Cluster Analysis of Transcriptomics Data to Identify Potential Cancer Biomarkers

A thesis submitted in partial fulfillment of the requirements for award of the degree of

Doctor of Philosophy

Koyel Mandal

Registration No. TZ155842 of 2015

Department of Computer Science and Engineering

School of Engineering, Tezpur University

Tezpur, Assam, India-784028

July, 2021

Dedicated

to my inspiring parents, beloved

sisters, and family

Abstract

With the rapid technological advancement in the field of genomics, proteomics, and transcriptomics, there is a tremendous surge in the volume of biological data such as genes, proteins, and transcriptomics which have been growing at an exponential rate. Extracting hidden biologically meaningful information from the massive amount of accumulated omics data essentially requires high-end computational methods. One such method is clustering which helps in investigating the putative functionally related groups of genes, understanding the activity of genes, and exhibiting inherent correlations across all the conditions.

It is believed that the alteration of genes is the principal cause of cancer. Moreover, the alteration of expression values may lead to dysregulation of biological pathways and may, in turn, lead to the growth of malignant cells causing cancer. On the other hand, abnormally expressed miRNAs also play a very critical role in various diseases such as cancer. This thesis extensively presents two computational studies viz, full-space and subspace clustering of transcriptomics data considering cancer microarray gene expression and miRNA expression datasets. The evaluation of all cluster results and identification of potential cancer biomarkers have been carried out in a manner that would aid in better cancer management in future. Our thesis contribution has also fanned out to include the development of two biomarker identification methods viz. frequency-based and network-based methods which determines potential biomarkers from the identified clusters.

The first study focuses on developing an unsupervised full-space clustering named Graph Attraction Clustering (GAClust) algorithm. Next, we propose two semi-supervised full-space clustering algorithms, i.e., Semi-supervised Density-based Clustering (SDC) and Semi-supervised Graph Attraction Clustering (SGAClust) guided by external biological knowledge, Gene Ontology (GO), in order to get good quality clusters. We apply clustering algorithms to synthetic and cancer microarray gene expression datasets in order to validate the clustering results.

We have extended the work by identifying some potential cancer biomarkers using a network-based method. We successfully prove that the semi-supervised algorithm significantly produces better quality clusters than the unsupervised algorithm.

Although clustering algorithms have their own advantages, they suffer from several limitations. From a biological standpoint, genes may not always be related to all experimental conditions but might be related to a subset of conditions. Therefore, the inefficiency of traditional clustering algorithms in extracting local structures inherent in the data due to their focus on finding global patterns has given rise to a new class of subspace clustering viz. biclustering algorithms. The second and the third work of the thesis present two proposed biclustering algorithms: an unsupervised Order-Preserving Biclustering (OPBic) and a semi-supervised Pathway-based Order-Preserving Biclustering (POPBic). We examine the strength of both the algorithms for synthetic, microarray gene expression, and miRNA expression datasets. Some candidate genes and miRNAs are identified as potential biomarkers using both frequency and network-based identification methods which are later validated to be responsible for several cancer types.

Our fourth study is concerned with the identification of genes that are co-expressed under a subset of samples across time points from Gene Sample Time (GST) data. Here, the biclustering algorithm fails since the data is of 3D data type. Therefore, we need to move ahead from biclustering to 3D subspace clustering or triclustering which can effectively handle the 3D gene expression data to fully understand the hidden biological knowledge. To this end, we propose a semi-supervised Pathway-based Order-Preserving Triclustering (POPTric) algorithm to analyze breast cancer GST data. We have investigated the performance of our triclustering algorithm with respect to synthetic and real datasets. Later, we identify hub genes from the resulting triclusters and try to establish them as potential biomarkers.

To sum up, all of our four contributions have unfolded from the two distinct branches of full-space and subspace clustering and focus on the very promising results obtained by proposed algorithms that will aid in cancer management.

Keywords: *Gene expression data, Gene Sample Time data, miRNA expression data, Cancer disease, Semi-supervised clustering, Gene Ontology, Pathway, Full-space clustering, Biclustering, Triclustering, Biomarker identification, Enrichment analysis, Order-preserving.*

Declaration

I, Koyel Mandal, hereby declare that the thesis entitled ***Cluster Analysis of Transcriptomics Data to Identify Potential Cancer Biomarkers*** submitted to the Department of Computer Science and Engineering under the School of Engineering, Tezpur University, in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy is based on bona fide work carried out by me. The results embodied in this thesis have not been submitted in part or in full, to any other university or institute for award of any degree or diploma.

(**Koyel Mandal**)

Tezpur University

Certificate

This is to certify that the thesis entitled ***Cluster Analysis of Transcriptomics Data to Identify Potential Cancer Biomarkers*** submitted to the School of Engineering Tezpur University in partial fulfillment for the award of the degree of **Doctor of Philosophy** in **Computer Science and Engineering** is a record of research work carried out by **Ms. Koyel Mandal** under my supervision and guidance.

All help received by her from various sources have been duly acknowledged.

No part of this thesis has been submitted elsewhere for award of any other degree.

Signature of Supervisor
(Dr. Rosy Sarmah)
Designation: Assistant Professor
School: Engineering
Department: Computer Science and Engineering

Tezpur University

Certificate

This is to certify that the thesis entitled ***Cluster Analysis of Transcriptomics Data to Identify Potential Cancer Biomarkers*** submitted to the School of Engineering Tezpur University in partial fulfillment for the award of the degree of **Doctor of Philosophy** in **Computer Science and Engineering** is a record of research work carried out by **Ms. Koyel Mandal** under my supervision and guidance.

All help received by her from various sources have been duly acknowledged.

No part of this thesis has been submitted elsewhere for award of any other degree.

Signature of Co-Supervisor
(Dr. Bhogeswar Borah)
Designation: Professor
Department of Computer Science and Engineering
School of Engineering, Tezpur University
Tezpur, Assam, India-784028

Acknowledgment

It gives me immense pleasure to express my due regards to all those who contributed directly or indirectly to the successful completion of my doctoral thesis. The entire journey would not have been possible without the supportive hands of my supervisors, family, and friends towards me. I am especially grateful for all the amazing opportunities that Tezpur University has provided to me during my doctoral program.

There are no words to express my sincere gratitude to my esteemed supervisor Dr. Rosy Sarmah, for her untiring support, encouragement, and invaluable guidance to make this herculean task a success. I would like to thank her for her continuous support, constructive criticism, and innumerable advice. Her constant motivation, inspiration, and dedication brought the best out of me. Indeed, her expertise and in-depth knowledge have sharpened my skills. I must thank my co-supervisor as well as the Head of the Department, Prof. Bhogeswar Borah for his valuable feedback, time, and energy. I thank both of them for giving me the freedom to explore my ideas to pursue this work at my own pace.

I would like to acknowledge my heartfelt thanks to the Doctoral Committee and Departmental Research Committee members of my research. I owe a debt of gratitude to Prof. Dhruba Kr. Bhattacharyya for his careful attention, valuable comments, time, and sharing of knowledge throughout this journey. I am deeply indebted to Professor Dilip Kr. Saikia, Professor Smriti Kumar Sinha, Professor Sarat Saharia, Professor Utpal Sharma, Professor Nityananda Sarma, Dr. Bhabesh Nath, and all other faculty members of the Department of Computer Science and Engineering, Tezpur University for their valuable feedback and insightful comments during the review meetings. I would like to extend my warmest thanks to Dr. Arindam Karmakar for his great teaching, priceless insight, useful information, fruitful discussion, and treasured support towards

the completion of my thesis. I wish to put on record my deepest appreciation to my thesis review committee for their precious time, constructive comments, important feedback, and suggestions.

A special thanks to the technical and office staff members especially Pranati ba and Golap da of the Department of Computer Science and Engineering for their generous help in multiple ways.

I am deeply thankful to my parents Mr. Narayan Chandra Mandal and Mrs. Kalpana Mandal for their encouragement, love, moral support, and cooperation in every possible way to follow my dreams. Thank you, for believing in me, as always and inspiring me to understand myself, to be happy, and to know others. I am fortunate enough to have the blessings and love of my grandmother. These few words are not sufficient to describe how thankful I am. My heartfelt thanks goes to my elder sister Mrs. Kakali Sarker and my younger sister Ms. Payel Mandal for their emotional support, encouragement, and unconditional love. To my nephew Aritra Sarker and niece Aratrika Sarker, you are the source of my happiness and joy. I love you all from the bottom of my heart. I certainly do not forget my uncle Mr. Biswajit Roy for his unending help. I also thank my aunt Mrs. Susoma Roy, my two cousins: Writtika, Abantika, and brother-in-law for their love and support.

I am extremely grateful to all my seniors and research scholars specifically Priyakshi di, Nazrul da, Ram da, Anindita, Sagarika, Parthajit, Upasana, Nilakshi, Hussain, Piyali, Prakash, Shafiul, Alexy da, and Nirmal da for their endless support and help in various ways. My appreciation goes to Nazreena for the cherished time spent in the department. A special thanks to Jamil and Barnali for their unforgettable help in administrative work, and unwavering support. Additionally, I would like to thank Reema for her moral support and clarifying discussions. To my friends and hostel-mates, Ananya, Elizabeth, and Lopa thank you for your patience in listening to me, suggesting advice, and supporting me throughout this entire process. The dinner, debates, evening tea, late night singing of Tagore's song, and cooking together at Dr. Karmakar's place were all greatly appreciated. This journey was a memorable one for me. Reshmi, Sujoy, Nandan, Sayak, Amit, Arghya, and to the friends scattered around the corner, thank you for being in my life with your well-wishes, phone calls, text messages, e-mails, advice, and thoughts. Last, but not least I want to thank Almighty for his blessings to complete this work.

Contents

List of Figures

List of Tables

List of Algorithms

Glossary of Terms

ALCS	All Substrings Common Subsequence
ANOVA	Analysis of variance
asc_sort	Sorting in ascending order
BD	Bicluster Diffusion
BH	Ball-Hall
BiBit	Bit-Pattern Biclustering algorithm
BP	Biological Process
C&C	Cheng and Church
CAST	Cluster Affinity Search Technique
Cc	Condition coverage
CC	Cellular Component
cDNA	Complementary DNA
CI	C index
CLICK	CLuster Identification via Connectivity Kernels
CS	Confirmation Score
DAC	Divide and Conquer
DAVID	Database for Annotation, Visualization and Integrated Discovery
DB	Davies Bouldin
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DNA	Deoxyribonucleic Acid
DPI	Distribution parameter identification
FABIA	Factor Analysis for BIcluster Acquisition
GAClust	Graph Attraction Clustering
GB	Grap-based
Gc	Gene coverage

GO	Gene Ontology
GIS	Greedy iterative search
GST	Gene Sample Time
HC	Hierarchical clustering
IC	Information content
JC	Jaccard Coefficient (used in bicluster evaluation)
KEGG	Kyoto Encyclopedia of Genes and Genome
KNN	K-Nearest Neighbor
LCS	Longest Common Subsequence
M	Match
MM	Maximal Match
MMRP	Maximally Matched Regulation Pattern
mALCS	Modified All Substrings Common Subsequence
MF	Molecular Function
miRNA	microRNA
mRNA	Messenger RNA
MS	Match score
MSE	Mean Squared Error
MSR	Mean Square Residue (MSR)
NGS	Next Generation Sequencing
NMMRP	Negative Maximally Matched Regulation Pattern
NP	Nondeterministic Polynomial
OPBic	Order-Preserving Biclustering
OPPM	Order-Preserving Pattern Matching
OPSM	Order-Preserving Sub-Matrices
PCC	Pearson Correlation Coefficient
PGL	Pathway Gene List
POPBic	Pathway-based Order-Preserving Biclustering
POPTric	Pathway-based Order-Preserving Triclustering

pri-miRNA	Primary miRNA
QUBIC	QUalitative BIClustering
RNA	Ribonucleic Acid
RNA-seq	RNA sequencing
SAMBA	Statistical-Algorithmic Method for Bicluster Analysis
SDC	Semi-supervised Density-based Clustering
SGAClust	Semi-supervised Graph Attraction Clustering
SGL	Significant Gene List
SSGL	Significant Seed Gene List
SOTA	Self Organising Tree Algorithm
SS	Semantic Similarity
TD	Tricluster Diffusion
Vol	Volume in tricluster

Symbols and Notations

$ED_{m\times n}$	Gene expression data matrix with m number of rows and n number of columns
$G = \{g_1, g_2, \ldots, g_m\}$	A set of m number of genes
$C = \{c_1, c_2, \ldots, c_n\}$	A set of n number of experimental conditions or samples in $ED_{m\times n}$
ge_{ij}	Expression value of a gene g_i under a specific condition c_j
$Dist(x, y)$	(Dis)similarity between a pair of objects x and y
$\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$	A set of K number of clusters
$\mathcal{O}$	Centroid of cluster
$Sim_{x,y}$	Similarity matrix, where $S_{x,y}$ denotes the similarity between gene g_x and g_y
$\beta(\mathcal{I}, \mathcal{J})$	A bicluster corresponds to a subset of rows $\mathcal{I} \subseteq G$ under a subset of conditions $\mathcal{J} \subseteq C$
$\mathcal{D}_{G\times S\times T}$	Three dimensional gene expression or Gene Sample Time data
$S = \{s_1, s_2, \ldots, s_n\}$	n number of samples or experimental conditions of GST data
$T = \{t_1, t_2, \ldots, t_v\}$	v number of time points of GST data
d_{xyz}	The expression level of x^{th} gene, y^{th} experimental condition at z^{th} time point
$\mathcal{T}(X, Y, Z)$	The submatrix $\mathcal{T}$ represents a subset of genes $X \subseteq G$ that are co-expressed under a subset of experimental conditions or samples $Y \subseteq S$ over a subset of time-points $Z \subseteq T$
$\mathbb{T}$	GO term
Υ	Neighborhood distance threshold
$\mathcal{G}^*(V, E)$	Weighted graph, where vertices V denote genes and E represent the edge set
$\mathcal{H}$	Clique graph

cq_i	Clique
$\mathcal{N}(g_x)$	Neighborhood of a gene g_x
$\mathcal{CN}(g_x, g_y)$	Common neighborhood between two genes g_x and g_y
$R_{m\times m}$	Similarity matrix of size $m \times m$ and R is the similarity between two genes g_x and g_y
$deg(g_x)$	Degree of a gene g_x
$\mathcal{A}$	Attraction
η	Attraction threshold for GAClust algorithm
$deg(g_x)$	Degree of a vertex V or g_x
μ	Mean
σ	Standard deviation
ED_{disct}	Discretized gene expression data
δ	Minimum matching threshold value (SDC algorithm)
ε-neighbor	ε-neighbors with respect to $g_i \in G$
Com_sim	Combined similarity
w_1, w_2	Weight factors of proximity (similarity) measure and semantic similarity measure, respectively
N_c	Core neighbors
M_p	Minimum points in a cluster
η'	Attraction threshold for SGAClust algorithm
Υ'	Neighborhood similarity threshold
φ	User-defined threshold for the number of biomarkers
ψ	User-defined threshold for the number of clusters
R_{min}	Minimum number of rows
C_{min}	Minimum number of conditions
W	Number of workers
O_{max}	Maximum overlap allowed

$\mathbb{O}$	Order
d	Different types of samples
$\mathbb{S}$	Speed-up
$\mathbb{E}$	Efficiency
$\mathcal{S}$	The number of substrings
R_L and R_U	Lower and upper bound for rows
C_L and C_U	Lower and upper bound for columns
$\mathcal{K}$	Number of mismatches
α	Significance level (POPBic and POPTric)
t_s	Seed selection criteria
ϵ	Maximum error (POPBic) or error tolerant threshold (POPTric)
G'	A gene list having maximum overlap scores with g_a
$Seed_{g_a}$	Seed gene
O_{score}	Overlap score
$P = \{p_1, p_2, \ldots, p_h\}$	A set of h number of KEEG pathways
$Reverse(pat)$	Opposite (positive or negative) pattern
OM	Order matrix
$\bar{S}$	The number of significant seed gene
$\bar{P}$	The number of Pathway Gene List
$\bar{G}$	The number of genes which has maximum overlap score with a seed gene
S_{min}	Minimum number of samples
T_{min}	Minimum number of time points
K	Number of clusters
$\bar{\delta}$	Merging threshold
θ	Overlapping threshold

1

Introduction

All living organisms are composed of one or collections of cells which are commonly known as structural units of life. Organisms are classified as Eukaryotes that contain nuclei whereas in Prokaryotes nuclei are absent. The most important component, chromosome, located in a nucleus[1] of a cell comprises of long chains of Deoxyribonucleic acid (DNA). A sequence of DNA carries hereditary information and contributes to function/phenotype can be related by the term gene. Cells share the same genes, but some genes are turned on or off for distinguishing the specific work of cell to cell. The complete genetic material is defined as genome. The pictorial representation of a cell is depicted in Figure 1.1.

