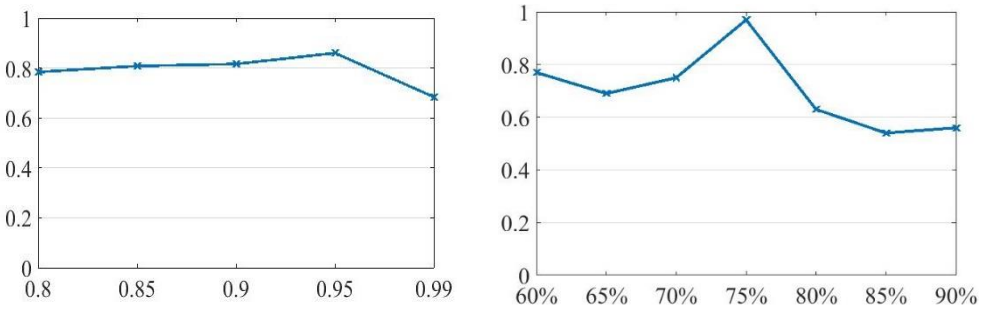
(b)



(c)



(d)



(e) (f)

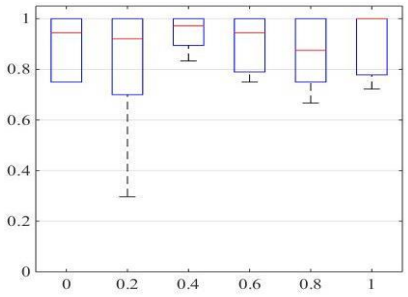
Figure 4.7. The accuracy of LBDP (y-axis) considering different parameter values (x-axis). (a) changing all parameter values at the same time to the same value (b) α_{rr} (c) α_{rc} (d) α_{ar} (e) α_{ac} (f) the number of rows and columns to be removed and/or added from/to a bicluster

4.3.2 Synthetic Data

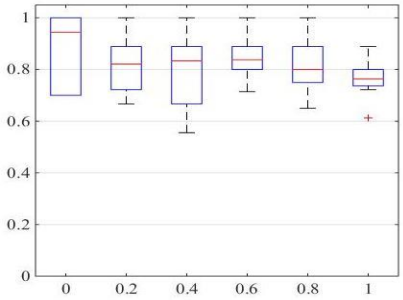
In order to investigate different aspects of LBDP, five experiments using synthetic datasets are performed. The generated data matrices include different types of patterns and different levels of noise.

Experiment 1: This experiment studies the ability of LBDP in detecting different types of patterns. We generate five data matrices with only one bicluster in each data matrix. Four data matrices consists of 20 rows and 15 columns with a noisy background generated from a uniform distribution U(-5,5). The size of the biclusters are 9×9 and the values of biclusters differ between 20 and 30. Each data matrix include different pattern, i.e. constant value pattern, constant row pattern, additive pattern, and linear pattern. The last data matrix is a 100×20 uniform distribution background U(-5,5) with a 25×10 constant column pattern bicluster. Gaussian noise with variance from zero to one is used to degrade each

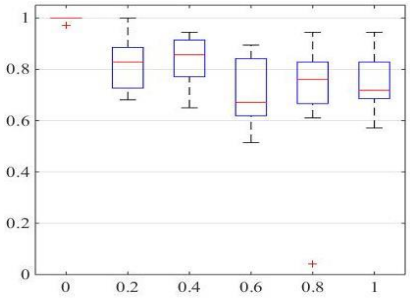
data matrix to evaluate the noise tolerance of LBDP. Figure 4.8 summarizes the distribution of the results. These figures show the boxplots of LBDP for these five data matrices after 10 runs. In these figures, x-axes shows different noise level and y-axes shows the Jaccard index values.



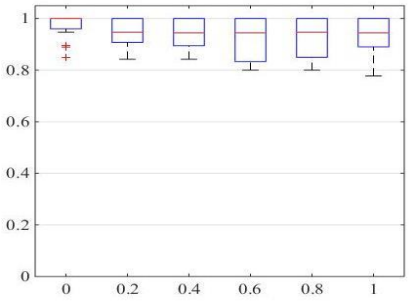
(a)



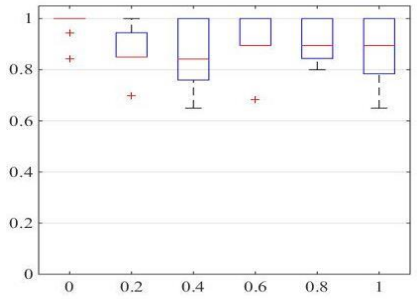
(b)



(c)



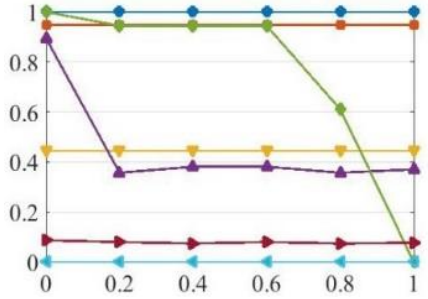
(d)



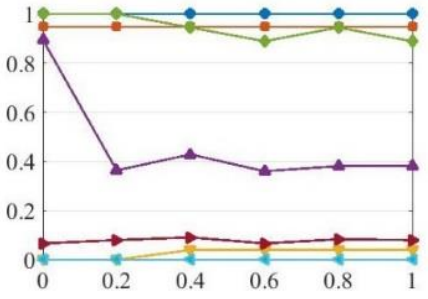
(e)

Figure 4.8. Boxplot of the accuracy (y-axes) for five different data matrices against different noise levels (x-axes) (a) constant value pattern data matrix, (b) constant row pattern data matrix, (c) constant column pattern data matrix, (d) additive pattern data matrix, (e) linear pattern data matrix

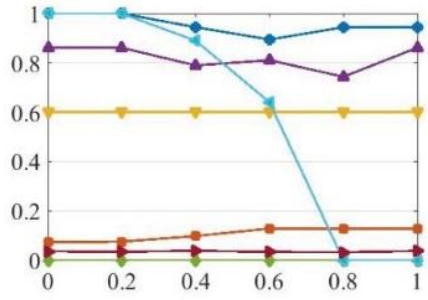
LBDP is also compared with CC (Cheng and Church, 2000), xMotifs (Murali and Kasif, 2003), ISA (Bergmann et al., 2003), OPSM (Ben-Dor et al., 2003) FABIA (Hochreiter et al., 2010) and LAS (Shabalin et al., 2009). BicAT toolbox (Barkow et al., 2006) is used to calculate the results of CC, xMotifs, ISA and OPSM. LAS and FABIA are publicly available at https://genome.unc.edu/las/ and http://www.bioinf.jku.at/software/fabia/fabia.html, respectively. Based on the discussion in (Pontes et al., 2013), we use the default parameters set as given in those methods.


1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

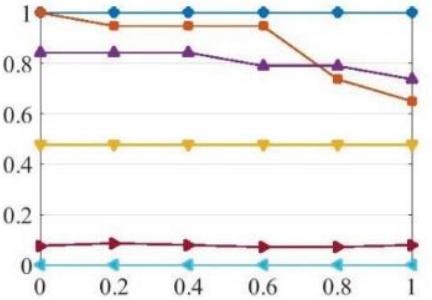
(a)


1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

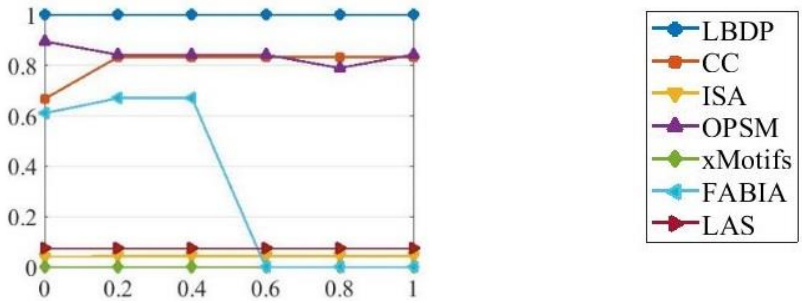
(b)


1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

(c)


1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

(d)



(e)

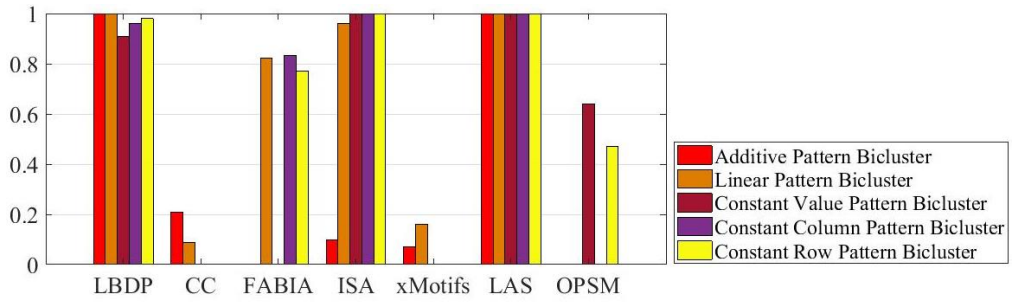
Figure 4.9. The accuracy of the detected biclusters (y-axis) against different noise level (x-axis) (a) constant value pattern data matrix, (b) constant row pattern data matrix, (c) constant column pattern data matrix, (d) additive pattern data matrix, (e) Linear pattern data matrix

Among the methods, xMotifs has the least performance by not detecting any bicluster in the constant column pattern data matrix, additive pattern data matrix and linear pattern data matrix. However, this method identifies constant value pattern data matrix when the noise level is low and constant row pattern data matrix. This is caused by the discretization step of the method in which the coherent pattern desired by the method is only achieved by the constant pattern data matrices. ISA also performs poorly with almost zero Jaccard index in the constant row pattern data matrix and the linear pattern data matrix. ISA detects well when the mean value of a bicluster or the rows or columns in the bicluster are high. ISA is also robust to noise. This is why the method identifies parts of biclusters in the constant value pattern data matrix, constant column pattern data matrix, and additive pattern data matrix. Regardless of the poor performance of LAS, this method exhibits a robust behavior with respect to noise. FABIA assumes a multiplicative model to detect biclusters and this is the reason FABIA

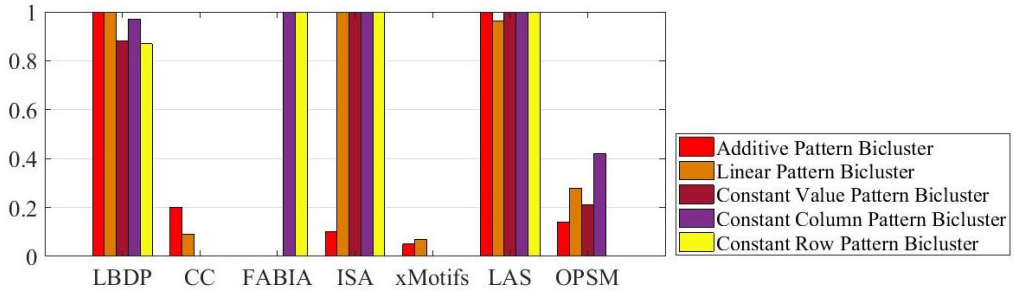
fails to detect any bicluster in the constant value pattern data matrix, constant row pattern data matrix, and additive pattern data matrix.

On the other hand, CC performs very well except for the constant column pattern data matrix and this behavior is explained by the fact that the embedded biclusters are between certain values and have a perfect mean square residue, which is the criteria CC tries to minimize. OPSM performs well in all data matrices. As it is clear, the performance of the LBDP outperforms all the other methods.

Experiment 2: The second experiment evaluates the performance of LBDP when all patterns are included in a data matrix. The background matrix is generated with 200 rows and 60 columns from a uniform distribution U(-5,5). The sizes of the biclusters are 40×7, 25×10, 40×8, 19×9, and 20×20 for constant row pattern, constant column pattern, constant value pattern, linear pattern and additive pattern, respectively. The values of these biclusters differ between 20 and 30. Gaussian noise with variance of 0.3 is used to degrade the biclusters. Figure 4.10 shows the comparison results.



(a)



(b)

Figure 4.10. The accuracy (y-axis) of detecting different patterns in a data matrix (a) without noise, (b) with Gaussian noise variance 0.3

Based on these figures, ISA recovers all patterns except for the additive pattern bicluster, which decreases the overall performance of the method. Unlike Experiment 1, CC performs weakly by identifying only parts of the linear pattern bicluster and additive pattern bicluster. Interestingly, the accuracy of the detected biclusters does not change when the noise degrades the biclusters. This happens because the noise is small enough to keep the MSR value below the given threshold.

In this experiment, LAS slightly outperforms the proposed method in a few bicluster patterns by about 8%. However, in Experiment 1, the accuracy of LAS was only about 10% for all five patterns, whereas LBDP achieved an accuracy of 100%. This huge performance variation indicates that LAS does not have a robust performance and behaves widely in different situation.

Experiment 3: In order to evaluate the computation time of LBDP, we generated ten new synthetic data matrices in Experiment 3. All these data

matrices have one linear pattern bicluster with 70 rows and 30 columns and one additive pattern bicluster with 90 rows and 50 columns. The values of the biclusters varies between 20 and 30. The dimension of the data matrices varies from 100 to 1000 with 100 intervals. Figure 4.11 shows the computation time of LBDP against the dimension of the data matrices. The computation time increases dramatically when the dimension of data matrix increases from 100 to 200. However, it remains almost steady when increasing the dimension from 200 to 1000.

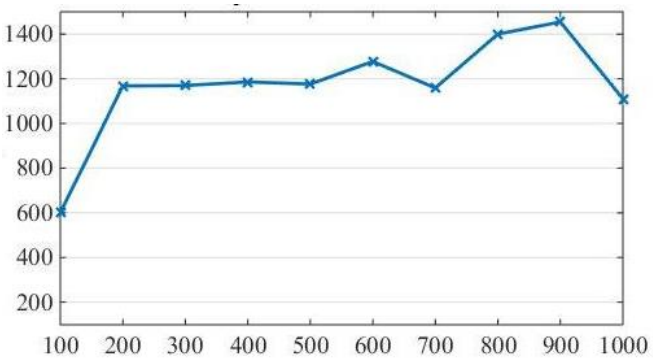
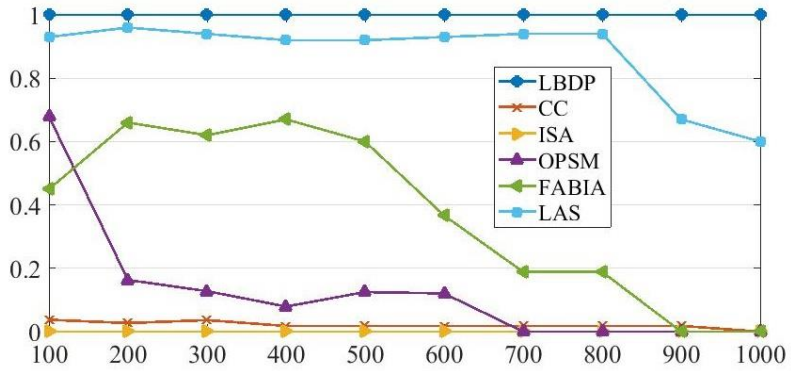


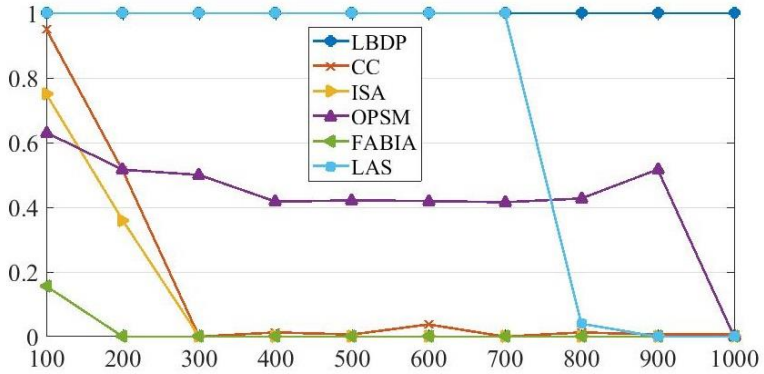
Figure 4.11. The computational time of LBDP (y-axis (ms)) against the dimension of data matrix (x-axis)

We also compare the accuracy of the detected biclusters with the other methods. xMotifs is not able to process the data matrices with dimension larger than 64. Figure 4.12 shows the comparison results. The accuracy of the linear pattern bicluster is under 10% for both CC and ISA, and their performance drop in the additive pattern bicluster when the dimension is greater than 300. Among these methods, LAS performs as good as LBDP by identifying over 90% of the biclusters when the dimension stays below 800, however its performance drops

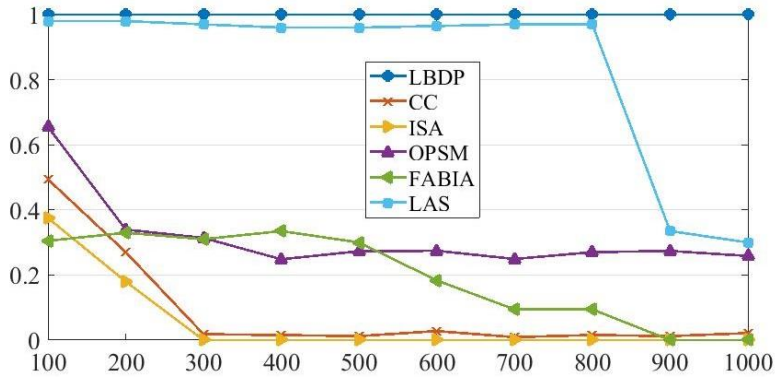
when the dimension of data matrix is larger than 800. LBDP identifies both biclusters accurately.