Another important type of biological molecule is Ribonucleic Acid (RNA). DNA and RNA are made up of nucleotides which are composed of a five carbon sugar, four different nitrogenous base, and a phosphate group. The pentose sugar in DNA is deoxyribose whereas in RNA it is ribose. A DNA is a prime genetic molecule that is arranged with two polynucleotide strands that intertwining each other to shape a double helix structure [338]. Each strand of DNA has two ends which are $5'$ and $3'$, referred to as the number of carbon atoms of deoxyribose. These two strands of DNA run in an antiparallel direction

[1] a special membrane-bound organelle

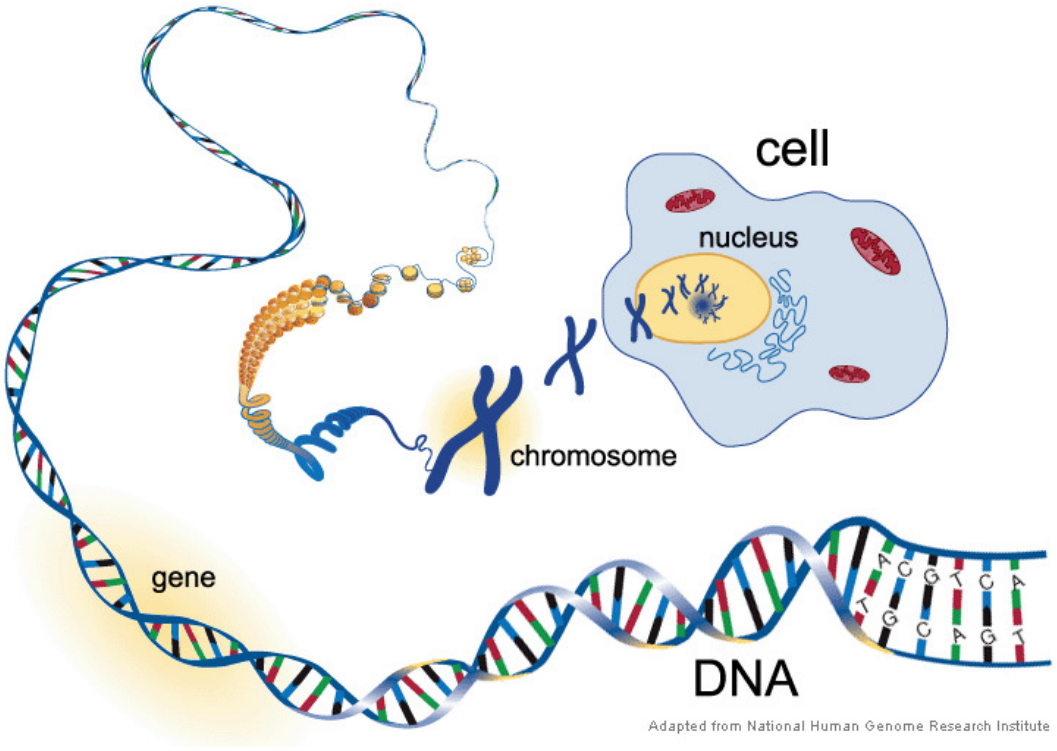


Figure 1.1: DNA is present in the chromosome of each cell.
Image credit: `https://www.tes.com/lessons/mGZszATpjnNdOw/chromosomes`

which means 5′ end of a strand is adjacent to 3′ end of the other strand. The different DNA nitrogenous bases are Adenine, Thymine, Guanine, and Cytosine. Nucleobase located in one strand of DNA structure is paired with complementary bases on another strand of DNA forming a base pair. For instance, Adenine binds with Thymine and Guanine joins with Cytosine by hydrogen bonds as shown in Figure 1.2. DNA is mainly responsible for the development, reproduction, and functioning. On the other hand, nucleic acid RNA is single-stranded which is mainly involved in regulation and protein synthesis. Nitrogenous bases of RNA are the same as DNA except for Thymine, which is replaced by Uracil. RNA is synthesized from DNA and DNA has the nature of self-replicating. The study of computational biology deals with analyzing a huge amount of biological data which solves many practical biological issues by developing algorithms based on computational and statistical methods.

1.1 Transcriptomics data

The main framework of molecular biology states that DNA sequences are transcribed into messenger RNA (mRNA) by the transcription process, followed by

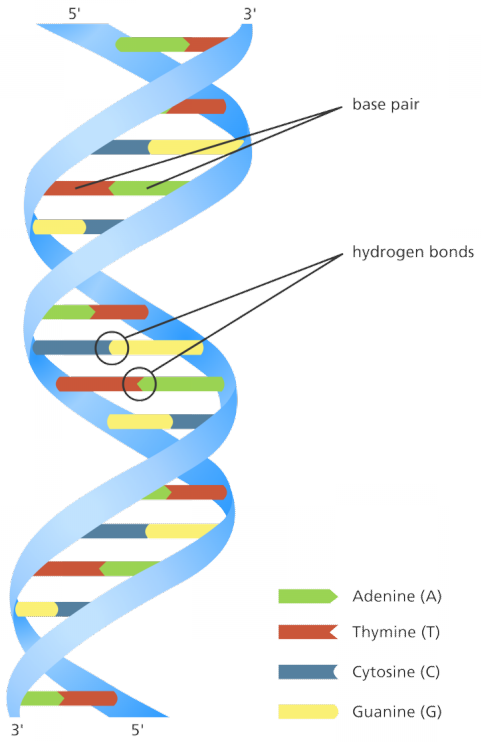


Figure 1.2: The structure of a DNA molecule.
Image credit: `https://www.yourgenome.org/facts/what-is-dna`

RNA splicing, and later translated into protein as shown in Figure 1.3. This biological phenomenon is termed as central dogma (proposed by Francis Crick in 1958) and consists of three major steps: i) replication: making a new copy of DNA from existing DNA, ii) transcription: making new RNA from DNA, and iii) translation: producing a protein that contains series of amino acid from RNA. During the transcription process, the RNA molecule is initially considered as precursor mRNA (pre-mRNA) which is an "immature" molecule. Later, pre-mRNA goes under several modifications such as capping, exclusion of introns[2], splicing of exons[3], and addition of a polyadenine (polyA) tail to become mature mRNA. Pre-mRNA includes both introns and exons. With the help of the RNA splicing process introns of the gene are removed and exons are joined together to form a coding sequence i.e., mature mRNA. RNA splicing is demonstrated in Figure 1.4. mRNA includes 5′ cap made up of 7 methyl guanosine and 3′ polyA tail, which further goes to the translation process. The newly formed mRNA is transported out from the nucleus into the cytoplasm to the ribosome which is known as the protein factory of the cell. The translation process is involved

[2]The region of RNA which is not used for protein code
[3]The region of RNA which is used for protein code

in decoding mRNA and using this information to build polypeptide (basically protein) or chain of amino acids. In mRNA, the polypeptide is built using the group of three nucleotides called a codon. The three basic parts of translation are i) initiation, ii) elongation, and iii) termination. To initiate the translation process ribosome assembles with first transfer RNA (tRNA) and mRNA. In the middle phase, tRNA helps in transferring the free amino acids from cytoplasm to ribosome and is linked together to form a polypeptide chain. Further, tRNAs continue to add more amino acids to the growing end of the chain until it reaches the stop codon. Towards the end of the translation process, the ribosome releases finished polypeptide or protein into the cell.

All synthesized RNA including protein coding (mRNA) and non-coding produced by genome under specific cell, tissue type, or in an organism is widely understood as the transcriptome. RNAs are exemplified by siRNA (short interfering RNA), rRNA (ribosome RNA), or lncRNA (long non-coding RNA) which are necessarily not involved in the translation process to produce a protein (non-coding) [45]. The study of transcriptome signifies transcriptomics which summarizes a global picture of how genes are expressed, interconnected, and functioning. It provides a comprehensive analysis of cells, tissues, or organisms under specific physiological conditions, time points, or development stages. Transcriptomics data or the entire set of RNA catalogs different transcripts, such as mRNA, microRNA (miRNA), and lncRNA.

1.1.1 Gene expression profiling

The central dogma is at the heart of molecular biology which represents the flow of information from DNA through RNA and finally into proteins. This is known as gene expression. The ways in which genes are expressed can affect the organisms phenotypes such as hair color, color of eyes etc. Gene expression profiling can reflect thousands of gene expressions simultaneously for a deep understanding of cellular function. Using modern high-throughput technologies, gene expression profiling quantifies the count of mRNA transcripts that in turn calculate the number of corresponding proteins at the transcription level. This originally means estimating relative mRNA amounts in different experimental conditions and after that assessing under which condition particular genes are expressed. If a gene is turned off, it is considered to be not used to produce mRNA and if a gene is turned on then the gene is used to make mRNA.

The data collected from gene expression profiling experiments is called gene expression data. A proper level of mRNA is an essential intermediate in

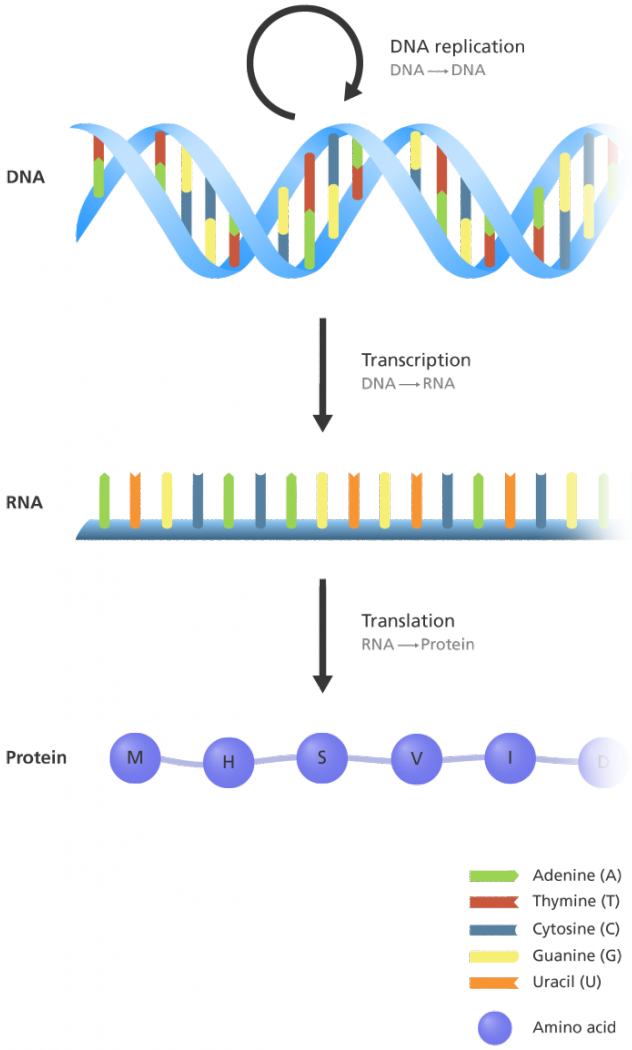


Figure 1.3: Central dogma explains the transfer of genetic information from DNA to protein.
Image credit: `https://www.yourgenome.org/facts/what-is-the-central-dogma`

the expression level of genes. Multi-dimensional high-throughput gene expression data can be organized in a matrix (2D) with row vectors representing the gene expression patterns and column vectors showing the expression profile of conditions, samples, or time points, as presented in Figure 1.5 [254]. The heatmap representation of gene expression data consists of six genes $\{g_1, g_2, g_3, g_4, g_5, g_6\}$, four columns $\{c_1, c_2, c_3, c_4\}$, and each cell represents expression value. The number of genes is significantly greater than that of the number of samples. The gene expression levels are measured in different experimental conditions, different medical conditions, different (diseased or healthy) sample tissue, the influence of drug application on sample tissue, development stages, different time points, or different organs.

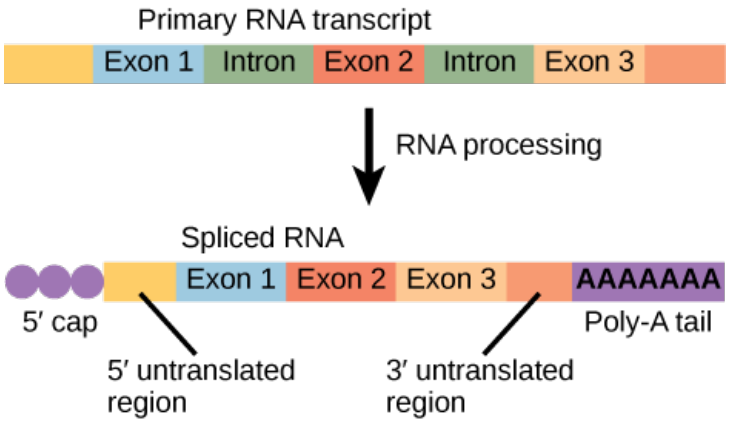


Figure 1.4: The process of a RNA splicing.
Image credit: `https://courses.lumenlearning.com/suny-osbiology2e/chapter/rna-processing-in-eukaryotes/#fig-ch15_03_02`

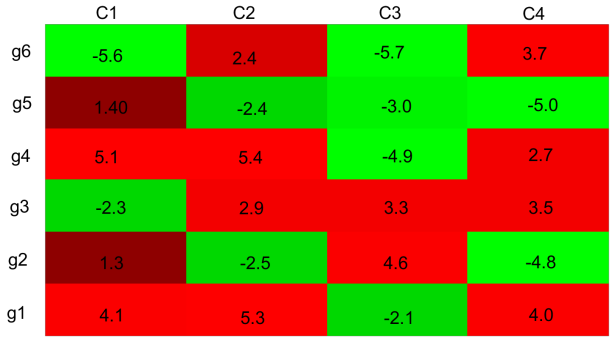


Figure 1.5: A toy example of a gene expression data matrix.

In recent years, due to rapid advancement in technology, it is now possible to collect gene expression values under a massive number of experimental conditions during various time points from a single experiment. This type of data is called 3D gene expression data or Gene Sample Time (GST) data [316]. Figure 1.6 presents a schematic diagram of GST data with six genes $\{g_1, g_2, g_3, g_4, g_5, g_6\}$, four samples $\{s_1, s_2, s_3, s_4\}$, and three time points $\{t_1, t_2, t_3\}$.