(a)



(b)



(C)

Figure 4.12. The accuracy of the detected biclusters (y-axis) against different dimension (x-axis) data matrices (a) linear pattern bicluster, (b) additive pattern bicluster, (c) the overall accuracy of the methods

Experiment 4: This experiment studies the behavior of LBDP in detecting the overlapped biclusters. We generated five new data matrices with 20 rows and 20 columns with a noisy background from U(-5,5). Each data matrix includes two 9×9 constant row biclusters with no noise. Figure 4.13 shows the comparison results of this experiment. Overlap levels shows the number of rows and columns that are in common (overlap level four means there are four rows and four columns in common). In general, the increasing level of overlap does not heavily influence the performance of the methods. CC and LAS are the two exceptions by dropping up to 70% in their performances.

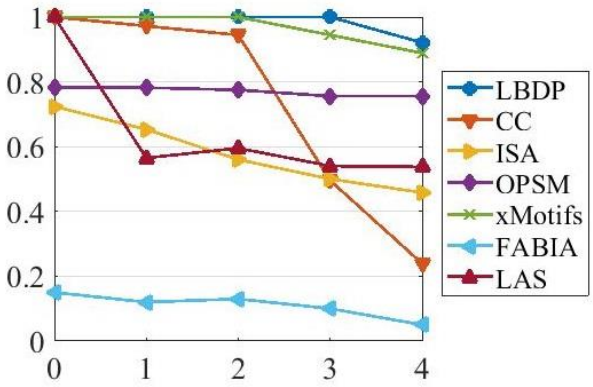


Figure 4.13. The accuracy of detected biclusters (y-axis) against the level of overlap (x-axis)

Experiment 5: In this experiment, we consider different scenarios in a new synthetic data matrix. In this new data matrix, there are 200 rows, and 100 columns with 4 embedded biclusters. The first bicluster is an additive pattern bicluster with 50 rows and 15 columns; the second bicluster is a linear pattern bicluster with 30 rows and 10 columns; the last two biclusters are constant row pattern biclusters with 50 rows and 30 columns and 20 rows and 20 columns, with 4 rows and 4 columns in common. The background matrix has uniform distribution U(-5,5). Gaussian noise with variance of 0.3 is used to degrade both biclusters and the background matrix. Figure 4.14 presents the results. CC performs very poor by identifying only scatter parts of one of the constant row pattern bicluster. As in Experiment 2, LAS achieves its best performance when the number of biclusters increases. LAS detects biclusters by locally maximizing a Bonferroni correction as described in Experiment 2.

In this experiment, LAS outperforms LBDP by about 4% in overall performance. However, LAS performs poorly in Experiment 1, Experiment 3, and Experiment 4. LBDP outperformed LAS by 90% in Experiment 1, 16% in

Experiment 3, and 34% in Experiment 4. The large differences in the performance comparison of LAS and LBDP show that the performance of LAS is not stable and it behaves differently in different situation.

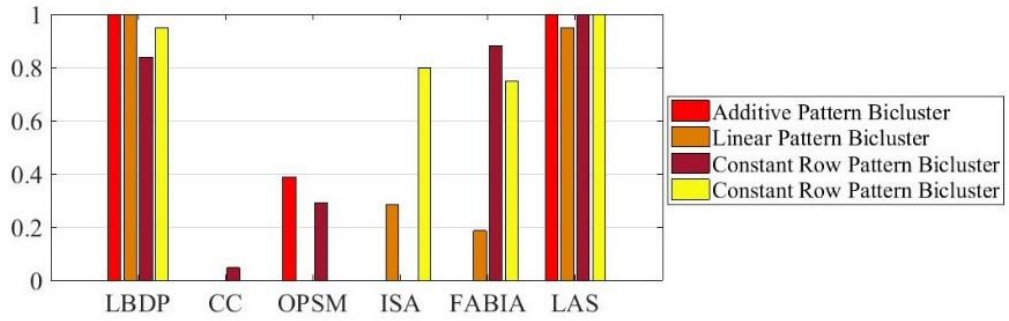


Figure 4.14. The accuracy of the detected biclusters in the overlapped, noisy data matrix

4.3.3 Gene Expression Data

In order to evaluate the performance of LBDP for biomedical data, five different real datasets are used. Table 4.2 summarizes the information of these datasets. The last three datasets are available online at https://www.ncbi.nlm.nih.gov/. We replace the null values in these datasets with a random value chosen among the same condition.

Table 4.2. Data matrices description

Data matrix	**# Genes**	**# Conditions**	**Organism**
Saccharomyces Cerevisiae Yeast	2884	17	Saccharomyces Cerevisiae
Multiple human organs	18927	158	Homo sapiens
GDS232 (medulloblastoma tumor)	2059	23	Homo sapiens

Data matrix	# Genes	# Conditions	Organism
GDS750 (unfolded protein response)	6091	13	Saccharomyces Cerevisiae
GDS4085 (breast cancer)	1259	19	Homo sapiens

The first dataset that is used is the *Saccharomyces Cerevisiae* Yeast data. We verify the enrichment of the results and 92% of the detected biclusters have the modular enrichment. Sixty four percent of the biclusters have the singular biological process, the singular cellular component, and the singular molecular function enriched. However, only 30% of the detected biclusters show annotation in the KEGG pathway. Table 4.3 presents the statistics of the 100 detected biclusters.

Table 4.3. Statistics of the 100 detected biclusters by LBDP on the Yeast dataset

	# Rows	# Cols	Size	RMSE
Average values	78.34	9.4	625.56	5.110402554
Maximum values	277	16	1944	20.58315
Minimum values	19	5	168	1.56239E-14

Table 4.4 to Table 4.8 summarize the evaluation of one of the detected biclusters using GO term and KEGG pathway. These tables show only the results having the lowest p-value. The bicluster contains 79 rows and 13 columns. The RMSE and size are 4.1 and 1027, respectively. The results show the enrichment of the detected bicluster in comparison to the background occurrence frequency.

Table 4.4 to Table 4.7 list the significant GO terms in the detected biclusters, and Table 4.8 shows the KEGG pathway of the same bicluster.

Table 4.4. Modular Enrichment Analysis (GO and KEGG Annotation)

Genes	NGR (TNGR)	NG (TNG)	Hyp	Hyp*	Annotations
YDL082W, YLR344W, YBR031W, YNL096C, YBR181C, YDL130W, YPR102C, YDR064W, YDR382W	96 (7109)	9 (79)	9.65392e-07	8.25411e-05	GO:0002181: cytoplasmic translation (BP) GO:0003735: structural constituent of ribosome (MF) GO:0005737: cytoplasm (GCC) (KEGG) 03010: Ribosome

Table 4.5. Cellular Component Ontology

Genes	NGR (TNGR)	NG (TNG)	Hyp	Hyp*	Annotations
YBR079C, YDL082W, YMR255W, YLR344W, YJR015W, YMR009W, YDR212W, YDR050C, YCL034W, YLR432W, YCL018W, YBR031W, YNL096C, YBR181C, YLR384C, YLR414C, YDL130W, YOL081W, YDL127W, YBL068W, YBR172C, YJL001W, YMR184W, YIL107C, YDR378C, YPR102C, YDR152W, YDL103C, YNL163C, YKR079C, YAR009C, YML039W, YBR155W, YFL039C, YDL143W, YDR064W, YDR016C, YAL053W, YDR382W, YDR017C, YDL182W, YNL281W	2082 (7109)	42 (79)	6.75886e-06	0.000750233	GO:0005737: cytoplasm (GCC)

Table 4.6. Biological Process Ontology

Genes	NGR (TNGR)	NG (TNG)	Hyp	Hyp*	Annotations
YER102W, YDL082W, YLR344W, YBR031W, YNL096C, YBR181C, YDL130W, YPR102C, YDR152W, YDR064W, YDR382W	171 (7109)	11 (79)	2.4763e-06	0.000440781	GO:0002181: cytoplasmic translation (BP)

Table 4.7. Molecular Function Ontology

Genes	NGR (TNGR)	NG (TNG)	Hyp	Hyp*	Annotations
YER102W, YDL082W, YLR344W, YBR031W, YNL096C, YBR181C, YDL130W, YPR102C, YDR064W, YDR382W	229 (7109)	10 (79)	0.000202629	0.0216813	GO:0003735: structural constituent of ribosome (MF)

Table 4.8. Singular Enrichment Analysis of KEGG Pathway

Genes	NGR (TNGR)	NG (TNG)	Hyp	Hyp*	Annotations
YDR064W, YPR102C, YDR382W, YNL096C, YBR031W, YBR181C, YDL082W, YER102W, YDL130W, YLR344W	159 (7109)	10 (79)	9.02328e-06	0.000234605	(KEGG) 03010: Ribosome

We next analyze the gene expression data of multiple human organs (Son et al., 2005). The data is publicly available at https://home.ccr.cancer.gov/oncology/oncogenomics/. In (Prelić et al., 2006), they used t-testing as the main analysis tool. Table 4.9 lists the detected biclusters

of the different organs. The gene expression pattern of each organ is the detected genes in a bicluster under replicated conditions of each organ.

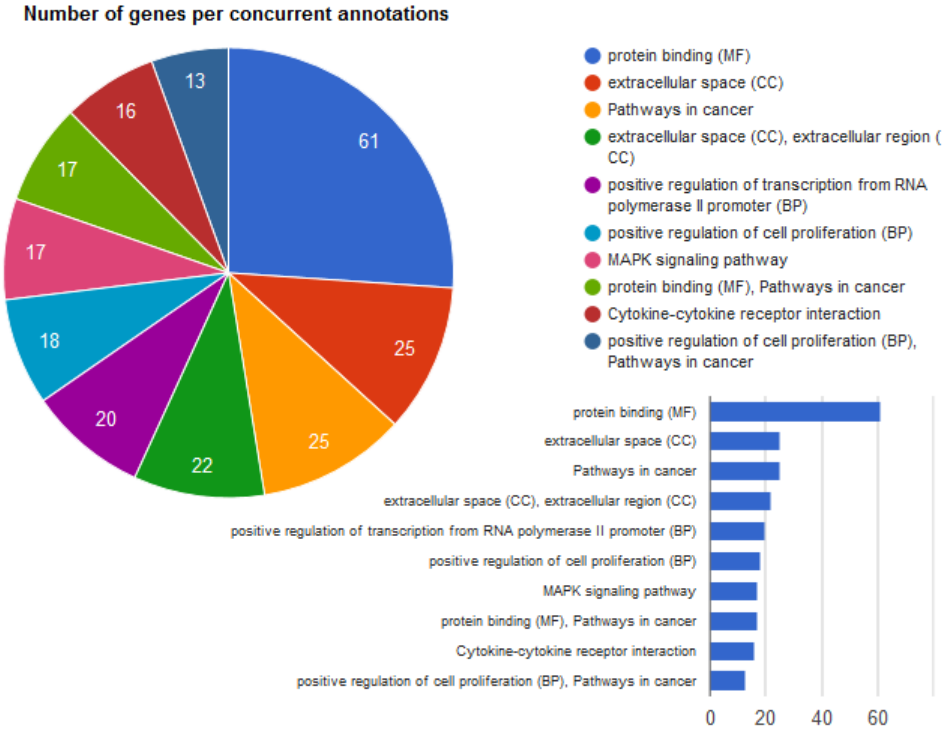
The enrichment analysis is also conducted to evaluate the enrichment of biological function with GO and KEGG pathways, and we compare the results with (Zhao et al., 2008) and (Prelić et al., 2006). In Table 4.9, the first column is the name of the organs in the dataset, the second column is the number of replicated conditions of each organ. The three following columns are the number of detected genes in (Prelić et al., 2006), the number of detected genes in (Zhao et al., 2008), and the number of detected genes in LBDP, respectively. Obviously, the number of detected genes in LBDP is larger than that in (Prelić et al., 2006) except in three organs, i.e. Bladder, Liver, and SkelM. LBDP is also able to extracts biclusters in Colon, Ileum, Ovary, Stomach, and Uterus. The number of detected genes are larger in comparison to (Zhao et al., 2008) in some organs and smaller in other organs. The last two columns are the significantly annotated gene ontology ID and the corresponding p-value in LBDP.

Table 4.9. The biclusters of 19 organs detected by LBDP and their GO term

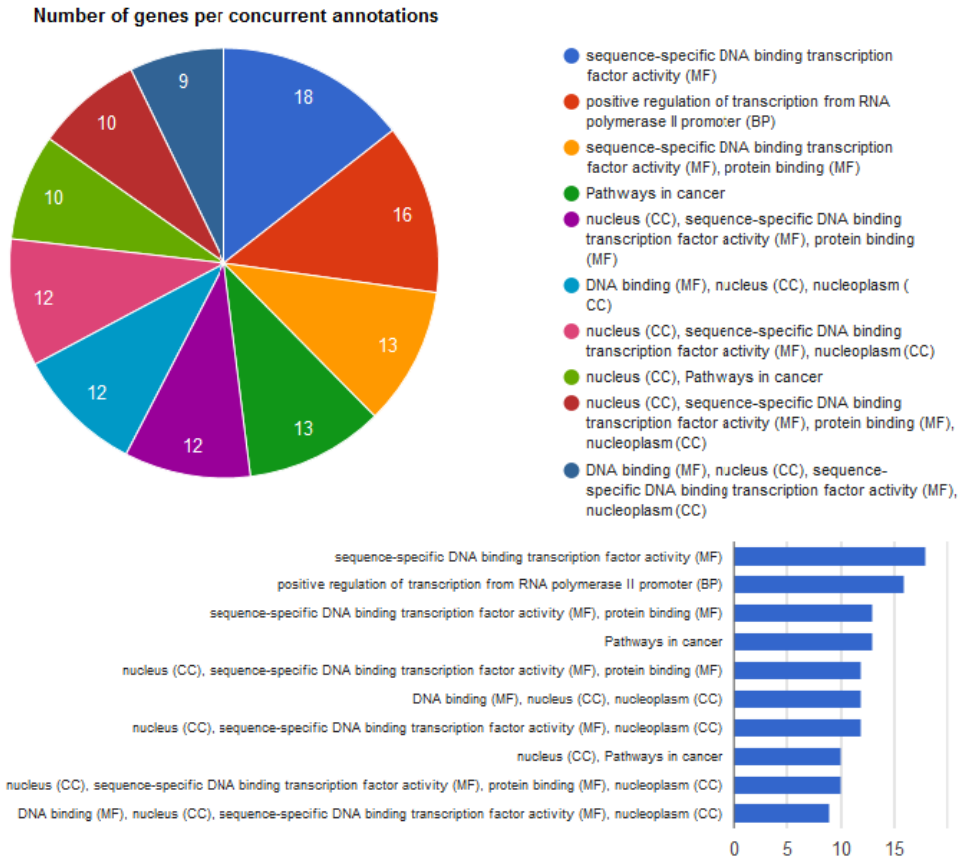
Organ	**# Conditions**	**# Genes in (Prelić et al., 2006)**	**# Genes in (Zhao et al., 2008)**	**# Genes in LBDP**	**Gene Ontology ID in LBDP**	
					Gene Ontology ID	**p-value**
Adrenal	9	2	36	60	GO:0005488	1.73E-09
Bladder	9	104	165	80	GO:0009987	1.94E-08
Cerebellum	6	4	41	41	GO:0065007	5.50E-06
Cerebrum	7	5	38	41	GO:0050789	1.60E-05
Colon	8	-	57	92	GO:0044424	3.75E-09
Heart	7	5	38	34	GO:0005575	3.91E-05

Organ	# Conditions	# Genes in (Prelić et al., 2006)	# Genes in (Zhao et al., 2008)	# Genes in LBDP	Gene Ontology ID in LBDP	
					Gene Ontology ID	p-value
Ileum	10	-	32	96	GO:0050794	9.24E-07
Kidney	10	11	59	41	GO:0044699	2.16E-05
Liver	10	54	118	19	GO:0005575	3.63E-05
Lung	9	17	64	26	GO:0044464	1.72E-05
Ovary	5	-	25	7	-	-
Pancreas	6	6	25	92	GO:0009987	1.44E-07
Prostate	8	3	22	42	GO:0003674	3.31E-06
SkelM	9	10	61	6	GO:0005575	3.63E-05
Spleen	10	7	26	31	GO:0005575	4.49E-07
Stomach	10	-	34	22	GO:0009889	1.28E-05
Testes	7	25	43	48	GO:0003674	7.13E-06
Ureter	8	4	37	73	GO:0043227	1.25E-05
Uterus	10	-	16	66	GO:0043226	1.77E-06

The next dataset we analyzed is GDS232 (MacDonald et al., 2001), in which LBDP is able to categories the two types of tumor correctly. Figure 4.15 shows the number of genes per concurrent annotations. Figure 4.15 (a) shows the modular enrichment analysis between GO terms (cellular component, molecular function, biological process) and KEGG pathway. Each part of the pie chart shows the number of genes participating in each annotation. 26.1% of detected genes in LBDP have protein bindings in human Medulloblastoma Tumor. In Figure 4.15 (b), 14.4% of detected genes have DNA binding transcription factor activity.