1.1.2 Introduction to microRNA

MicroRNAs (miRNA) are small non-coding RNA molecules of 19-25 nucleotides (nt) in length, that can regulate translation and regulation of various target genes [301]. More than 2500 miRNAs are referenced in microRNA database, named miRbase[4] [17]. miRNA is first discovered in the year 1993 by Lee et al. [191]. Since then, miRNAs have emerged as a key regulator in cellular

[4]http://www.mirbase.org/

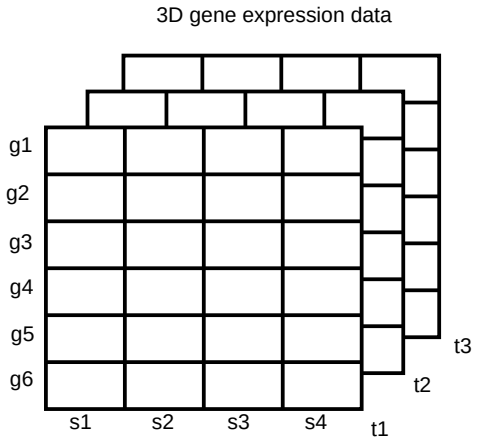


Figure 1.6: A schematic diagram of Gene Sample Time (GST) data.

functions, metabolic pathways, and physiological processes. It appears to be actively participating in a variety of biological processes such as cell cycle, cell growth, differentiation, apoptosis[5], and proliferation[6] through negative regulation of gene expression to a post-transcriptional level [123, 301]. In other words, miRNA performs a ubiquitous role to regulate mRNAs which are associated with protein translation and suppress gene expression.

The mechanism of miRNA function can greatly be understood by the process of biogenesis [232]. Biogenesis occurs in both the nucleus and cytoplasm through multiple steps. Initially, miRNAs are transcribed by RNA polymerase II as long, pri-miRNAs (primary miRNA transcripts) which are comprised of 5′ cap and polyA tail having more than 1000 nt [236]. In the nucleus, pri-miRNAs are cleaved into a hairpin-like structure called pre-miRNAs by the microprocessor complex comprising of RNase III enzyme Drosha [259]. Subsequently, pre-miRNAs are then exported to the cytoplasm by the karyopherin exportin 5 (Exp5) and Ran-GTP, where they undergo the final event by a second RNase III-like enzyme, Dicer [43]. Now, Dicer gives rise to duplexed miRNA strands. In general, one strand of miRNA which is destined to be loaded into RISC (RNA-induced silencing complex), takes active participation in gene regulation. This strand is known as 'guide' strand, represented by miR whereas another strand is termed as 'minor miRNA' or 'passenger miRNA', denoted by miR*. These strands are then included in the RISC loaded with an Argonaute (AGO) protein in order to select one strand to become mature miRNA. The complex causes the passenger strand to unwind from the guide strand via different mechanisms on the basis of degree complementarity and is discarded [260]. Thus mature RISC

[5]The process of cell self-destruction
[6]The process results increasing number of cells

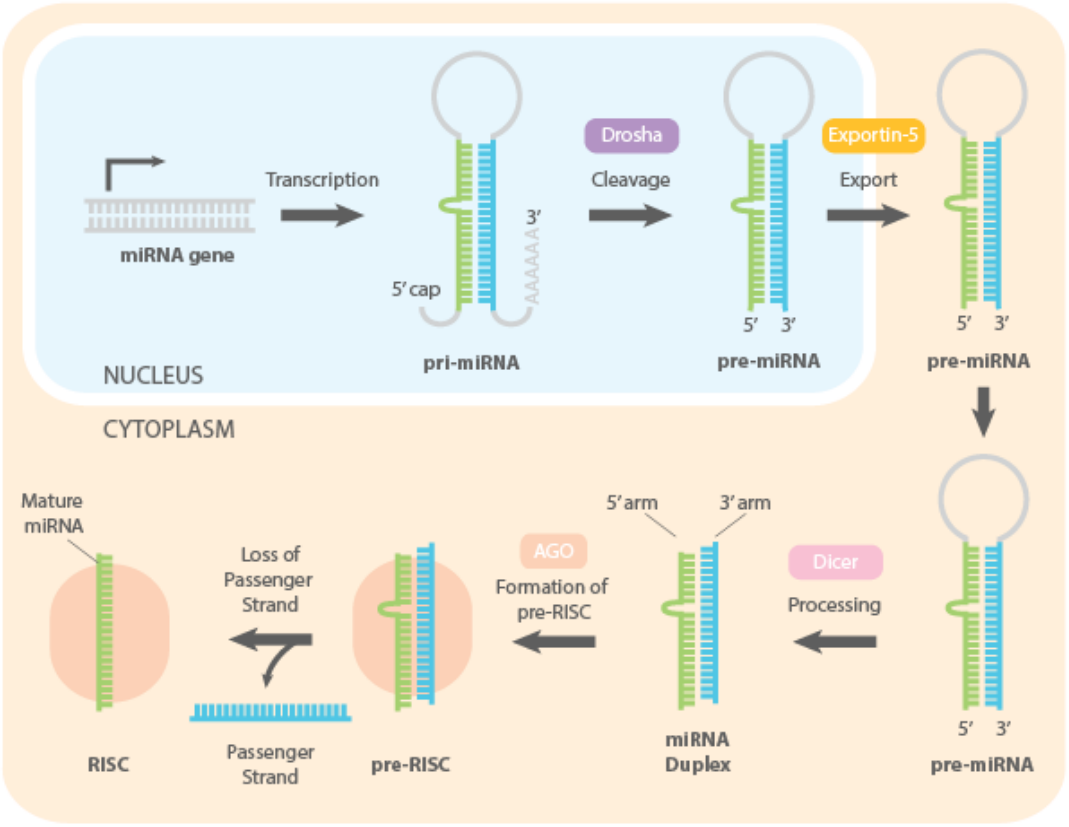


Figure 1.7: Mechanism of miRNA biogenesis.
Image credit: `https://old.abmgood.com/marketing/knowledge_base/miRNA_Introduction.php`

carries mature, single stranded miRNA which is basically a part of RISC and silences the expression of target genes. Moreover, RISC with the help of pre-miRNA act as the source of small miRNA and can cleave target mRNA [233]. The entire process can be elucidated from Figure 1.7. In addition to this, many other varieties have been discovered [259, 260].

1.2 Experimental techniques for transcriptome analysis

Experimental techniques to measure transcriptomics data can be broadly classified into two types viz, low and high-throughput depending on the amount of data they produce. Quantitative Reverse Transcription Polymerase Chain Reaction (qRT-PCR) and Northern Blot fall under low-throughput technology, allowing a limited number of transcripts for measurement with high specificity. In the past three decades, several breakthroughs in high-throughput technologies have made

it possible to analyze a massive amount of the expression of multiple transcripts in different pathological conditions for understanding the link between transcriptome and cellular phenotypes. Currently, two major well-established methods i.e., microarray (hybridization-based technique) and next-generation sequencing (sequence-based approach) are leading methods used to quantify the expression profiles of genes and miRNAs [271, 279]. Next, we discuss in detail these emerging methods in sequence.

1.2.1 Microarray technology

DNA microarray technology [95] is one of the best-known representatives in the field of bioinformatics, molecular biology, biomedical studies etc. Previously, studying the bioinformatics data such as genomics[7], proteomics[8] was just a dream. With the invention of DNA microarray technology, this dream was realized. This technology is considered to be extremely beneficial in drug discovery, drug identification, drug validation, pathological behaviors, cell development, and cell differentiation. Microarray has the power to monitor thousands of expression levels of different genes in a parallel fashion, under certain conditions like time series, drug application at different stages, diseased cells etc. Complementary DNA (cDNA) and Oligonucleotides are two types of array used in microarray experiments [164]. Regardless of which type of microarray is used the basic purpose is the same except for some differences in the experiment.

Different variants of this method can be found in the literature but the basic underlying idea is the same. This method is based on the hybridization concept which is based on spontaneously joining complementary base pairs. DNA Microarray (DNA chip) is typically made up of glass slides coated with chemical, silicon, or nylon membranes, where DNA's are spotted precisely in a sequential manner. Each spot can encompass a million copies of identical DNA. As a whole one microarray contains thousands of genes. At first two RNA samples (say, sample tissue are kept under condition A and same sample tissue in condition B) are reverse transcripted to get the cDNA. cDNA is then labeled with fluorescent dyes (red for condition A and green for condition B) to distinguish between two samples (diseased or healthy). The labeled cDNA is hybridized into a microarray slide. Now, the hybridized microarrays are being excited under the laser and scanned under a particular wavelength to determine the red and green dye. The emitted fluorescence determines the abundance of RNA (nucleic acid). Loosely

[7]The study of genome
[8]The study of proteins

speaking it generates an image comprised of red (produced cDNA in condition A is greater than produced cDNA in condition B), green (produced cDNA in condition B is greater than produced cDNA in condition A), yellow (gene expression level is the same in both A and B conditions), and black (no expression levels of the genes in both the conditions) spots which represents the expression level of each gene. The transforming of this red-green false color image into a gene expression matrix is rendered using image processing techniques. The overview of microarray technology for gene expression data is depicted in Figure 1.8. Of note, Affymetrix microarray or Oligonucleotides measure expression level in absolute terms whereas cDNA measures in relative terms.

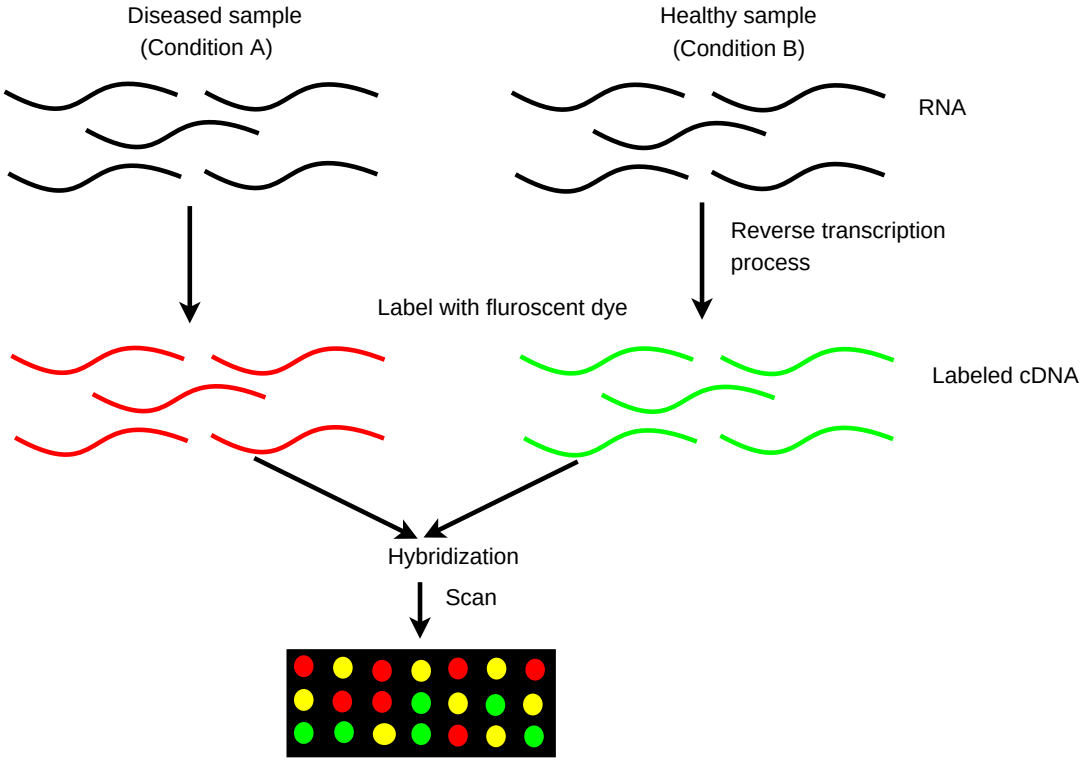


Figure 1.8: A brief overview of microarray technology for gene expression data.

Microarray technology has been facilitated for miRNA expression profiles also. This is the oldest method to be used for simultaneous analysis of miRNA expression profiles. A similar approach as mentioned before is being performed to analyze miRNAs in parallel at the time of profiling. The steps involved in this method are fluorescent labeling of miRNAs, using DNA-based probes for hybridization, washing, and scanning of the array and finally, quantification of fluorescence to profile the miRNAs [279]. Several variations have been developed including DNA-based probe to fluorescent tagging methods of miRNA in different samples for executing hybridization method [279].

1.2.2 Next generation sequencing techniques

Notwithstanding the success of microarray technology, it suffers from some disadvantages. Prior knowledge is needed on species or transcript specified probes, limitation of accurate measurement of expression due to background hybridization and consideration of relative values instead of absolute are some of the deficiencies of microarrays [109]. More recently, RNA sequencing (RNA-seq) technology is quickly superseding microarray technology. RNA-seq data uses the emergence of the NGS (Next Generation Sequencing) platform which reveals the quantity of RNA and generates DNA sequence from the RNA reference molecule. In a simplified way, it provides qualitative and quantitative information about RNA present in different samples. RNA-seq is not limited to pre-specified genes to be assayed rather it can detect new transcripts [42]. Additionally, the sequencing of the genetic material resulting from different biological samples is the core dependency of RNA-seq. A vast range of NGS technologies has been developed such as Illumina Genome Analyzer and SOLiD (Sequencing by Oligonucleotide Ligation and Detection) for different biological fields to examine genome-wide expression level of genes including miRNAs [210]. RNA-seq procedure is far more complex than microarray. The fundamental steps of RNA-seq are summarized below.

In the very first step, RNA molecules from the samples of interest are collected and then transcribed into cDNA fragments allowing them to enter into the NGS workflow. To perform the sequencing on genetic material, adapters, or short constant sequences are added to each of the fragmented molecules. These adapters permit sequencing which carries functional elements such as primary sequencing site and amplification element. Next, cDNA is analyzed by NGS, which generates short sequences considering both or one end of the fragments. At the end of this sequencing technology, each sequence for the biological sample provides a discrete expression known as count. A detailed review of different NGS technology and RNA-seq can be found in [235, 336].

1.3 Transcriptomics data analysis

Research in computational biology has been revolutionized with the advent of high-throughput technologies which has exponentially increased the amount of data. Extracting biologically hidden predictive information from a large amount of data delivers a great challenging task. To understand the complex mechanisms of biological phenomena at the molecular level, it is necessary to identify groups

of genes interacting with each other to accomplish a biological function [12]. Investigation of these data should be done using computational methods. One of the promising areas of data mining techniques, clustering is used in various fields such as object recognition, speech processing, text mining, and image processing [58, 100]. Clustering attempts to identify a group (cluster) of objects so that objects within the same group share a similar pattern while being dissimilar from the objects in other groups.

The application of clustering approaches in gene expression data is wide ranging from functional genomic application [315], biotechnological research to clinical applications. It has been proved to be extremely valuable in health care applications like disease diagnosis, drug identification, accurate treatment procedure, insights of different diseases, and identification of subtypes of diseases to name a few. Clustering helps in gaining a deep understanding of cell regulations, functions of an uncharacterized gene, subtype of cells, biological pathways, and cellular processes [164, 174]. It also helps to understand the genetic behavior of life, gene function prediction, and regulatory motifs from gene expression data [86].

Clustering can be broadly classified into two types, (i) full-space and (ii) subspace clustering. Subspace clustering is an NP-hard problem which in turn makes the algorithm computationally expensive compared to the classical one [314]. The reason behind the complexity of the algorithm is the checking of all possible subsets of a given dataset rather than just looking at the dataset once and finding the clusters. Biclustering and triclustering fall under subspace clustering algorithms. We start by introducing the basic concepts of all these algorithms in the next section.