Number of genes per concurrent annotations
61
25
25
22
20
18
17
17
16
13
protein binding (MF)
extracellular space (CC)
Pathways in cancer
extracellular space (CC), extracellular region (CC)
positive regulation of transcription from RNA polymerase II promoter (BP)
positive regulation of cell proliferation (BP)
MAPK signaling pathway
protein binding (MF), Pathways in cancer
Cytokine-cytokine receptor interaction
positive regulation of cell proliferation (BP), Pathways in cancer
protein binding (MF)
extracellular space (CC)
Pathways in cancer
extracellular space (CC), extracellular region (CC)
positive regulation of transcription from RNA polymerase II promoter (BP)
positive regulation of cell proliferation (BP)
MAPK signaling pathway
protein binding (MF), Pathways in cancer
Cytokine-cytokine receptor interaction
positive regulation of cell proliferation (BP), Pathways in cancer
0
20
40
60

(a)



(b)

Figure 4.15. Number of genes in concurrent annotations including GO terms and KEGG pathways in (a) human Medulloblastoma Tumour (b) Brain Tumour

The fourth dataset we analyzed is GDS750 (Leber et al., 2004). Figure 4.16 shows the biological enrichment of the detected biclusters. The GO level does not affect the results of KEGG pathway analysis. On the other hand, if GO annotations are used the GO annotation level shows the level on the hierarchy. The lowest level is the deepest level of the ontology, which shows annotations obtained directly from the database (Tabas Madrid et al., 2012, Nogales Cadenas et al., 2009, Carmona Saez et al., 2007).

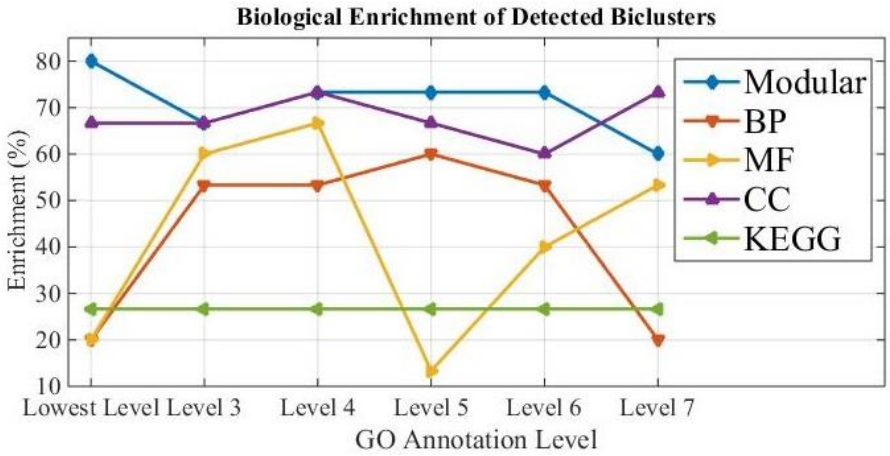


Figure 4.16. Biological enrichment (y-axis (%)) of detected biclusters against different GO annotation level (x-axis)

The last dataset we analyzed is GDS4085 (Julien et al., 2011). We evaluated the detected genes by LBDP for biological enrichment. Table 4.10 summarizes the biological enrichment of the detected genes. The first column indicates if the detected conditions are cancerous or not.

Table 4.10. The biological enrichment of the breast cancer

Carcinoma	Annotation	p-value
Positive	GO:0010811: positive regulation of cell-substrate adhesion (BP)	0.00327021
Positive	GO:0004576: oligosaccharyl transferase activity (MF)	0.000467665
Positive	GO:0033276: transcription factor TFTC complex (GCC)	0.00163611
Positive	(KEGG) 00510: N-Glycan biosynthesis	0.0057176
Negative	GO:0006488: dolichol-linked oligosaccharide biosynthetic process (BP)	0.00187006
Negative	GO:0051879: Hsp90 protein binding (MF)	0.000876808
Negative	(KEGG) 00510: N-Glycan biosynthesis	0.00286282

4.3.4 Image Data

We also apply LBDP to cluster the Image Segmentation dataset from the UCI Machine Learning Repository. The Image segmentation (IS) dataset is publically available at https://archive.ics.uci.edu/ml/datasets.html which includes 2310 rows and 19 columns. The samples (rows) are chosen randomly from seven outdoor images, namely grass, path, window, cement, foliage, sky, and brick face. The images are hand segmented and for each image, 19 features (columns) is extracted including color information, contrast information, and region information. LBDP is able to cluster images into meaningful clusters. The biggest detected bicluster includes three major image groups that is grass, path, and cement, with three common features. From visual inspection and as shown in Figure 4.17, we can see that the images in the right group (cement) and left group (grass) correspond to components of the path images (in the middle). This shows that LBDP can uncover semantic information and higher-level concepts within images.

Based on the reported results in (Tung et al., 2005), our results show more comprehensibility. Here, we can see why images ended up in the same bicluster while the relationship of the images is not clear in (Tung et al., 2005).

Figure 4.17. Detected biclusters group images with similar concepts

4.3.5 Facebook Data

We apply LBDP to the Social Circles Facebook dataset (Leskovec and Mcauley, 2012), which consists of circles (friends' lists) from Facebook. This dataset is publicly available at https://snap.stanford.edu/data/egonets-Facebook.html and includes 4039 nodes and 88234 edges. In this dataset, there are 10 networks where each user is represented by a set of features. We run LBDP for each network separately as in PBD-SPEA (Golchin and Liew, 2017).

In Table 4.11, we report the number of IDs and features in the detected biclusters in LBDP and the number of IDs and features in the detected biclusters in PBD-SPEA (Golchin and Liew, 2017). We also compare the mean value of the pairwise cosine distance of the detected biclusters μ_{cd} in LBDP and PBD-SPEA. Generally, the sizes of the detected biclusters in LBDP are smaller in comparison to the ones in PBD-SPEA. However, the mean values are also smaller which shows the detected biclusters in LBDP are more coherent.

Table 4.11. Biclustering results on 10 different Facebook network

Net. NO.	Detected bicluster in LBDP		Detected bicluster in PBD-SPEA (Golchin and Liew, 2017)		μ_{cd} detected bicluster in LBDP	μ_{cd} detected bicluster in PBD-SPEA (Golchin and Liew, 2017)
	# IDs	# features	# IDs	# features		
1	96	22	180	30	0.1052	0.2267
2	130	120	496	200	0.1579	0.3577
3	89	20	91	20	0.0967	0.0986
4	45	14	78	10	0.0965	0.0986
5	63	23	63	29	0.2571	0.2574
6	30	10	24	8	0.1834	0.1622
7	319	52	416	97	0.1872	0.1984
8	290	103	449	106	0.1989	0.2043
9	131	23	221	26	0.08	0.0893
10	14	10	20	10	0.108	0.1165

4.4 Conclusion

In this Chapter, we propose a multi-objective evolutionary algorithm called LBDP to uncover bicluster patterns in data. LBDP uses the strength Pareto front evolutionary optimization algorithm to handle the conflicting objectives in the fitness function. The multi-objective function consists of the RMSE, bicluster size, Jaccard distance of each subgroup to other subgroups, and Jaccard distance between individuals of each subgroup. LBDP first generates a population of individuals, each consisting of bicluster with random rows and columns. Then, it tries to find the hyperplane that best fit the bicluster using SVD. To obtain multiple biclusters in the final solution, the Pareto front individuals are divided into groups based on the number of biclusters expected in the final solution. The k-means clustering algorithm is then applied to each group to select the final bicluster.

LBDP is able to detect multiple and different type of biclusters in a dataset with high accuracy, even in the presence of noise. The geometric biclustering framework allows the robust detection of biclusters since hyperplane detection is robust with respect to noise.

We evaluated the performance of LBDP using several gene expression datasets, and validated the detected biclusters using enrichment analysis based on the GO and KEGG databases. To demonstrate the applicability of LBDP to other applications, we showed that LBDP was able to uncover semantic information within images using the Image Segmentation dataset.

5

Conclusions and Future Work

This chapter summarizes the contributions of the research presented in this thesis. In Section 5.1 we present our contributions to the biclustering problem. In Section 5.2, we outline a few potential future directions of this research and conclude the thesis.

5.1 Contributions

In this research, we study the problem of biclustering and proposed two novel biclustering algorithms based on evolutionary optimization. To achieve this goal we first investigated existing approaches and identified several shortcomings as follows. First, despite the efforts that have been made in evolutionary biclustering so far, they have not been able to effectively solve the problem of detecting multiple biclusters concurrently. In most cases, these methods use multi population or multiple run to tackle the problem of finding multiple biclusters. In addition, these methods detect only limited types of patterns and are not able to detect the most general class of linear patterns. As a result, there is still a need to develop novel EA based approaches that address the above issues.

Geometrical biclustering has recently being proposed as an effective framework for finding linear bicluster patterns. However, most geometrical biclustering algorithms rely on the use of Hough transform. The computational

cost of HT is $O(N\ N_a^{n-1})$ and the memory complexity of HT is $O(N_a^n)$ that is expensive, where N is the size of data, N_a is the size of accumulation arrays, n is the dimension of the data (XU and OJA, 1993). For reasonable quantization of the Hough parameter space, Na is large, hence HT is costly in both speed and memory.

In this study, we have addressed the issues we identified and developed two novel algorithms, namely PBD-SPEA and LBDP. The PBD-SPEA algorithm has the following innovations. First, we proposed a new encoding scheme that groups all biclusters in a single individual, thus allowing the detection of multiple biclusters. Second, we introduced a new crossover operation to search for similar biclusters inside the individuals and to generate the offspring based on the exploration and exploitation strategy. Third, we proposed a new heuristic mutation operation based on local search to improve the quality of the biclusters. Finally, we applied a multi-objective evolutionary algorithm based on strength Pareto front to guide our search space. Our experiments using both biomedical and non-biomedical datasets have shown its effectiveness and robustness in comparison to several state-of-the-arts. However, PBD-SPEA is not able to detect general linear patterns in biclusters. To be able to detect multiple biclusters with linear patterns, we incorporate the concept of geometric biclustering into multi-objective evolutionary optimization and proposed the LBDP algorithm. The EA based search allows LBDP to use SVD to detect the hyperplane in an effective manner, and allows the analysis of large datasets.

For the gene expression data, our methods are able to detect highly enriched biclusters. For the image data matrix, we are able to discover higher-level semantic information within groups of images. For the Facebook data matrix, PBD-SPEA is able to detect coherent sub-matrices of users and features that are useful for subsequent analysis.

5.2 Future Works

The research conducted in this thesis leads to a number of possible future directions. Our future works in the field of biclustering can be generally categorised into three groups as follows:

Studying the impact of proposed methods in big data matrices: As it was shown in this study, we have successfully applied our methods in high dimensional datasets (Chapter 4). However, the computational cost of detecting biclusters when both the number of rows and the number of columns are large is still expensive. Our future approach will be to develop heuristics based on a divide-and-conquer strategy, where a big data matrix is divided into a number of smaller matrices along the row direction. In order to find maximal biclusters, we will unify the sub-biclusters with common columns.

Extending the application of our algorithms: we aim at investigating the application of our methods for more general cases such as prediction analysis and market segmentation. In prediction analysis, first our proposed methods are used to extract biclusters among datasets. Then, for each bicluster, a classifier identifies if a sample is inside or outside of that bicluster. Finally, all the classifiers are combined to classify samples in the datasets into several subgroups.

In the field of market segmentation, each bicluster is a subgroup of customers with similar characteristic in the bicluster and different in the remaining ones. These biclusters can be used to take full advantage of customer knowledge for product development. Biclustering can also overcomes the data dimensionality problem.

Developing an online biclustering server: As we extend our studies for different problems, we intend to build an online biclustering web server and make it available publicly. The web server will provides an easy to use and effective online tool to find coherent patterns in datasets.

List of References

ADOMAS, A., HELLER, G., OLSON, Å., OSBORNE, J., KARLSSON, M., NAHALKOVA, J., VAN ZYL, L., SEDEROFF, R., STENLID, J. & FINLAY, R. 2008. Comparative Analysis of Transcript Abundance in Pinus Sylvestris after Challenge with a Saprotrophic, Pathogenic or Mutualistic Fungus. *Tree Physiology,* 28**,** 885-897.

ALIZADEH, A. A., EISEN, M. B., DAVIS, R. E., MA, C., LOSSOS, I. S., ROSENWALD, A., BOLDRICK, J. C., SABET, H., TRAN, T. & YU, X. 2000. Distinct Types of Diffuse Large B-cell Lymphoma Identified by Gene Expression Profiling. *Nature,* 403**,** 503-511.

ALON, N. & SPENCER, J. H. 2004. *The Probabilistic Method* John Wiley & Sons, INC.

ARAÚJO, D. R., BASTOS-FILHO, C. J., BARBOZA, E. A., CHAVES, D. A. & MARTINS-FILHO, J. F. A Performance Comparison of Multi-objective Optimization Evolutionary Algorithms for All-optical Networks Design. IEEE Symposium on Computational Intelligence in Multicriteria Decision-making (MDCM), 2011. IEEE, 89-96.

ARUN, K. S., HUANG, T. S. & BLOSTEIN, S. D. 1987. Least-squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 9**,** 698-700.

ASHBURNER, M., BALL, C. A., BLAKE, J. A., BOTSTEIN, D., BUTLER, H., CHERRY, M. J., DAVIS, A. P., DOLINSKI, K., DWIGHT, S. S. & EPPIG, J. T. 2000. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics,* 25**,** 25-29.

BABU, M. M. 2004. Introduction to Microarray Data Analysis. *Computational Genomics: Theory and Application,* 17, 225-249.

BAILEY, K. D. 1994. Numerical Taxonomy and Cluster Analysis. *Typologies and Taxonomies.* Thousand Oaks, California: SAGE Publications, Inc.

BARKOW, S., BLEULER, S., PRELIĆ, A., ZIMMERMANN, P. & ZITZLER, E. 2006. BicAT: A Biclustering Analysis Toolbox. *Bioinformatics,* 22**,** 1282-1283.

BEN-DOR, A., CHOR, B., KARP, R. & YAKHINI, Z. 2003. Discovering Local Structure in Gene Expression Data: The Order-preserving Submatrix Problem. *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology,* 10**,** 373-384.

BERGMANN, S., IHMELS, J. & BARKAI, N. 2003. Iterative Signature Algorithm for the Analysis of Large-scale Gene Expression Data. *Physical Review E,* 67**,** 403–449.

BINDEA, G., MLECNIK, B., HACKL, H., CHAROENTONG, P., TOSOLINI, M., KIRILOVSKY, A., FRIDMAN, W.-H., PAGÈS, F., TRAJANOSKI, Z. & GALON, J. 2009. ClueGO: A Cytoscape Plug-in to Decipher Functionally Grouped Gene Ontology and Pathway Annotation Networks. *Bioinformatics,* 25**,** 1091-1093.

BOLOTAEVA, V. & CATA, T. 2011. Marketing Opportunities with Social Networks. *Journal of Internet Social Networking and Virtual Communities,* 2011**,** 1-8.

BOYLE, E. I., WENG, S., GOLLUB, J., JIN, H., BOTSTEIN, D., CHERRY, J. M. & SHERLOCK, G. 2004. GO:: TermFinder—Open Source Software for Accessing Gene Ontology Information and Finding Significantly Enriched Gene Ontology Terms Associated with a List of Genes. *Bioinformatics,* 20**,** 3710-3715.

CARMONA SAEZ, P., CHAGOYEN, M., TIRADO, F., CARAZO, J. M. & PASCUAL MONTANO, A. 2007. GENECODIS: A Web-based Tool for Finding Significant Concurrent Annotations in Gene Lists. *Genome Biology,* 8**,** R3.

CELEBI, M. E. 2014. *Partitional Clustering Algorithms*, Springer.

CHA, K., OH, K., HWANG, T. & YI, G.-S. Identification of Coexpressed Gene Modules across Multiple Brain Diseases by a Biclustering Analysis on Integrated Gene Expression Data. Proceedings of the ACM 8th International Workshop on Data and Text Mining in Bioinformatics, 2014. ACM, 17-17.

CHAUDHARI, P., DHARASKAR, R. & THAKARE, V. M. 2010. Computing the Most Significant Solution from Pareto Front Obtained in Multi-objective Evolutionary. *International Journal of Advanced Computer Science and Applications (IJACSA),* 1**,** 63-68.

CHEN, X., YANG, M., HUANG, J. Z. & MING, Z. 2018. TWCC: Automated Two-way Subspace Weighting Partitional Co-clustering. *Pattern Recognition,* 76**,** 404-415.

CHENG, K. O., LAW, N. F., SIU, W. C. & LIEW, A. W. C. 2008. Identification of Coherent Patterns in Gene Expression Data using an Efficient Biclustering Algorithm and Parallel Coordinate Visualization. *BMC Bioinformatics,* 9**,** 1-28.

CHENG, Y. & CHURCH, G. M. Biclustering of Expression Data. Proceeding of Intelligent Systems for Molecular Biology (ISMB), 2000. American Association for Artificial Intelligence (AAAI), 93-103.

CHO, R. J., CAMPBELL, M. J., WINZELER, E. A., STEINMETZ, L., CONWAY, A., WODICKA, L., WOLFSBERG, T. G., GABRIELIAN, A. E., LANDSMAN, D. & LOCKHART, D. J. 1998. A Genome-wide Transcriptional Analysis of the Mitotic Cell Cycle. *Molecular Cell,* 2**,** 65-73.

COELHO, G. P., DE FRANÇA, F. O. & VON ZUBEN, F. J. 2008. A Multi-objective Multipopulation Approach for Biclustering. *Artificial Immune Systems.* Springer.

DE AMORIM, R. C. & HENNIG, C. 2015. Recovering the Number of Clusters in Data Sets with Noise Features using Feature Rescaling Factors. *Information Sciences,* 324, 126-145.

DE CASTRO, L. N. & TIMMIS, J. 2002. *Artificial Immune Systems: A New Computational Intelligence Approach*, Springer Science & Business Media.