1.3.1 Full-space cluster analysis

Full-space clustering essentially means a grouping of genes (or conditions) based on their proximity (similarity or dissimilarity) utilizing the entire dimension of conditions (or genes). The prior job of a full-space clustering algorithm is to identify co-expressed genes which in turn reveals the information about unknown genes by forming clusters with some known genes [37]. Functionally related co-expressed genes in a cluster also reveal how various biological pathways and processes respond to the underlying biological systems [100]. Figure 1.9 depicts a scenario of a full-space clustering algorithm where the left side shows the heatmap of gene expression data and on the right side shows the corresponding three gene clusters associated with co-expressed patterns in graphs.

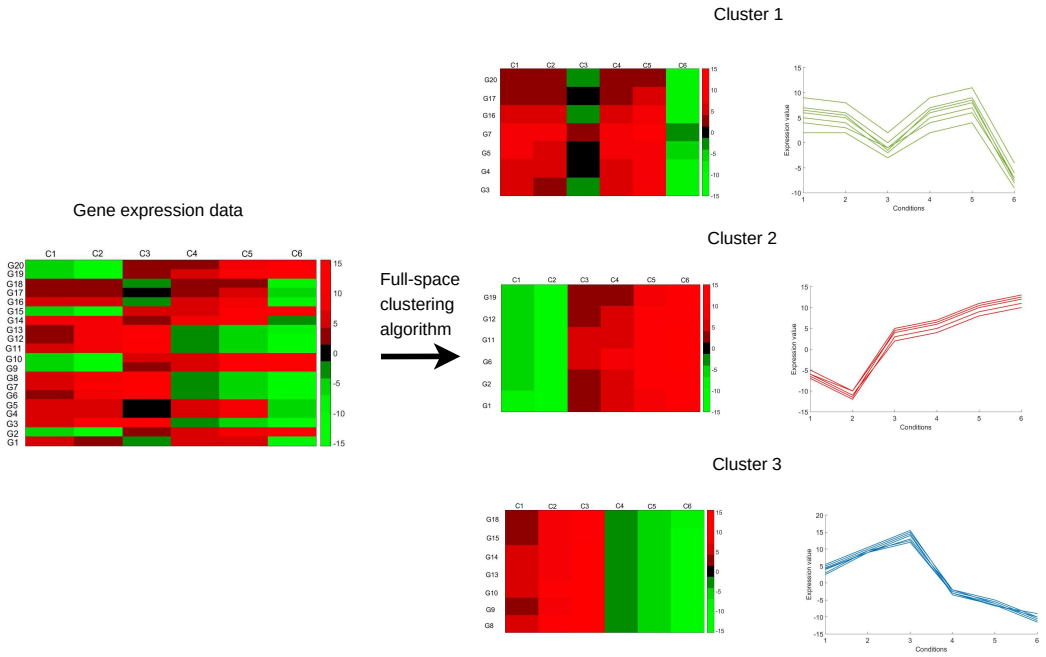


Figure 1.9: The input and output of a full-space clustering algorithm.

1.3.2 Bicluster analysis

From a biological point of view, an important fact is that genes are not really related to all experimental conditions but to a subset of conditions [109, 277]. At this point, traditional clustering algorithms fail to identify a subset of genes under a subset of conditions because it considers the whole set of conditions. Additionally, a recent perception is that genes are co-regulated and co-expressed under certain experimental conditions and behaves almost independently for the remaining conditions [227]. This conception has given birth to biclustering algorithms and outperforms traditional clustering algorithms [99, 278]. Biclustering which is a type of subspace clustering simultaneously clusters both rows and columns. Formally, a biclustering algorithm groups a subset of genes in association with a subset of samples, which convey biologically more significant groups of genes. The different nomenclature of biclustering algorithms includes subspace clustering, bi-dimensional clustering, co-clustering, simultaneous clustering, and two-way clustering. Figure 1.10 shows the input and output of a biclustering algorithm where the left panel shows the heatmap of gene expression data and the right panel shows three biclusters. It is highly appreciable to analyze data by simultaneous row-column clustering to uncover many biological pathways via discovering local expression patterns from transcriptomics data. Besides the biological application, it is also used in many other fields like text mining, market research, information retrieval, and many more.

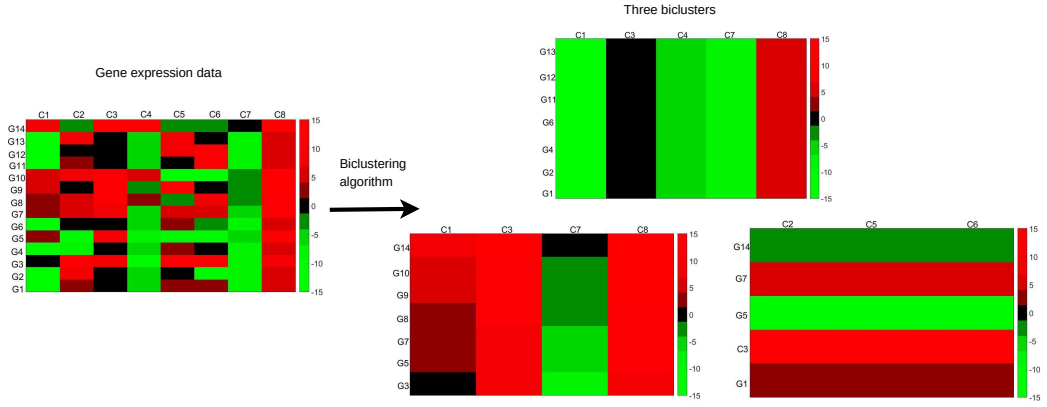


Figure 1.10: The input and output of a biclustering algorithm.

1.3.3 Tricluster analysis

Biclustering algorithm finds the submatrices at the same time, but nowadays researchers are interested to identify genes that are co-expressed under a subset of samples or experimental conditions across time points from GST data. Here, the biclustering algorithm fails to deal with such 3D data. Therefore, we need to move one step ahead from biclustering to 3D subspace clustering which can effectively handle the 3D gene expression data to fully understand the hidden biological knowledge. The subspace clustering for 3D data is essentially known as triclustering and the term tricluster can be delineated as a subset of genes exhibiting similar expression profiles under certain samples across a series of time-points [37, 364]. The schematic diagram of GST data and corresponding triclusters can be found in Figure 1.11.

1.3.4 Incorporating biological knowledge

In the field of biology, one of the most important knowledge-based biological database is Gene Ontology (GO) [19]. The focus of current research has steered towards the functional categorization of gene expression data using GO. Ontology is a knowledge representation in a specific domain. According to the Gene Ontology Consortium, 2000, the GO database shows the hierarchical structure of gene annotations reflecting the association among genes and biological terms. GO provides the controlled vocabulary of about 30,000 terms for the three distinct domains Biological Process (BP), Cellular Component (CC), and Molecular Function (MF) to represent the gene properties, gene functionalities, or gene itself. BP refers to the contribution of gene products towards biological objectives, MF defines biochemical activities of a gene product, and CC tells the location in

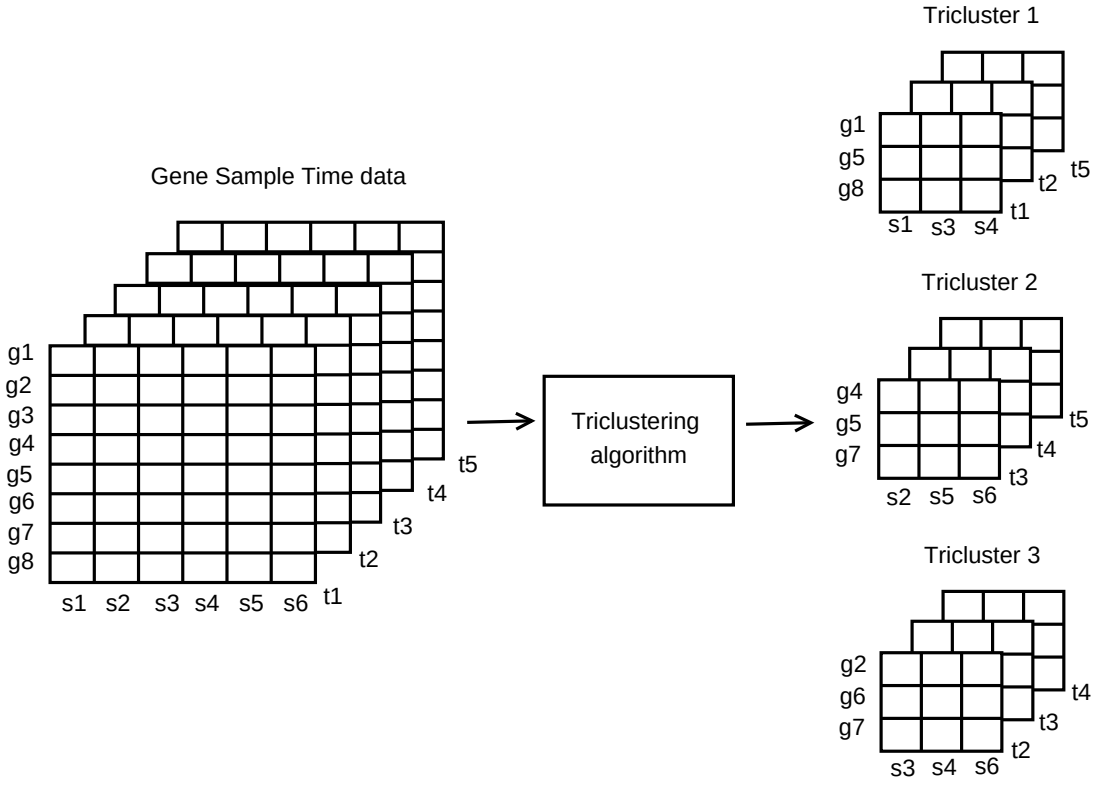


Figure 1.11: The input and output of a triclustering algorithm.

the cell of gene product activation. Ontology is actually some set of terms with different hierarchical relationships or parent-child relationships (*is_ a*, *part_ of*) which is functioning in the previously mentioned domain. If a child term is a specialization of a parent term, then it is denoted by the *is_ a* relation and if a child term is a component of a parent term, then it is described by the *part_ of* relation. Figure 1.12 depicts the relations in the GO [41]. GO terms are related to each other with negative regulation and positive regulations. GO is represented as a rooted *Directed Acyclic Graph* (DAG) where each node is represented by a GO term and the edge represents the relationship between the nodes. This graph forms as a hierarchy in such a way that one GO term is related to other GO terms, but the child node may have more than one parent. The root nodes are Biological Process, Cellular Component, and Molecular Function. Annotation is the association among the genes and the controlled vocabulary (gene ID) [272]. For example gene *COL1A1* annotated with 7 MF terms (GO:0048407, GO:0046872,...), 11 CC terms (GO:0005581, GO:0005584,...), and 54 BP terms (GO:0001503, GO:0001649,...). In GO, two genes may be annotated with the same term or they may be annotated via a shared term depending on the GO hierarchy [225]. The knowledge represented in the GO hierarchy may be used to guide the unsupervised clustering process into a semi-supervised clustering which

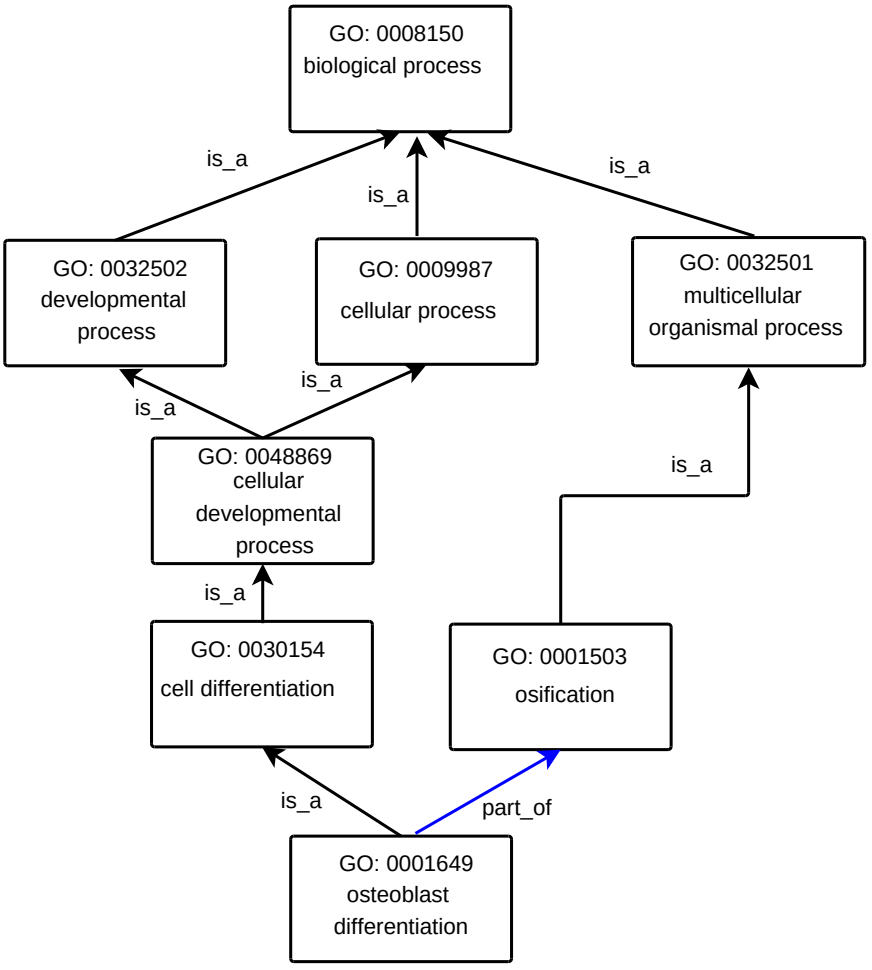


Figure 1.12: Relations in the GO.

will give functionally enriched clusters.

Another biological knowledge is pathway. A pathway is comprised of a set of interactions among molecules in a cell leading to a significant biological process [93]. In systems biology, to understand the functionality of a group of genes, a pathway plays a crucial role. Pathways can also unveil many complex cellular phenomena or genetic mechanisms related to complex diseases such as cancer at molecular levels, which may not be possible only by using expressions of genes. Complex biological pathways have been proven extremely beneficial in analyzing biological data when coupled with algorithmic framework [249]. Numerous pathway databases are available such as Kyoto Encyclopedia of Genes and Genomes (KEGG), WikiPathways, Reactome, and NCIPathways. These databases differ in the presence of the number of pathways, the average number of proteins in the pathway, categories of pathways, and the biochemical interactions. Notably, among these databases, KEGG is over-represented in the context of publication [249]. Hence, KEGG is considered a good source [172]. KEGG is

a knowledge base that links genomic information with higher functional information. Biological function is not subjected to a single gene rather it is comprised of multiple molecules and processes. Functional assignment defines the linking of a set of genes in the genome with the help of a network of molecules, termed as pathways, basically representing higher order biological functions.

1.4 Cancer transcriptomic profiling

One of the leading causes of death is cancer which has become a serious life-threatening disease for human beings. According to the statistical report, the total number of new cases of cancer has risen to 19.3 million globally with 10.0 million deaths in 2020 [50]. In India, new cases of cancer have been estimated to be 13.9 lakh and it is expected to reach up to 15.7 lakh by 2023[9]. Worldwide, the number of breast cancer in women is escalating. Amongst men cancer, lung cancer is the most frequently occurring cancer, prostate cancer being the second, and colorectum cancer is the third most familiar type of cancer. Patients suffering from advanced stages of cancer result in poor prognosis and also high recurrence rate [208]. Despite having therapeutic advancement of pharmacogenomics and medicine, early cancer detection for increasing the patient's survival rate is still a challenging task.

Genetic mutation of a gene is the reason that cancer develops and which in turn, loses control over the cell proliferation [288]. Another cause of cancer or developing malignant cells is dysregulation of biological pathways [109, 215]. Recently, cancer types can generally be categorized by molecular characteristics which include gene expression data [231]. Abnormal expression of transcripts involves the main characteristics of cancer disease by altering its expression levels, polymorphism, and isoforms [312]. Therefore, gene expression data is used to study the transcriptome of cancer for detecting novel transcripts and alternative splicing with higher accuracy [208]. Gene expression data also helps to identify diseased genes by varying the expression value under standard and diseased conditions [147]. From emerging evidence, it is strongly believed that abnormally expressed miRNAs play a critical role in various diseases including cancer [123]. Over the recent past, it has been observed that alteration of the expression of miRNA genes is responsible for malignant tumor development, cell proliferation, and apoptosis [73]. It has also been verified that miRNA differentially expresses in specific tissue and several types of cancer [109].

[9] `https://www.ndtv.com`

1.5 Potential biomarkers identification

The term biomarkers or biological markers are considered to be a relevant index of normal and abnormal biological processes by analyzing different biomolecules. According to the definition given by World Health Organization, "A biomarker is any substance, structure or process that can be measured in the body or its products and influence or predict the incidence of outcome or disease". Over the past decade, the definition of biomarker has been evolving. Generally, a biomarker can distinguish between healthy and diseased persons. A large variety of biomarkers include proteins, genes, and miRNAs. Clinically, a cancer biomarker may measure the risk factors growing in the specific cell, the possible response over several treatments or cancer progression [120].

Predictive, prognostic, and diagnostic biomarkers are the main types of biomarkers based on their usage [187]. A predictive biomarker is useful in identifying individuals that predict response to a particular medical therapeutic [120]. The mutation of *BRCA1/BRCA2* (BReast CAncer genes 1 and 2) may be considered as predictive biomarkers while women suffering from ovarian cancer [190]. On the other hand, a prognostic biomarker is not directly linked with a particular treatment rather it indicates the physician about the likelihood of future clinical outcomes (deaths or new medical conditions), disease recurrence, or progression to therapy. For example, *BRCA1/BRCA2* can be treated as prognostic biomarkers to assess the second occurrence of breast cancer while evaluating women diagnosed with cancer [28]. Another type of biomarker, i.e., a diagnostic biomarker is used for a critical determination that a patient suffers from a specific medical condition for which he/she should get enrolled in a particular clinical trial for a specific disease. For example, stool DNA testing can identify cancers for patients suffering from colorectal cancer [153].

The molecular characterization of gene expression data is considered to have a great role in early diagnosis of cancer by discovering prognostic biomarkers [291, 369]. Identification of cancer risk groups will facilitate better treatment and increase the survival rate of a patient's lifetime. It helps in determining the risk of developing cancer. For instance, a woman with having a strong family background of having ovarian cancer can go for genetic testing for the purpose of knowing if she is a carrier for mutation of *BRCA1* which may increase the risk factor of cancer [137]. Reliable biomarkers are extremely beneficial in understanding the complexity of various diseases [288], reduction of cost, simplifying the experimental setup, and providing a reference to the actual wet laboratory experimental results [231]. A substantial amount of literature study demonstrates

that miRNAs may act as oncogenes, potential biomarkers, or tumor suppressors since they might be involved negatively and positively in the regulation of human diseases by controlling the intracellular signaling pathways, thereby directly affecting the proliferation and apoptosis [109]. Three specific characteristics such as high sensitivity, specificity, and predictivity make miRNA play a key role as biomarkers in cancer [187]. The main challenge is to identify the miRNA biomarkers involved in dysregulation of the pathways in cancer.

The identification of cancer-related biomarkers will boost cancer management successfully and help diagnostics in a better way. Though several biomarkers have been identified which are used for diagnosis, still there is a need for improving the process of identifying new biomarkers. Many strategies have made it possible to discover biomarkers and selecting a proper method is a very challenging task [245]. In addition to conventional methods [59, 168, 206, 231], some frequently used strategies dealing with biomarkers or causal gene identification for cancer include differential expression analysis [309, 322], network analysis [183], co-expression network analysis [211], top gene ranking [206], statistical analysis [175], and classification [97]. The thesis outlines the biomarker identification method utilizing the clustering and subspace clustering methods.

1.6 Motivation

Clustering algorithms are unsupervised by nature, i.e., no priori knowledge is required. For transcriptomics data, no prior knowledge is available previously for discovering interesting patterns. Even though clustering analysis is an exploratory technique for determining the relationship in gene expression data [275], still it does not give the biologically meaningful correlation between genetic co-regulation and affiliation to a common biological process [2]. Moreover, using only expression values do not give the biological relationships in clusters. Therefore, sufficient attention is given to incorporate the biological knowledge during the search process to ensure that co-expressed genes are highly relevant biologically [255]. While integrating biological knowledge into gene expression matrix during clustering, it no longer seems to be an unsupervised approach and turns into a semi-supervised clustering. Basically, biological knowledge is used as a posterior criterion to ensure the relevancy of the discovered clusters. Bryan [51] has identified some limitations of gene expression data clustering and pointed out that the lack of natural gene clusters can be overcome by using semi- or supervised learning. With this belief, we have been motivated to develop clustering

algorithms with biological knowledge in this thesis.

Interestingly, during the last few years, knowledge-driven approaches have been gaining popularity because statistically significant and homogeneity solutions may not be biologically relevant [135]. GO plays a pivotal role in capturing the relationship among genes and hence give an added advantage if it is incorporated into the full-space clustering process as GO contains the biological classifications of all known genes. This motivates us to investigate external information from GO in this particular domain.

The active involvement of semi-supervised learning approaches have led to their advancement and gaining popularity in the field of clustering and biclustering [135, 214, 225, 255, 325]. The key features of a biclustering algorithm is to identify various types of biclusters, viz. (i) constant bicluster, (ii) row-constant bicluster, (iii) column-constant bicluster, (iv) additive bicluster, (v) multiplicative, (vi) additive-multiplicative, (vii) up-regulated, and (viii) trend-preserving (Explained in Chapters 2 and 4). Literature suggests that biclustering algorithms work well for a particular bicluster type [266]. Therefore, developing an efficient biclustering algorithm that can discover all types of bicluster models is of utmost importance. Since biclustering is an NP-hard problem, it becomes very slow while comparing tens of thousands of patterns. To explore biclusters from such large datasets we take the help of parallel computing to accelerate our proposed algorithm. We have proposed two biclustering algorithms, one unsupervised parallel biclustering algorithm and another semi-supervised biclustering algorithm that can identify all types of biclusters. To obtain better quality biclusters knowledge base i.e., GO annotations have been incorporated. In the previous work, knowledge-driven clustering algorithm we use GO as biological knowledge. Recently, it has been demonstrated that pathway-based approaches are more reliable and robust for analysis of gene expression data [365]. This is because gene expression data and protein-protein networks do not yield coherent gene subsets as pathways do [178]. According to Kim et al. [178], integrating pathways and gene information improves the performance of semi-supervised learning with the goal of differentiating disease phenotypes. We have proposed a biclustering technique incorporating KEGG pathways as the biological knowledge base as KEGG is open source, well established and highly cited. It is noteworthy that miRNA expression analysis follows a similar approach like mRNA expression analysis [356].

Li and Tuck [194] have proposed a triclustering algorithm that integrates gene expression and gene regulatory information for clustering. Integration of any

biological information from the open access databases is a challenging task and is currently one of the most prominent research directions [103]. The rationale behind this particular work is the intuitive idea of combining KEGG pathway knowledge in the triclustering process as we do for biclustering. We propose a novel triclustering algorithm for 3D data. To the best of our knowledge, KEGG pathway information has not been investigated till the writing of this thesis in triclustering algorithms. We have also explored all types of tricluster models such as additive, multiplicative, and additive-multiplicative, whereas the state-of-the-art methods fail to identify all types of triclusters by a single algorithm. Towards the end of each work, we have tried to identify biomarkers employing the strength of clustering and subspace clustering algorithms.

1.7 Objectives

The problem statement of our research is as follows.
Given a transcriptomics cancer data, the key goal of our research is to develop semi-supervised full-space and subspace clustering techniques incorporating different biological knowledge to find biologically relevant clusters and identify potential cancer biomarkers.
To achieve this purpose, we have subdivided our work into several objectives as below.

1 To develop unsupervised and semi-supervised full-space clustering algorithms incorporating Gene Ontology for cancer gene expression in order to identify potential biomarkers.

2 To develop a biclustering algorithm for cancer transcriptomics data that is able to detect all types of biclusters and identify potential biomarkers.

3 To develop a semi-supervised biclustering algorithm for cancer transcriptomics data incorporating KEGG pathway information, capable of discovering all types of patterns and identify potential biomarkers.

4 To develop a semi-supervised triclustering algorithm guided by KEGG pathway information which can find additive, multiplicative, and additive-multiplicative patterns from cancer GST data, allowing to find potential biomarkers.

The assessment of the aforementioned clustering algorithms has been carried out through empirical study with the help of both synthetic and real datasets. The entire workflow of our thesis is given in Figure 1.13.

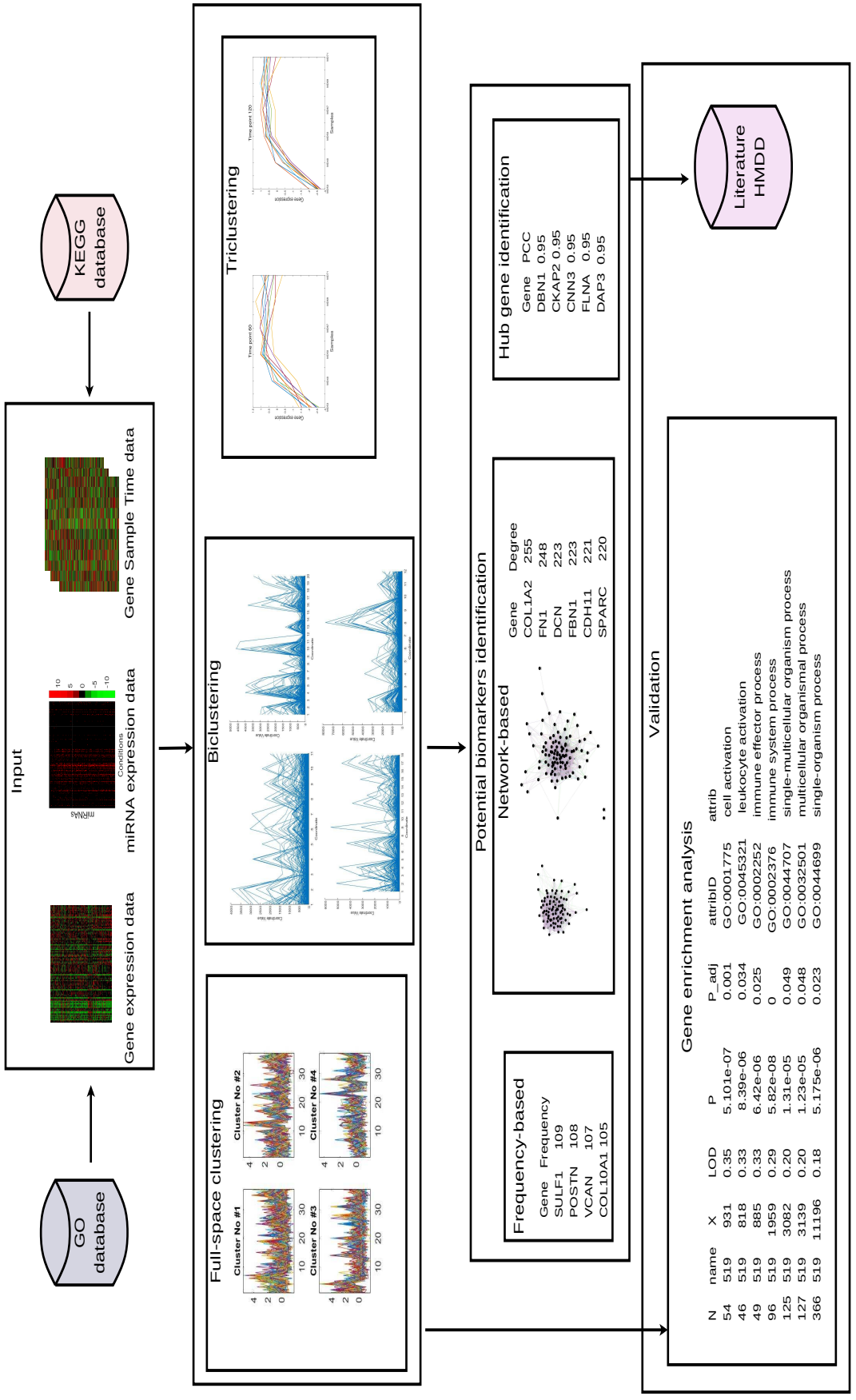


Figure 1.13: Workflow of our thesis.

1.8 Contributions

This thesis is divided into seven chapters. Based on the objectives in mind, we have organized the contributions of the thesis as follows.

Chapter 2 - Background: The key concepts of cluster analysis of transcriptomics data are presented. The fundamentals of classical clustering, biclustering, and triclustering from the literature are summarized. The pitfalls of traditional clustering algorithms over subspace clustering are reported. Additionally, relative cluster validation criteria are briefly reviewed.

Chapter 3 - Full-space Cluster Analysis of Cancer Gene Expression Data: Three contributions regarding the clustering algorithms are developed. In this chapter, firstly, we introduce an unsupervised full-space clustering algorithm, **G**raph **A**ttraction **Clust**ering (GAClust). Secondly, we propose a **S**emi-supervised **D**ensity-based **C**lustering (SDC) algorithm which takes GO information during clustering procedure. At last, being motivated by the incorporation of Gene Ontology as a biological knowledge, we modify GAClust into **S**emi-supervised **G**raph **A**ttraction **Clust**ering (SGAClust). Moreover, a network-based biomarker identification methodology is demonstrated that employs identified clusters. An empirical analysis of unsupervised clustering algorithms has been explored for synthetic datasets. To evaluate the efficacy of these algorithms, we use cancer microarray gene expression datasets and then identify potential biomarkers using a network-based method. Finally, the chapter discusses the goodness of semi-supervised over unsupervised methods experimentally.

Chapter 4 - Bicluster Analysis of Cancer Transcriptomics Data: This section, deals with a biclustering algorithm named **O**rder-**P**reserving **Bic**lustering (OPBic) algorithm. Not only that but we also present a frequency-based biomarker identification method utilizing discovered biclusters. We apply the proposed algorithm in synthetic datasets, real cancer microarray gene expression, and breast cancer miRNA expression datasets. In association with this, we identify potential biomarkers using frequency and network-based methods for both types of real datasets. A detailed analysis of the results is presented along with a discussion towards the end of this chapter.

Chapter 5 - Semi-supervised Bicluster Analysis of Cancer Transcriptomics Data: It has been observed from Chapter 3 that a knowledge-driven clustering algorithm improves the cluster quality over an unsupervised clustering algorithm. Therefore, we propose a semi-supervised biclustering algorithm i.e., **P**athway-based **O**rder-**P**reserving **Bic**lustering (POPBic) algorithm

which is guided by pathway information. Literature suggests that pathway is more reliable than GO which is demonstrated in this chapter. We evaluate our algorithm using synthetic and cancer expression data. Like chapter 4, we identify potential biomarkers. The chapter ends with a discussion on the importance of semi-supervised biclustering algorithms.

Chapter 6 - Semi-supervised Tricluster Analysis of Cancer Gene Expression Data: Inspired from the promising results of the semi-supervised biclustering algorithm in Chapter 5, we outline a pathway-based triclustering algorithm named **P**athway-based **O**rder-**P**reserving **Tric**lustering (POPTric) algorithm in order to analyze breast cancer GST data. The assessment of the proposed algorithm is done using both synthetic and breast cancer microarray gene expression data. A separate method is used to find biomarkers from identified triclusters. The chapter closes with a discussion providing the advantages and disadvantages of POPTric algorithm with state-of-the-art methods.

Chapter 7 - Conclusion and Future Work: In this chapter, first, the main contribution of the entire thesis is highlighted. Next, strong points of all the proposed methods are discussed along with the concluding remarks. Finally, we discuss the shortcomings of all proposed algorithms and possible ways to provide future directions of our work for further research. In the next chapter, we present the background of our work.

2

Background

In the last two decades, analysis of genome-wide transcriptome profiles has received a lot of attention. This chapter aims to describe cluster analysis of transcriptomics data. Here, we introduce mathematical formulations, background concepts, and definitions which is used throughout the thesis. Firstly, we start by briefly describing the input data to be used and highlighting the importance of cluster analysis of biological data in section 2.1. Next, we survey the fundamentals of full-space clustering, biclustering, and triclustering which are major three methodologies used in this thesis along with cluster validation, respectively in section 2.2, 2.3, and 2.4.

2.1 Introduction

The knowledge about gene expression data may lead to understanding the work of different types of cells and organisms. It has the potential to unveil the mechanism of various diseases such as cancer at the molecular level. Hence, it can identify the disease related genes that can be targeted in personalized treatment. In the previous Chapter 1, we have provided the necessary background of gene expression data. Now, we focus on the mathematical notation of gene expression data. Let, $ED_{m \times n}$ be an expression data matrix organized

in terms of m rows, $G = \{g_1, g_2, \ldots, g_m\}$ and n experimental conditions or samples, $C = \{c_1, c_2, \ldots, c_n\}$. Rows represent genes or miRNAs. Each entry $ge_{ij} \in ED_{m \times n}$ of the matrix corresponds to the value of a row g_i under a specific column c_j, where $i = \{1, 2, \ldots, m\}$ and $j = \{1, 2, \ldots, n\}$. The number of genes is significantly larger than the number of columns.

A popular technique, clustering is used in analyzing the activities of genes from the profiles of expression data [164, 174, 265, 275]. The common aim of the clustering method is to find co-expressed, co-regulated, and coherent pattern genes. If two genes share a similar expression profile, then they are called co-expressed genes [163]. Co-expressed genes have a similar functional category. Analysis of a group of genes is vital rather than taking a single gene under consideration. The term co-regulated indicates those genes that are regulated (up or down) by some common transcription factors (one kind of protein helps in transcribing process). Co-expressed pattern may also indicate co-regulation if it has a strong expression pattern [162]. According to Daxin Jiang et al. [163], a coherent pattern represents the common trend (centroid or mean) of the expression level of genes in a particular co-expressed cluster. It has been observed that co-expressed genes reveal coherent patterns indicating similar expression profiles and may share similar biological functions [224, 275]. Moreover, co-expressed genes are regulated by each other or by parent genes [243].

2.2 Full-space clustering

The aim of traditional clustering is to find groups of genes with similar behavior across the entire set of conditions. In general, full-space clustering clusters either set of G (or C) as objects and set of C (or G) as features. In this context, there is no well-accepted global definition of the term cluster. Traditional or single dimension clustering which captures a global pattern of data can be classified into two types, gene-based and sample-based [164]. In gene-based clustering genes (rows) are treated as objects whereas conditions (columns) are considered to be features. Sample-based clustering is the opposite of gene-based clustering where genes are features and conditions are objects. Here, we have focused to identify gene-based clustering. Next, we discuss some of the details of full-space clustering algorithms which are employed as the basis of developing algorithms in this thesis. Before presenting the main algorithmic concepts of full-space clustering algorithms, we review proximity measures in the next section which are employed in the clustering methods.

2.2.1 Proximity measures

The term proximity measure reveals (dis)similarity between a pair of objects x and y, where $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$. The dissimilarity metric between x and y i.e., $Dist(x, y)$ should satisfy the following conditions.

- $Dist(x, y) \geq 0$
- $Dist(x, x) = 0$
- $Dis(x, y) \leq Dist(x, z) + Dist(z, y)$, where z is another object
- $Dist(x, y) = Dist(y, x)$

Choosing an appropriate proximity measure is considered to be a more important task than choosing the clustering algorithm [158]. There is no complete recommendation about the selection of proximity measures associated with the clustering algorithm. Pablo A. Jaskowiak et al. [158] have surveyed different proximity measures useful for clustering of gene expression analysis. In literature different proximity measures are available such as Pearson Correlation Coefficient (PCC), Goodman-Kruskal (GK), Kendall (KE), Spearman (SP), Rank-Magnitude (RM), and Weighted Goodman-Kruskal (WGK) as correlation coefficients; Euclidean distance, Cosine distance (CSD), Minkowski distance as classical measures; and Jackknife (JCK), Short Time-Series dissimilarity (STD), Local Shape-based Similarity (LSS), YR1, and YS1 dissimilarity as time-series specific measures have been used for clustering purpose [158]. Another study of Pablo A Jaskowiak et al. [157] have shown that appropriate selection of proximity measure can make a significant difference between a meaningful and poor clustering result. First, we elaborate in the sequel six correlation coefficients. After that, we discuss three classical measures and finally, we review five proximity measures, especially proposed for the clustering of gene time-series data.

(I) **Correlation coefficients:** Correlation coefficients are the most commonly used measures in the clustering of gene expression data. Two genes are said to be similar if they exhibit similar shapes rather than considering the absolute difference in their values. The correlation coefficient values range from -1 to 1. It can be easily converted to the distance measure by subtracting the correlation value from 1.

Pearson Correlation Coefficient: The most common metric Pearson Correlation Coefficient (PCC) is sensitive in the presence of outliers still it is

used in many research due to lesser time complexity [158] compared to other measures. The PCC [164] between two sequences of objects is defined as follows,

$$PCC(x,y) = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}} \tag{2.2.1}$$

where μ_x and μ_y are the means of the sequence x and y, respectively.

Spearman: Spearman correlation (SP) can be obtained by Equation 2.2.1, where x and y values are replaced by its rank in their sequences [305]. Due to the use of ranks, SP measure is less sensitive towards outliers than PCC.

Goodman-Kruskal: Goodman-Kruskal (GK) [119] is a rank-based measure that takes into consideration the ranks of x and y. It is defined by Equation 2.2.2, where C_+ and C_- denote the number of concordant and discordant pairs of elements in x and y, respectively. If $(x_i < x_j)$ and $(y_i < y_j)$ or $(x_i > x_j)$ and $(y_i > y_j)$, then it is considered to be concordant pair, i.e., relative order is present in both sequences. On the other hand, inverse relative order is applied for both sequences to get discordant pairs, i.e., $(x_i > x_j)$ and $(y_i < y_j)$ or $(x_i < x_j)$ and $(y_i > y_j)$. Rest of the pairs which are neither discordant nor concordant are deemed neutrals.

$$GK(x,y) = \frac{C_+ - C_-}{C_+ + C_-} \tag{2.2.2}$$

Kendall Correlation Coefficient: Kendall Correlation Coefficient (KE) is also a rank-based approach and uses the same principle as introduced in GK [173]. The computation of KE is shown in Equation 2.2.3, where the denominator is the total number of possible pairs of elements in the sequences.

$$KE(x,y) = \frac{C_+ - C_-}{\frac{n(n-1)}{2}} \tag{2.2.3}$$

Weighted Goodman-Kruskal: The weighted Goodman-Kruskal (WGK) [53] takes into account both the rank and magnitude of the given x and y and is presented in Equation 2.2.4. The equation 2.2.5 defines $\hat{\mathcal{W}}_{ij}$, where $\hat{\mathcal{W}}^x_{ij}$ and $\hat{\mathcal{W}}^y_{ij}$ are determined by 2.2.6. $\hat{\mathcal{W}}^x_{ij}$ and $\hat{\mathcal{W}}^y_{ij}$, basically represent the percentual (signed) difference in i^{th} and j^{th} elements of the sequences under consideration. $\mathcal{W}_{ij}$ is represented by Equation 2.2.7, where $\mathcal{W}^x_{ij} = sign(x_i - x_j)$ and

$\mathcal{W}^y_{ij} = sign(y_i - y_j)$.

$$WGK(x,y) = \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\hat{\mathcal{W}}_{ij}}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}|\mathcal{W}_{ij}|} \tag{2.2.4}$$

$$\hat{\mathcal{W}}_{ij} = \begin{cases} \min\{\frac{\hat{\mathcal{W}}^x_{ij}}{\hat{\mathcal{W}}^y_{ij}}, \frac{\hat{\mathcal{W}}^y_{ij}}{\hat{\mathcal{W}}^x_{ij}}\} & \text{if } \hat{\mathcal{W}}^x_{ij}, \hat{\mathcal{W}}^y_{ij} > 0 \\ \max\{\frac{\hat{\mathcal{W}}^x_{ij}}{\hat{\mathcal{W}}^y_{ij}} \frac{\hat{\mathcal{W}}^y_{ij}}{\hat{\mathcal{W}}^x_{ij}}\} & \text{if } \hat{\mathcal{W}}^x_{ij}, \hat{\mathcal{W}}^y_{ij} < 0 \\ 1 & \text{if } \hat{\mathcal{W}}^x_{ij} = \hat{\mathcal{W}}^y_{ij} = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.2.5}$$

$$\hat{\mathcal{W}}^a_{ij} = \begin{cases} \frac{a_i - a_j}{a_{max} - a_{min}} & \text{if } a_{max} \neq a_{min} \\ 0 & \text{otherwise} \end{cases} \tag{2.2.6}$$

$$\mathcal{W}_{ij} = \begin{cases} \frac{\mathcal{W}^x_{ij}}{\mathcal{W}^y_{ij}} & \text{if } \mathcal{W}^y_{ij} \neq 0 \\ 1 & \mathcal{W}^x_{ij} = 0 \text{ and } \mathcal{W}^y_{ij} = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.2.7}$$

Rank-Magnitude: Rank-Magnitude (RM) correlation coefficient considers two sequences where one sequence is made of ranks and the other one is composed by real numbers. This is an asymmetric measure as shown in Equation 2.2.8, where $r_{max} = \sum_{i=1}^{n} i\overline{y_i}$, $r_{min} = \sum_{i=1}^{n}(n+1-i)\overline{y_i}$, and $R(x_i)$ is the rank of i^{th} position for sequence x. $\overline{y_i}$ is referred to the i^{th} element of the sequence by rearranging x in ascending order. Another symmetric version of RM is mentioned in Equation 2.2.9 which compares two real valued sequences considering both rank and magnitude [156].

$$\hat{\mathcal{R}}(x,y) = \frac{2\sum_{i=1}^{n} R(x_i)y_i - r_{max} - r_{min}}{r_{max} - r_{min}} \tag{2.2.8}$$

$$RM(x,y) = (\hat{\mathcal{R}}(x,y) + \hat{\mathcal{R}}(y,x))/2 \tag{2.2.9}$$

(II) **Classical measures:** Now, we review some of the commonly used traditional distance measures.

Cosine distance: The Cosine similarity (CS) [85] or angular separation is defined by Equation 2.2.10 and it is related to PCC with some small differences. CS is interpreted as the normalized inner product between x and y. Basically, it is the cosine of the angle between two data points with respect to the origin.

The Cosine distance (CSD) can be calculated as $CSD(x, y) = 1 - CS(x, y)$.

$$CS(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n}(x_i)^2}\sqrt{\sum_{i=1}^{n}(y_i)^2}} \tag{2.2.10}$$

Minkowski: It has been found from the literature that Minkowski has been considered to be the most popular distance measure as defined in Equation 2.2.11 [154]. With different parametric values of p we can obtain different distance measures, such as Manhattan distance for $p = 1$, Euclidean distance for $p = 2$.

$$DM(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}} \tag{2.2.11}$$

(III) **Time-series specific distances:** We discuss some of the proximity measures which are generally used for time-series gene expression data. Let, $t = \{t_1, t_2, \ldots, t_n\}$ be the time points at which gene expression values are measured.

Jackknife: We know that PCC is susceptible to outliers. Therefore, to overcome the shortcoming of PCC, the Jackknife correlation (JCK) is introduced via reducing the effect of single outliers by eliminating values from both the sequences during the PCC calculation [139]. JCK is represented in Equation 2.2.12, where $PCC^i(x, y)$ is the Pearson Correlation Coefficient of x and y after removal of i^{th} value. If $i = 0$, it means that no value has been removed. Moreover, if outliers are not present in both the sequences then their correlation value is stable otherwise the value is decreased.

$$JCK(x, y) = \min_{0 \le i \le n} PCC^i(x, y) \tag{2.2.12}$$

Short Time-Series dissimilarity: Short Time-Series dissimilarity (STD) computes the distance of two gene time-series data considering the fact of having $n - 1$ slopes in time-series [246]. The formula of STD is defined in Equation 2.2.13.

$$STD(x, y) = \sqrt{\sum_{i=1}^{n-1}\left(\frac{y_{i+1} - y_i}{t_{i+1} - t_i} - \frac{x_{i+1} - x_i}{t_{i+1} - t_i}\right)^2} \tag{2.2.13}$$

Local Shaped-based Similarity: The underlying concept behind the Local Shaped-based Similarity (LSS) is based on the understanding of biological relationships between genes. It has been observed that similarity between two

genes may occur locally or in the subspace of the features from the x and y sequences [25]. LSS is delineated in Equation 2.2.14, which finds the most similar subsequences having k size between two time-series sequences. Here, S is the base similarity of two k sized subsequences between x and y. min_k is the minimum subsequence size which is set to $n-2$. Furthermore, LSS gives the maximum similarity value among the variable size of subsequences.

$$LSS(x,y) = \max_{min_k \leq k \leq n} (\max_{1 \leq 1, j \leq n-k+1} (S(x[i, i+k-1], y[j, j+k-1]))) \quad (2.2.14)$$

YR1 and YS1 dissimilarity: Correlation is unable to capture similarity in gene time-series data. Therefore, to overcome this problem YR1 and YS1 dissimilarity has been proposed taking into account various information along with correlation measure [157]. YR1 and YS1 are shown in Equations 2.2.15 and 2.2.16, respectively, where ζ_1, ζ_2, and ζ_3 are weights and should satisfy $\sum_{i=1}^{3} \zeta_i = 1$. YR1 considers PCC, where $R(x,y) = (PCC(x,y)+1)/2$ and YS1 takes into account SP, where $S(x,y) = (SP(x,y)+1)/2$. Equation 2.2.17 indicates the comparison of two time-series having n features with $n-1$ slopes, where 2.2.18 provides the definition of function L. In Equation 2.2.17, function $\mathbb{F}$ returns 1 for agreement or 0 otherwise. The slope of a gene x is determined by Equation 2.2.19 for a certain time interval. Moreover, an another agreement between x and y concerns the minimum and maximum expression values are from the same timestamps or not. Such concept is presented in Equation 2.2.20.

$$YR1(x,y) = \zeta_1 R(x,y) + \zeta_2 A(x,y) + \zeta_3 M(x,y) \quad (2.2.15)$$

$$YS1(x,y) = \zeta_1 S(x,y) + \zeta_2 A(x,y) + \zeta_3 M(x,y) \quad (2.2.16)$$

$$A(x,y) = \sum_{i=1}^{n-1} \frac{\mathbb{F}(L(x,i) = L(y,i))}{n-1} \quad (2.2.17)$$

$$L(z,i) = \begin{cases} 1 & \text{if } slope(z,i) > 0 \\ -1 & \text{if } slope(z,i) < 0) \\ 0 & \text{if } slope(z,i) = 0 \end{cases} \quad (2.2.18)$$

$$slope(z,i) = \frac{z_{i+1} - z_i}{t_{i+1} - t_i} \quad (2.2.19)$$

$$M(x,y) = \begin{cases} 1 & \text{if } t_x^{min} = t_y^{min} \text{ and } t_x^{max} = t_y^{max} \\ 0.5 & \text{if } t_x^{min} = t_y^{min} \text{ or } t_x^{max} = t_y^{max} \\ 0 & \text{if } t_x^{min} \neq t_y^{min} \text{ and } t_x^{max} \neq t_y^{max} \end{cases} \tag{2.2.20}$$

2.2.2 Full-space clustering algorithms

This section reviews some of the clustering techniques applied in the field of bioinformatics and their pros and cons. A plethora of clustering algorithms have been developed in the context of gene expression data for identifying biologically relevant groups. It can be broadly classified into four major categories: Partitional, Hierarchical, Graph-theoretic, and Density-based, which are discussed below.

(I) **Partitional-based clustering approaches:** The most fundamental clustering algorithm is a partitional-based clustering algorithm. This type of algorithm partitions the dataset ED containing m genes into K (used defined) clusters ($K \leq m$) $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$, where each data is residing in only one cluster $\mathcal{C}_i \cap \mathcal{C}_j = \phi$. We can comprehend the notion of partitional clustering algorithms as hard or crisp because each data object belongs to only one cluster. The optimal partition is being determined with respect to some objective partitioning criterion.

The simplest partitional-based algorithm is K-means [226] which is widely used in gene expression data. Each cluster representative is the mean (centroid) of all data points residing in the same cluster. The algorithm initiates by choosing K random centroid from the m data and tries to assign the data to its nearest centroid based on some dissimilarity measure. It then recalculated the cluster centroid of each cluster and repeats the procedure until the centroids do not change or the sum squared error (SSE) as shown in Equation 2.2.21 is minimized enough.

$$SSE = \sum_{i=1}^{K} \sum_{g \in \mathcal{C}_i} Dist(\mathcal{O}_i, g)^2 \tag{2.2.21}$$

Here, g is the data point in the $\mathcal{C}_i$ clusters and $\mathcal{O}_i$ is the mean or centroid of the cluster. Though the K-means algorithm is very fast and easy to implement, it suffers from several drawbacks. The algorithm suffers from a predefined input parameter determination problem and is unable to find the arbitrary shaped clusters. Here, initial cluster centroids are determined randomly and domain knowledge is required to predict the appropriate value of K. Another drawback

of the K-means algorithm is non-robustness, which is very necessary for analyzing noisy gene expression datasets. There are several variations of the K-means algorithm using soft computing techniques [185, 219, 295, 339]. The limitations of the K-means algorithm can be overcome by using incremental clustering [349]. At present, datasets are dynamic where new data is added to the existing dataset. Now, such datasets can not be taken care of by non-incremental clustering algorithms because these algorithms re-cluster the entire data and reduce the efficiency of the algorithms. In such a scenario, incremental clustering algorithms will play a critical role to group new data and updating the new cluster with previous results. Apart from this, the study mentioned in [300] has proposed a modified K-means algorithm to remove the outliers and enhance the overall performance of the clustering algorithm.

Self-Organizing Map (SOM) [313] based on an unsupervised artificial neural network, is more robust (cluster a huge amount of noisy data), reasonably fast, and easy to implement. SOM is first proposed by Kohonen [179]. In the beginning, SOM has a set of nodes with simple topology and a dissimilarity metric $Dist(N_1, N_2)$ on the node. During the training procedure iteratively nodes are mapped into k-dimensional (user given) data space. Let, $f_i(N)$ is the position of node N at iteration i. The initial mapping f_0 is random. On the next iteration, a data point says Q is selected and moving the node N_Q to its nearest Q by Equation 2.2.22.

$$f_{i+1}(N) = f_i(N) + \tau(Dist(N, N_Q), i)(Q - f_i(N)) \tag{2.2.22}$$

τ is the learning rate which is defined by Equation 2.2.23. τ decreases with the distance between two nodes N and N_Q with iteration number i.

$$\tau(x, i) = \begin{cases} \frac{0.02T}{(T+100i)} & \text{for } x = r(i) \\ 0 & \text{otherwise} \end{cases} \tag{2.2.23}$$

T is maximum number of iterations, radius $r(i)$ decreases linearly with i and becomes 0. SOM is a *topology-preserving* neural network and it depends on the initial grid structure (hexagonal, rectangular, grid, ring, lines). Though it has several drawbacks, SOM is successfully used in several gene expression studies [49, 60, 131, 140, 258, 310, 331].

Recently, Abu-Jamous and Kelly have proposed a partition-based method, named Clust [1]. The aim of this algorithm is not to consider the whole set of input data to be partitioned into clusters, instead, it identifies subsets that

are assigned to clusters.

(II) **Hierarchical clustering approaches:** Hierarchical clustering (HC) algorithm produces a group of nested clusters forming a tree like structure called dendrogram rather than forming a set of disjoint clusters like a partitional algorithm. Unlike the partition-based clustering approach, we do not have to assume a fixed number of clusters, rather we can get any number of clusters by cutting the dendrogram at a proper level. The variation of HC is of two types agglomerative and divisive. Agglomerative is a bottom-up approach where each data object is considered to be a single cluster. Two data objects are merged based on single linkage, average linkage, centroid linkage, or complete linkage. The process continues until all data points are merged into a single cluster. Whereas the divisive approach is just the opposite of the former one, it is a top-down approach. It starts with all data objects as a single cluster, data points are split to meet some heuristic criteria, until singleton clusters remain. HC finds a similar pattern and displays the result graphically which is easy to interpret for biologists. The problem associated with the hierarchical algorithm is its high computational complexity splitting or merging of each step takes $\frac{m^2-m}{2}$ times. The total time complexity of the agglomerative clustering algorithm is $O(m^2 \log(m))$. HC is a greedy approach, where once a decision has been taken, it can never be changed. Another drawback of HC is the lack of robustness.

Unweighted Pair Grouping Method (UPGMA) [95] is an agglomerative hierarchical approach based on the average linkage method. At the beginning of the algorithm, the similarity matrix is being created using PCC. The highest value of the matrix is being considered as the most similar pair of genes. These two genes are grouped and replaced by a single node. The gene expression matrix is computed by averaging recently joined genes. The similarity matrix is then updated and the same process is repeated until only one element remains. The HC with heatmap visualization of Michael B. Eisen [95] is widely used in analyzing gene expression data.

$$P_j(g_q) = exp\frac{-\beta|g_q - \mathcal{O}_j|^2}{\sum_j exp(-\beta|g_q - \mathcal{O}_j|^2)} \tag{2.2.24}$$

Alon et al. [7] have proposed a divisive HC based on Deterministic-Annealing Algorithm (DAA) [285]. Initially, two cluster centroids are chosen randomly, say $\mathcal{O}_i$ and $\mathcal{O}_j$. The gene expression pattern of gene q is represented by g_q. The probability of gene g_q to be in the cluster j is assigned according

to the Gaussian model represented in Equation 2.2.24. The centroid is updated iteratively by Equation 2.2.25.

$$\mu_j = \frac{\sum_q g_q P_j(g_q)}{\sum P_j(g_q)} \tag{2.2.25}$$

To solve P_j and $\mathcal{O}_j$, an iterative process is applied. There is one cluster i.e., $\mathcal{O}_1 = \mathcal{O}_2$ if $\beta = 0$. β value is increased slowly until it reaches a threshold in order to get two distinct, converged centroids. The procedure is repeated by splitting the whole dataset until we get a single gene in each cluster.

Contrary to the hierarchical clustering algorithm, the Self Organising Tree Algorithm (SOTA) is a divisive approach, where the clustering process is performed from top to bottom i.e., highest levels are resolved first before the lowest level [138]. SOTA is an unsupervised neural network that grows by adopting binary tree topology. As SOTA combines the good features from both the neural network of SOM and hierarchical clustering, that is why it can easily overcome the problem associated with the classical hierarchical approach. The algorithm initiates with a binary tree with a root node and two leaves, each of them represents a single cluster. Then the tree grows by converting the leaf with the largest resources into the node and attaching two new leaves to it. The resource of each cluster is calculated by the mean value of distance among a cluster and the expression profiles genes associated with it. The number of clusters can be determined by stopping the growth of the tree after some specific number of loops. The interesting feature of the SOTA algorithm is its linear time complexity. But it is slower than UPGMA. Other algorithms which follow a similar strategy are Dynamically Growing Self-Organising Tree (DGSOT) [221] and Growing Hierarchical Tree SOM (GHTSOM) [111].

(III) **Density-based clustering approaches:** The density-based clustering approach separates the highly dense region from low density regions. The well-known Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [101] can find clusters of arbitrary shapes in the presence of noise. The main concept of the DBSCAN algorithm is directed by two definitions, such as density-connectivity and density-reachability [101]. The algorithm depends on the user-defined parameters: neighborhood distance d_n and the minimum number of points in a cluster m_p. The algorithm starts with a random point, say x which is not visited and all density-reachable data are retrieved with respect to d_n and m_p. If x is a core point (minimum number of points m_p within its d_n distance), then a cluster is formed. If x is a border point then no data is found

to be density-reachable from x. Therefore, it visits the next point. The process is repeated until all points are visited once. The problem associated with this kind of algorithm is its dependency on input parameters and high computational complexity.

An extension of DBSCAN is Prototype-Based Modified DBSCAN (MDBSCAN) [94] which handles gene expression data. The first step of the MDBSCAN algorithm is to apply any squared error clustering algorithm such as K-means on the dataset in order to get K number of clusters. The centroid of each cluster is considered as a prototype and thereafter DBSCAN algorithm is applied to these prototypes. The main advantage of MDBSCAN is the elimination of extra computation of distance with the help of a prototype and can handle noise in the dataset.

Daxin Jiang et al. [162] have proposed an algorithm called Density-Based Hierarchical Clustering (DHC) which actually solves the problem of the density-based clustering approach and visualize the internal structure of clusters by graphical representation. The development of the DHC algorithm is based on the notion of two concepts "density" and "attraction". The key idea of the DHC algorithm is to find clusters with high-dimensional dense areas where data are attracted to each other. DHC organizes the data in two levels hierarchical structure. "Attraction tree" is constructed in the first level, which is used to represent the relationship between data points in the dense region. Every node of the tree presents data and the parent of each node denotes the attractor. The root of the attraction tree has the highest density in the dataset. The data structure becomes very complicated for large datasets and it is hard to interpret for large datasets of the attraction tree. In order to avoid this situation authors have developed a density tree in the second level to represent the cluster structure of the attraction tree. In the density tree, each node of the tree represents a dense area. Initially, the root node of the density tree represents the whole dataset as a single dense area. The dense area is further split into sub-dense areas based on some criteria. This process is repeated until each sub-dense area contains a single cluster. DHC is robust in the presence of noise and it is scalable to handle large datasets. Some other algorithms of this category are DGC [74], OverDBC [238], Bayesian-OverDBC [239] etc.

(IV) **Graph-theoretic clustering approaches:** The key goal of the graph-theoretic approach is to partition the data into subgraphs with the help of some geometric property. From the given dataset, we can make a proximity matrix and a weighted graph or proximity graph $G(V, E)$ from the matrix. The nodes V

of the graph are genes and edges E are the connection between two genes. The weighting scheme differs from algorithm to algorithm. This approach can easily handle the outliers and does not depend upon the parameter which determines the number of clusters.

Amir Ben-Dor et al. [33] have proposed a graph-theoretic algorithm for gene expression data Cluster Affinity Search Technique (CAST) which introduces an idea of corrupted clique graph model. CAST algorithm takes an $m \times m$ real-valued similarity sim matrix and a user-defined affinity threshold t_r as input. The affinity of data say d with respect to the current cluster $\mathcal{C}_{open}$ is defined as the sum of the similarity measure between d and all the data residing in cluster $\mathcal{C}_{open}$, $af(d) = \sum_{x \in \mathcal{C}_{open}} sim(d, x)$. Initially $\mathcal{C}_{open}$ is empty, the data point is added to the current cluster if d has high affinity i.e., $af(d) \geq t_r |\mathcal{C}_{open}|$ and removes the data point if it has low affinity. Adding and removing phases are simultaneously executed until $\mathcal{C}_{open}$ stabilizes and again restarts the process with a new cluster. Thus only one cluster is constructed at a time. The CAST algorithm provides several advantages; such as the number of clusters is not required apriori and identifies outliers from noisy data. But it needs proper tuning of the affinity threshold value, t_r.

CAST algorithm has two drawbacks- i) the user-defined affinity threshold value and ii) the cleaning step is required to define the position of the data points among all the clusters. Abdelghani bellaachia el al. [30] overcome the problem of threshold calculations CAST by proposing Enhanced-CAST (ECAST) which dynamically calculates the threshold value at the starting of every new cluster. The threshold value is calculated by Equation 2.2.26.

$$t_r = \left(\frac{\sum_{i,j \in U' \text{and } sim(i,j) \geq 0.5} sim(i,j) - 0.5}{|u : u \in U' \text{and } af(u) \geq 0.5|}\right) + 0.5 \tag{2.2.26}$$

U considers the total genes to be clustered and U' are those genes which are yet to be clustered, $U' = U \setminus (\mathcal{C}_1 \cup \mathcal{C}_2 \cup \ldots \cup \mathcal{C}_K)$. The similarity value sim above 0.5 has been tested over several datasets and gives good results.

CLuster Identification via Connectivity Kernels (CLICK) [293] does not depend upon the number of clusters and discovers "true" clusters via the graph-theoretic method and statistical techniques. The basic concept of the CLICK algorithm is to iteratively split the weighted graph into components with the help of a minimum cut. To assign the weight of the edge and the stopping criteria of splitting some probabilistic computation is used.

The review of full-space clustering algorithms is not restricted to the above-mentioned four categories. Literature is flooded with different full-space clustering algorithms. Further, it can be divided into soft clustering, multi-objective optimization methods, grid-based method, model-based methods which are beyond our scope of research. We refer interested readers to know more about clustering algorithms to [164, 174, 265, 275].

2.2.3 Cluster evaluation methods

Clustering algorithms will produce output with respect to the input data provided. It may be also possible that the algorithm discovers clusters even if data has no "true" clusters. In real data, it is not possible to know the number of clusters present in the data beforehand. Therefore, cluster validation is useful in order to avoid spurious or misleading clustering results. This is one of the most challenging parts of clustering. Clustering validation methods as used to evaluate the clusters in an objective and quantitative manner [154].

The issue associated with clustering is to select the best algorithm as well as the best setting of parameters. These issues can be addressed by cluster validation technique assuming the presence of clusters in the input data. In this context, evaluation methods help researchers to figure out the final solution for further analysis. The cluster validation technique is classified into two parts internal and external validation. In this section, we review some of the validation techniques.

(I) **Internal validation:** Internal validation evaluates the clustering solution with the help of the given partition and the data, without using any external information. A rich variety of validation indices can be found in the literature. Next, we review a collection of such indices.

Mean Squared Error: Within-cluster dispersion is measured by Mean Squared Error (MSE) [1]. Let, us consider that the clusters of any algorithm is $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$. If a cluster $\mathcal{C}_i$ has p number of genes and n number columns of each gene then MSE of a cluster is calculated using Equation 2.2.27.

$$MSE(\mathcal{C}_i) = \frac{1}{n \times p} \sum_{j=1}^{p} ||\overrightarrow{g_j} - \overrightarrow{z}|| \tag{2.2.27}$$

$\overrightarrow{g_p}$ is the expression profile of p^{th} gene in this cluster, $\overrightarrow{z}$ is the average expression profiles of all genes belonging to that cluster, and $||\overrightarrow{g_i} - \overrightarrow{z}||$ is the euclidean distance between these two vectors.

Davies Bouldin: The Davies Bouldin (DB) [75, 84] is the mean value of all clusters and is defined as in Equation 2.2.28, where δ_i is intra cluster distance whereas Δ_{ij} is the inter cluster distance between clusters $\mathcal{C}_i$ and $\mathcal{C}_j$.

$$DB(\mathcal{C}) = \frac{1}{K}\sum_{i=1}^{K} max_{j\neq i}(\frac{\delta_i + \delta_j}{\Delta_{ij}}) \tag{2.2.28}$$

The intra cluster distance is computed by the data belonging to a cluster $\mathcal{C}_j$ to their barycenter $\mathcal{O}^{\{j\}}$ which is a row vector as given below, where cluster $\mathcal{C}_j$ can be represented by a submatrix $M^{\{j\}}$ and I_j is set of indices of all genes present in this cluster.

$$\delta_j = \frac{1}{|\mathcal{C}_j|}\sum_{a\in I_j} ||M_a^{\{j\}} - \mathcal{O}^{\{j\}}|| \tag{2.2.29}$$

The submatrix $M^{\{j\}}$ can also be denoted as $M_{\{I_j\}}$. The inter cluster distance Δ_{ij} is the distance between the centroids $\mathcal{O}_i$ and $\mathcal{O}_j$ of clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ as presented in Equation 2.2.30.

$$\Delta_{ij} = d(\mathcal{O}^{\{i\}}, \mathcal{O}^{\{j\}}) = ||\mathcal{O}^{\{j\}} - \mathcal{O}^{\{i\}}|| \tag{2.2.30}$$

Ball-Hall: The Ball-Hall (BH) index measures the mean dispersion of a cluster which formally means the squared distance of the data residing in a cluster to their centroid [26, 84]. The mathematical formula of Ball-Hall for mean through all clusters is indicated by Equation 2.2.31, where n_k is the cardinality of a cluster.

$$BH(\mathcal{C}) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{b\in I_k} ||M_b^{\{k\}} - \mathcal{O}^{\{k\}}||^2 \tag{2.2.31}$$

Let us consider the total number of distinct pairs in the dataset be $N_T = \frac{N(N-1)}{2}$, where $N = \sum_{k=1}^{K} n_k$. In a cluster $\mathcal{C}_k$, the number of distinct pairs is $\frac{n_k(n_k-1)}{2}$. Therefore, the total number of such pairs is N_W shown in Equation 2.2.32.

$$N_W = \sum_{k=1}^{K}\frac{n_k(n_k-1)}{2} \tag{2.2.32}$$

C index: The C index (CI) is defined as mentioned in Equation 2.2.33 [84, 146].

$$CI = \frac{S_W - S_{min}}{S_{max} - S_{min}} \tag{2.2.33}$$

Inside a cluster, the sum of N_W distances between all pairs of points is denoted

by S_W. Among all N_T pairs in the entire dataset, S_{min} takes the sum of N_W smallest distances whereas S_{max} takes the sum of N_W largest distances.

Dunn's index: Dunn's index (DI) is the quotient of minimum distance among data points of the different clusters to the maximum within-cluster distance [92]. It is defined in Equation 2.2.34.

$$DI = \frac{min_{k\neq k'}(min_{i\in I_k, j\in I_{k'}}||M_i^{\{k\}} - M_j^{\{k'\}}||)}{max_{1\leq k\leq K}(max_{i,j\in I_k, i\neq j}||M_i^{\{k\}} - M_j^{\{k\}}||)} \tag{2.2.34}$$

Xie Beni: Xie Beni (XB) index is proposed for fuzzy clustering, however, it is also used for crisp clustering [344]. XB index is the ratio of mean quadratic error to minimum minimal squared error distances between data points belonging to a cluster. XB can be written by Equation 2.2.35, where $\mathcal{O}^{\{k\}}$ is the barycenter of the cluster.

$$XB = \frac{1}{N}\frac{\sum_{i\in I_k}||M_i^{\{k\}} - \mathcal{O}^{\{k\}}||}{min_{k,k'}(min_{i\in I_k, j\in I_{k'}}Dist(M_i, M_j)^2)} \tag{2.2.35}$$

(II) **External validation:** To test the performance of cluster solution, external validation can be used if the "ground truth" of cluster structure is present. In this section, we will present some of the well known and popularly used external measures in the literature. Let, us again consider the clustering results $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$. We can construct a binary matrix $\mathbb{B}$ of size $m \times m$, where m is the number of genes. If two genes g_i and g_j belong to the same cluster then $\mathbb{B}_{ij} = 1$, otherwise $\mathbb{B}_{ij} = 0$. In the similar manner we can create a binary matrix $\mathbb{P}$ for "ground truth" also $\mathbb{P} = \{\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_s\}$. Further, we can represent the following notations:

(a) z_{00} denotes the number of genes pairs (g_i, g_j), where $\mathcal{C}_{ij} = 0$ and $\mathbb{P}_{ij} = 0$.

(b) z_{01} represents the number of genes pairs (g_i, g_j), where $\mathcal{C}_{ij} = 0$ and $\mathbb{P}_{ij} = 1$.

(c) z_{10} is the number of genes pairs (g_i, g_j), where $\mathcal{C}_{ij} = 1$ and $\mathbb{P}_{ij} = 0$.

(d) z_{11} shows the number of genes pairs (g_i, g_j), where $\mathcal{C}_{ij} = 1$ and $\mathbb{P}_{ij} = 1$.

Here, $z_{00}+z_{01}+z_{10}+z_{11} = Z$ which is the maximum number of all pairs in dataset and $Z = \frac{m(m-1)}{2}$, where m is the total number of data points. Aforementioned notations are used to define following indices to measure the degree of similarity.

Rand index:

$$RI = \frac{z_{11} + z_{00}}{z_{00} + z_{01} + z_{10} + z_{11}} \tag{2.2.36}$$

Jaccard coefficient:

$$Jc = \frac{z_{11}}{z_{01} + z_{10} + z_{11}} \tag{2.2.37}$$

Adjusted Rand index:

$$ARI = \frac{z_{11} - \frac{(z_{11}+z_{10})(z_{11}+z_{01})}{Z}}{\frac{(z_{11}+z_{10})(z_{11}+z_{01})}{2} - \frac{(z_{11}+z_{10})(z_{11}+z_{01})}{Z}} \tag{2.2.38}$$

Minkowski measure:

$$Minkowski = \sqrt{\frac{z_{10} + z_{01}}{z_{11} + z_{01}}} \tag{2.2.39}$$

Folkes and Mallows index:

$$FM = \sqrt{(\frac{z_{11}}{z_{11} + z_{10}}) \times (\frac{z_{11}}{z_{11} + z_{01}})} \tag{2.2.40}$$

Rand index (RI) [281] and Jaccard coefficient (Jc) as shown in Equations 2.2.36 and 2.2.37, respectively estimate the agreement (consistent classification) between $\mathcal{C}$ and $\mathbb{P}$. The value of both the metrics lies between 0 and 1. Both the metrics give maximum value when $K = s$. The rationale behind the Jc is essentially the same as RI, except the absence of the term z_{00}. RI gives importance to z_{00} and z_{11}, thus it can not make distinctions between two objects whether they are separated or joined in both the original and evaluated partition [324]. This observation suggests removing the term z_{00} in Jc. One of the main disadvantages of RI is that it is not "corrected for chance", which means RI does not give 0 while considering random partitioning [324]. To overcome this shortcoming, Hubert and Arabie [145] have proposed Adjusted Rand index (ARI) as defined in Equation 2.2.38. ARI is actually normalized to get 1 value when the partition is perfect and 0 while the partition is by chance. Minkowski is defined in Equation 2.2.39, which measures the proportion of disagreement to the total number of gene pairs (g_i, g_j) where both the genes share the same original class [164]. Minkowski also ignores z_{00} term, hence like Jc, Minkowski is useful in gene-based clustering. The term z_{00} dominate the remaining three terms in "good" and "bad" solutions. Equation 2.2.40 represents Folkes and Mallows index (FM). Higher FM value signifies better clustering result and exhibits strong similarity in between $\mathcal{C}$ and $\mathbb{P}$.

Biological validation: Another important part of external evaluation is biological validation. The necessity of biological validation is to check whether the

genes belonging to the same cluster are biologically related or not. It functionally analyzes the identifying biological function based on functional annotation on the clustered data. Clusters should not form by chance, they should reflect the biological relevancy and reliability [275]. To determine the reliability it is necessary to validate using external biological databases and compare with other algorithms [275]. For comparing the effectiveness of various clustering algorithms the standard methodology, enrichment analysis of groups of genes based on GO and KEGG pathway enrichment has been popularly used. One of the applications of cluster analysis is gene function prediction.

Enrichment analysis assigns a biological name to a set of genes. Previously, to assign the function of a gene, each gene product is studied individually, but nowadays the job is easier with existing tools. GO database allows the researcher to assign attributes on gene sets, which is generated by clustering methods. It helps biologists to infer about the groups of genes rather than investigating genes individually. To statistically validate the GO terms associated with a group of genes, the p-value is used. According to Gene Ontology Consortium p-value is defined as "the probability or chance of seeing at least x number of genes out of the total z genes in the list annotated to a particular GO term, given the proportion of genes in the whole genome that are annotated to that GO Term". This is calculated using Equation 2.2.41 as follows, where A represents the total number of genes available in the genome and f is the genes present in the functional category.

$$p = 1 - \sum_{i=0}^{x-1} \frac{\binom{f}{i}\binom{A-f}{z-i}}{\binom{A}{z}} \tag{2.2.41}$$

Smaller probability (less than significant cut-off) indicates a more significant cluster. KEGG pathway analysis is used to find the clusters of co-expressed genes which share the same pathway.

We have surveyed various indices regarding the internal and external validations. But, there are several others validation indices. We provide further reference for more validation indices [84]. Above all, validation still remains the most challenging task. Jain and Dubes [154] state in their book on clustering that "The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage". Despite the success of clustering methods in different fields, the above-mentioned line is still true after 30 years. There is no well-accepted method that can be used to serve this purpose. Hence, there is always a demand

for improvements and new validation methods. But this is not the scope of our thesis.

2.3 Biclustering

Even though the advantages of using clustering algorithms for expression data are many, it still suffers from limitations. The inefficiency of traditional clustering algorithms in extracting local structures inherent in the data due to their focus on finding global patterns has given rise to biclustering algorithms and is highly explored for analyzing transcriptomics data.

Definition 2.3.1 *From a given gene expression data* $ED_{m \times n}$*, a bicluster* $\beta(\mathcal{I}, \mathcal{J}) = \{ge_{ij}, i \in \mathcal{I}, j \in \mathcal{J}\}$ *(submatrix) corresponds to a subset of rows* $\mathcal{I} \subseteq G$ *showing some coherent tendency under a subset of conditions* $\mathcal{J} \subseteq C$*.*

The main goal of biclustering algorithms is to find a set of biclusters $\beta_k(\mathcal{I}_k, \mathcal{J}_k)$, for $k = \{1, 2, ..., K\}$, where K is the total number of biclusters.

2.3.1 Bicluster types

The key feature of a biclustering algorithm is to identify various types of biclusters, viz. (i) constant valued bicluster, (ii) row-constant valued bicluster, (iii) column-constant valued bicluster, (iv) coherent valued bicluster, and (v) coherent evolution bicluster. Figure 2.1 depicts different bicluster types.

(i) A *constant valued bicluster* exhibits similar expression values in a group of genes under a group of conditions. A *perfect constant bicluster* $\beta(\mathcal{I}, \mathcal{J})$ has equal values (*const*) in every cell of the bicluster, as given by Equation 2.3.1.

$$\beta(\mathcal{I}, \mathcal{J}) = \{ge_{ij} = const, i \in \mathcal{I} \text{ and } j \in \mathcal{J}\} \tag{2.3.1}$$

(ii) A *constant row bicluster* $\beta(\mathcal{I}, \mathcal{J})$ has same value for all the elements of each row. A *perfect constant row bicluster* can be additive (shifting) as given by Equation 2.3.2 or multiplicative (scaling) as given by Equation 2.3.3, where a_i and b_i are the additive and multiplicative coefficients of i^{th} row, respectively. Shifting pattern of a gene expression row can be obtained by both addition or subtraction of constant values from the pattern.

$$\beta(\mathcal{I}, \mathcal{J}) = \{ge_{ij} = const \pm a_i, i \in \mathcal{I} \text{ and } j \in \mathcal{J}\} \tag{2.3.2}$$

$$\beta(\mathcal{I}, \mathcal{J}) = \{ge_{ij} = const \times b_i, i \in \mathcal{I} \text{ and } j \in \mathcal{J}\} \tag{2.3.3}$$

(iii) A *constant column bicluster* $\beta(\mathcal{I}, \mathcal{J})$ has the same value for all the elements of each column. A *perfect constant column bicluster* can be represented by Equation 2.3.4 or Equation 2.3.5, where c_j and d_j are the additive and multiplicative coefficients of j^{th} column, respectively.

$$\beta(\mathcal{I}, \mathcal{J}) = \{ge_{ij} = const + c_j, i \in \mathcal{I} \text{ and } j \in \mathcal{J}\} \tag{2.3.4}$$

$$\beta(\mathcal{I}, \mathcal{J}) = \{ge_{ij} = const \times d_j, i \in \mathcal{I} \text{ and } j \in \mathcal{J}\} \tag{2.3.5}$$

(iv) *Coherent valued bicluster* captures the complex relationship between subset of genes in association with subset of columns. The coherent values of the additive pattern can be expressed in Equation 2.3.6, where each row and column has some shifting coefficients a_i and c_j, respectively. The notation for multiplicative pattern is defined in Equation 2.3.7, where each row and column has some scaling coefficient b_i and d_j, respectively. Coherent valued bicluster also may include both the additive and multiplicative patterns together, where each row and column has both shifting and scaling coefficients. Mathematical formula of a perfect additive multiplicative pattern is shown in Equation 2.3.8.

$$\beta(\mathcal{I}, \mathcal{J}) = \{ge_{ij} = const + a_i + c_j, i \in \mathcal{I} \text{ and } j \in \mathcal{J}\} \tag{2.3.6}$$

$$\beta(\mathcal{I}, \mathcal{J}) = \{ge_{ij} = const \times b_i \times d_j, i \in \mathcal{I} \text{ and } j \in \mathcal{J}\} \tag{2.3.7}$$

$$\beta(\mathcal{I}, \mathcal{J}) = \{ge_{ij} = const \times b_i \times d_j + a_i + c_j, i \in \mathcal{I} \text{ and } j \in \mathcal{J}\} \tag{2.3.8}$$

Equation 2.3.8 corresponds to the most general situation, while the other mathematical notations discussed before are only the special cases that can be easily derived from it.

(v) A *coherent evolution bicluster* captures the subgroup of *up-regulated* or *down-regulated* genes under a subgroup of conditions regardless of the exact expression values. This type of pattern does not obey any mathematical formula.

2.3.2 Bicluster structures

Biclustering algorithms can also be classified based on their underlying structure, i.e., how the rows and columns are incorporated in the formation of the bicluster from the input data. Pontes et al. [277] define the structure of a bicluster according to the following categories.