DE JONG, K. A. 2006. *Evolutionary Computation: A Unified Approach,* Cambridge, Massachusetts, London, England, MIT Press.

DEB, K., PRATAP, A., AGARWAL, S. & MEYARIVAN, T. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation,* 6**,** 182-197.

DENITTO, M., BICEGO, M., FARINELLI, A. & FIGUEIREDO, M. A. T. 2017a. Spike and Slab Biclustering. *Pattern Recognition,* 72**,** 186-195.

DENITTO, M., FARINELLI, A., FIGUEIREDO, M. A. & BICEGO, M. 2017b. A Biclustering Approach based on Factor Graphs and the Max-sum Algorithm. *Pattern Recognition,* 62**,** 114-124.

DIVINA, F. & AGUILAR RUIZ, J. S. 2006. Biclustering of Expression Data with Evolutionary Computation. *IEEE Transactions on Knowledge and Data Engineering,* 18**,** 590-602.

DU, D., LI, K., LI, X. & FEI, M. 2014. A Novel Forward Gene Selection Algorithm for Microarray Data. *Neurocomputing,* 133**,** 446-458.

ERIC, C. C., GENEVERA, A. I. & RICHAD, B. G. 2017. Convex Biclustering. *Biometrics Journal of the International Biometric Society,* 73**,** 10-19.

FAN, J., HAN, F. & LIU, H. 2014. Challenges of Big Data Analysis. *National Science Review,* 1**,** 293-314.

FEI-FEI, L. & PERONA, P. A Bayesian Hierarchical Model for Learning Natural Scene Categories. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005. IEEE, 524-531.

GAN, X., LIEW, A. W. C. & YAN, H. 2008. Discovering Biclusters in Gene Expression Data based on High-dimensional Linear Geometries. *BMC Bioinformatics,* 9**,** 209-223.

GAN, X. C., LIEW, A. W. C. & YAN, H. Biclustering Gene Expression Data based on a High Dimensional Geometric Method. Proceedings of 2005 International Conference on Machine Learning and Cybernetics, 2005. IEEE, 3388-3393.

GANDER, W. & HREBICEK, J. 2011. *Solving Problems in Scientific Computing using Maple and Matlab®*, Springer Science & Business Media.

GOLCHIN, M., DAVARPANAH, S. H. & LIEW, A. W. C. Biclustering Analysis of Gene Expression Data using Multi-objective Evolutionary Algorithms. Proceeding of the 2015 International Conference on Machine Learning and Cybernetics 2015 Guangzhou, China. IEEE, 505-510.

GOLCHIN, M. & LIEW, A. W. C. Bicluster Detection using Strength Pareto Front Evolutionary Algorithm. Proceedings of the Australasian Computer Science Week Multiconference, 2016 Canberra, Australia. ACM, 1-6.

GOLCHIN, M. & LIEW, A. W. C. 2017. Parallel Biclustering Detection using Strength Pareto Front Evolutionary Algorithm. *Information Sciences,* 415-416**,** 283-297.

GOLCHIN, M. & LIEW, A. W. C. 2018. Geometric Biclustering by Hyperplane Projection and Multi-objective Evolutionary Algorithm. *Pattern Recognition.*

HAN, J., PEI, J. & KAMBER, M. 2011. *Data Mining: Concepts and Techniques*, Elsevier.

HARTIGAN, J. A. 1972. Direct Clustering of a Data Matrix. *Journal of the American Statistical Association,* 67**,** 123-129.

HOCHREITER, S., BODENHOFER, U., HEUSEL, M., MAYR, A., MITTERECKER, A., KASIM, A., KHAMIAKOVA, T., VAN SANDEN, S., LIN, D. & TALLOEN, W. 2010. FABIA: Factor Analysis for Bicluster Acquisition. *Bioinformatics,* 26**,** 1520-1527.

HOLMES, M. P., GRAY, A. & ISBELL, C. L. 2007. Fast SVD for Large-scale Matrices. *Workshop on Efficient Machine Learning at NIPS.*

HORN, J., NAFPLIOTIS, N. & GOLDBERG, D. E. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, 1994. IEEE, 82-87.

HOUGH, P. V. Machine Analysis of Bubble Chamber Pictures. 2nd International Conference on High-energy Accelerators and Instrumentation (HEACC), 1959 CERN, Geneva, Switzerland. 554-558.

HOUGH, P. V. 1962. *Method and Means for Recognizing Complex Patterns*. United States patent application.

JULIEN, S., IVETIC, A., GRIGORIADIS, A., QIZE, D., BURFORD, B., SPROVIERO, D., PICCO, G., GILLETT, C., PAPP, S. L. & SCHAFFER, L. 2011. Selectin Ligand Sialyl-lewis X Antigen Drives Metastasis of Hormone-dependent Breast Cancers. *Cancer Research,* 71**,** 7683-7693.

KANEHISA, M. & GOTO, S. 2000. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research,* 28**,** 27-30.

KLÉMA, J., MALINKA, F. & ŽELEZNÝ, F. 2017. Semantic Biclustering for Finding Local, Interpretable and Predictive Expression Patterns. *BMC Genomics,* 18**,** 41-53.

KLUGER, Y., BASRI, R., CHANG, J. T. & GERSTEIN, M. 2003. Spectral Biclustering of Microarray Data: Coclustering Genes and Conditions. *Genome Research,* 13**,** 703-716.

KONAK, A., COIT, D. W. & SMITH, A. E. 2006. Multi-objective Optimization using Genetic Algorithms: A Tutorial. *Reliability Engineering & System Safety,* 91**,** 992-1007.

KRIEGEL, H. P., KRÖGER, P., SANDER, J. & ZIMEK, A. 2011. Density-based Clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* 1**,** 231-240.

LAZEBNIK, S., SCHMID, C. & PONCE, J. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CPRV), 2006 New York, United States. IEEE, 2169-2178.

LEBER, J. H., BERNALES, S. & WALTER, P. 2004. IRE1-independent Gain Control of the Unfolded Protein Response. *PLOS Biology,* 2**,** e235.

LESKOVEC, J. & MCAULEY, J. J. Learning to Discover Social Circles in Ego Networks. Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS), 2012 Lake Tahoe, Nevada. 539-547.

LIEW, A. W. C. 2016. Biclustering Analysis of Gene Expression Data using Evolutionary Algorithms *In:* IBA, H. & NOMAN, N. (eds.) *Evolutionary*

Computation in Gene Regulatory Network Research. Hoboken, NJ, USA: John Wiley & Sons Inc.

LIU, B., YU, C. W., WANG, D. Z., CHEUNG, R. C. & YAN, H. 2014. Design Exploration of Geometric Biclustering for Microarray Data Analysis in Data Mining. *IEEE Transaction on Parallel and Distributed Systems,* 25**,** 2540 - 2550.

LIU, J. & CHEN, Y. Dynamic Biclustering of Microarray Data with MOPSO. IEEE International Conference on Granular Computing (GrC), 2010. IEEE, 330-334.

LIU, J., LI, Z. & CHEN, Y. Microarray Data Biclustering with Multi-objective Immune Optimization Algorithm. Fifth International Conference on Natural Computation (ICNC) 2009a. IEEE, 200-204.

LIU, J., LI, Z. & CHEN, Y. Microarray Data Biclustering with Multi-objective Immune Optimization Algorithm. Fifth International Conference on Natural Computation (ICNC'09), 2009b. IEEE, 200-204.

LIU, J., LI, Z., HU, X., CHEN, Y. & LIU, F. 2012. Multi-objective Dynamic Population Shuffled Frog-leaping Biclustering of Microarray Data. *BMC Genomics,* 13**,** S6.

LIU, J., LI, Z., HU, X., CHEN, Y. & PARK, E. K. 2011. Dynamic Biclustering of Microarray Data by Multi-objective Immune Optimization. *BMC Genomics,* 12**,** S11.

LIU, J., LI, Z., LIU, F. & CHEN, Y. Multi-Objective Particle Swarm Optimization Biclustering of Microarray Data. IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2008. IEEE, 363-366.

MACDONALD, T. J., BROWN, K. M., LAFLEUR, B., PETERSON, K., LAWLOR, C., CHEN, Y., PACKER, R. J., COGEN, P. & STEPHAN, D. A. 2001. Expression Profiling of Medulloblastoma: PDGFRA and the RAS/MAPK Pathway as Therapeutic Targets for Metastatic Disease. *Nature Genetics,* 29**,** 143-152.

MADEIRA, S. C. & OLIVEIRA, A. L. 2004. Biclustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* 1**,** 24-45.

MAHÉ, P., ARSAC, M., CHATELLIER, S., MONNIN, V., PERROT, N., MAILLER, S., GIRARD, V., RAMJEET, M., SURRE, J. & LACROIX, B. 2014. Automatic Identification of Mixed Bacterial Species Fingerprints in a MALDI-TOF Mass-spectrum. *Bioinformatics,* 30**,** 1280-1286.

MAULIK, U., MUKHOPADHYAY, A. & BANDYOPADHYAY, S. 2009. Finding Multiple Coherent Biclusters in Microarray Data using Variable String Length

Multiobjective Genetic Algorithm. *IEEE Transactions on Information Technology in Biomedicine,* 13**,** 969-975.

MAULIK, U., MUKHOPADHYAY, A., BHATTACHARYYA, M., KADERALI, L., BRORS, B., BANDYOPADHYAY, S. & EILS, R. 2013. Mining Quasi-Bicliques from HIV-1-Human Protein Interaction Network: A Multiobjective Biclustering Approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* 10**,** 423-435.

MIRKIN, B. G. E. 1996. *Mathematical Classification and Clustering,* Dordrecht Boston Kluwer Academic Publishers.

MISLOVE, A., VISWANATH, B., GUMMADI, K. P. & DRUSCHEL, P. You Are Who You Know: Inferring User Profiles in Online Social Networks. Proceedings of the Third ACM International Conference on Web Search and Data Mining, 2010. ACM, 251-260.

MITRA, S. & BANKA, H. 2006. Multi-objective Evolutionary Biclustering of Gene Expression Data. *Pattern Recognition,* 39**,** 2464-2477.

MUKHOPADHYAY, A., MAULIK, U., BANDYOPADHYAY, S. & COELLO, C. A. C. 2014a. A Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part I. *IEEE Transactions on Evolutionary Computation,* 18**,** 4-19.

MUKHOPADHYAY, A., MAULIK, U., BANDYOPADHYAY, S. & COELLO, C. A. C. 2014b. Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part II. *IEEE Transactions on Evolutionary Computation,* 18**,** 20-35.

MURALI, T. & KASIF, S. Extracting Conserved Gene Expression Motifs from Gene Expression Data. Proceedings of the Pacific Symposium on Biocomputing, 2003. 77-88.

NICKOLOFF, J. A. & HABER, J. E. 2001. *Mating-type Control of DNA Repair and Recombination in Saccharomyces cerevisiae,* Totowa, NJ, Humana Press.

NOGALES CADENAS, R., CARMONA SAEZ, P., VAZQUEZ, M., VICENTE, C., YANG, X., TIRADO, F., CARAZO, J. M. & PASCUAL MONTANO, A. 2009. GeneCodis: Interpreting Gene Lists through Enrichment Analysis and Integration of Diverse Biological Information. *Nucleic Acids Research,* 37**,** W317-W322.

OLIVA, A. & TORRALBA, A. 2001. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision,* 42**,** 145-175.

PADILHA, V. A. & CAMPELLO, R. J. G. B. 2017. A Systematic Comparative Evaluation of Biclustering Techniques. *BMC Bioinformatics,* 18**,** 55.

PONTES, B., GIRÁLDEZ, R. & AGUILAR-RUIZ, J. S. 2013. Configurable Pattern-based Evolutionary Biclustering of Gene Expression Data. *Algorithms for Molecular Biology,* 8**,** 4-26.

PONTES, B., GIRÁLDEZ, R. & AGUILAR RUIZ, J. S. 2015. Biclustering on Expression Data: A Review. *Journal of Biomedical Informatics,* 57**,** 163-180.

PRELIĆ, A., BLEULER, S., ZIMMERMANN, P., WILLE, A., BÜHLMANN, P., GRUISSEM, W., HENNIG, L., THIELE, L. & ZITZLER, E. 2006. A Systematic Comparison and Evaluation of Biclustering Methods for Gene Expression Data. *Bioinformatics,* 22**,** 1122-1129.

ROH, H. & PARK, S. A Novel Evolutionary Algorithm for Bi-clustering of Gene Expression Data based on the Order Preserving Sub-matrix (OPSM) Constraint. 8th IEEE International Conference on BioInformatics and BioEngineering (BIBE), 2008. IEEE, 1-14.

SELVARAJ, S. & NATARAJAN, J. 2011. Microarray Data Analysis and Mining Tools. *Bioinformation,* 6**,** 95-99.

SERIDI, K., JOURDAN, L. & TALBI, E.-G. Parallel Hybrid Metaheuristic for Multi-objective Biclustering in Microarray Data. 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012. IEEE 625-633.

SERIDI, K., JOURDAN, L. & TALBI, E. G. Multi-objective Evolutionary Algorithm for Biclustering in Microarrays Data. IEEE Congress on Evolutionary Computation (CEC), 2011. IEEE, 2593-2599.

SERIDI, K., JOURDAN, L. & TALBI, E. G. 2015. Using Multiobjective Optimization for Biclustering Microarray Data. *Applied Soft Computing,* 33**,** 239-249.

SHABALIN, A. A., WEIGMAN, V. J., PEROU, C. M. & NOBEL, A. B. 2009. Finding Large Average Submatrices in High Dimensional Data. *The Annals of Applied Statistics***,** 985-1012.

SÖDERKVIST, I. 1993. Perturbation Analysis of the Orthogonal Procrustes Problem. *BIT Numerical Mathematics,* 33**,** 687-694.

SON, C. G., BILKE, S., DAVIS, S., GREER, B. T., WEI, J. S., WHITEFORD, C. C., CHEN, Q.-R., CENACCHI, N. & KHAN, J. 2005. Database of mRNA Gene Expression Profiles of Multiple Human Organs. *Genome Research,* 15**,** 443-450.

SZETO, L. K., LIEW, A. W. C., YAN, H. & TANG, S. S. 2003. Gene Expression Data Clustering and Visualization based on a Binary Hierarchical Clustering Framework. *Journal of Visual Languages & Computing,* 14, 341-362.

TABAS MADRID, D., NOGALES CADENAS, R. & PASCUAL MONTANO, A. 2012. GeneCodis3: A Non-redundant and Modular Enrichment Analysis Tool for Functional Genomics. *Nucleic Acids Research,* 40**,** W478-W483.

TARCA, A. L., ROMERO, R. & DRAGHICI, S. 2006. Analysis of Microarray Experiments of Gene Expression Profiling. *American Journal of Obstetrics and Gynecology,* 195**,** 373-388.

TO, C. & LIEW, A. W. C. Genetic Algorithm Based Detection of General Linear Biclusters. International Conference on Machine Learning and Cybernetics (ICMLC), 2014 Lanzhou, China. IEEE, 550-555.

TOMASI, C. 2013. Orthogonal Matrices and the Singular Value Decomposition. Available: https://www.cs.duke.edu/courses/fall13/compsci527/notes/svd.pdf [Accessed 17/01/2017].

TUNG, A. K., XU, X. & OOI, B. C. Curler: Finding and Visualizing Nonlinear Correlation Custers. Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, 2005. ACM, 467-478.

URSEM, R. K. Multinational Evolutionary Algorithms. Proceedings of the 1999 Congress on Evolutionary Computation (CEC). , 1999. IEEE, 1633-1640.

WALKER, L. J., ALDHOUS, M. C., DRUMMOND, H. E., SMITH, B. R. K., NIMMO, E. R., ARNOTT, I. D. R. & SATSANGI, J. 2004. Anti-saccharomyces Cerevisiae Antibodies (ASCA) in Crohn's Disease are Associated with Disease Severity but not NOD2/CARD15 Mutations. *Clinical & Experimental Immunology,* 135**,** 490-496.

WANG, D. Z. & YAN, H. Geometric Biclustering Analysis of DNA Microarray Data based on Hypergraph Partitioning. IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW), 2010. 246-251.

WANG, D. Z. & YAN, H. 2013. A Graph Spectrum Based Geometric Biclustering Algorithm. *Journal of Theoretical Biology,* 317**,** 200-211.

WANG, Z., YU, C. W., CHEUNG, R. C. & YAN, H. 2012. Hypergraph Based Geometric Biclustering Algorithm. *Pattern Recognition Letters,* 33**,** 1656-1665.

XU, L. & OJA, E. 1993. Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities. CVGIP: Image Understanding, 57, 131-154.

YIP, K. Y., CHEUNG, D. W. & NG, M. K. 2004. Harp: A Practical Projected Clustering Algorithm. *IEEE Transactions on Knowledge and Data Engineering,* 16**,** 1387-1397.

ZHAO, H., LIEW, A. W. C., WANG, D. Z. & YAN, H. 2012. Biclustering Analysis for Pattern Discovery: Current Techniques, Comparative Studies and Applications. *Current Bioinformatics,* 7**,** 43-55.

ZHAO, H., LIEW, A. W. C., XIE, X. & YAN, H. 2008. A New Geometric Biclustering Algorithm based on the Hough Transform for Analysis of Large-scale Microarray Data. *Journal of Theoretical Biology,* 251**,** 264-274.

ZHU, X., LUO, X. & XU, C. 2017. Editorial Learning for Multimodal Data. *Neurocomputing,* 253**,** 1-5.

ZITZLER, E., LAUMANNS, M. & THIELE, L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Proceedings of the Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN), 2001 Athens. Greece. Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK).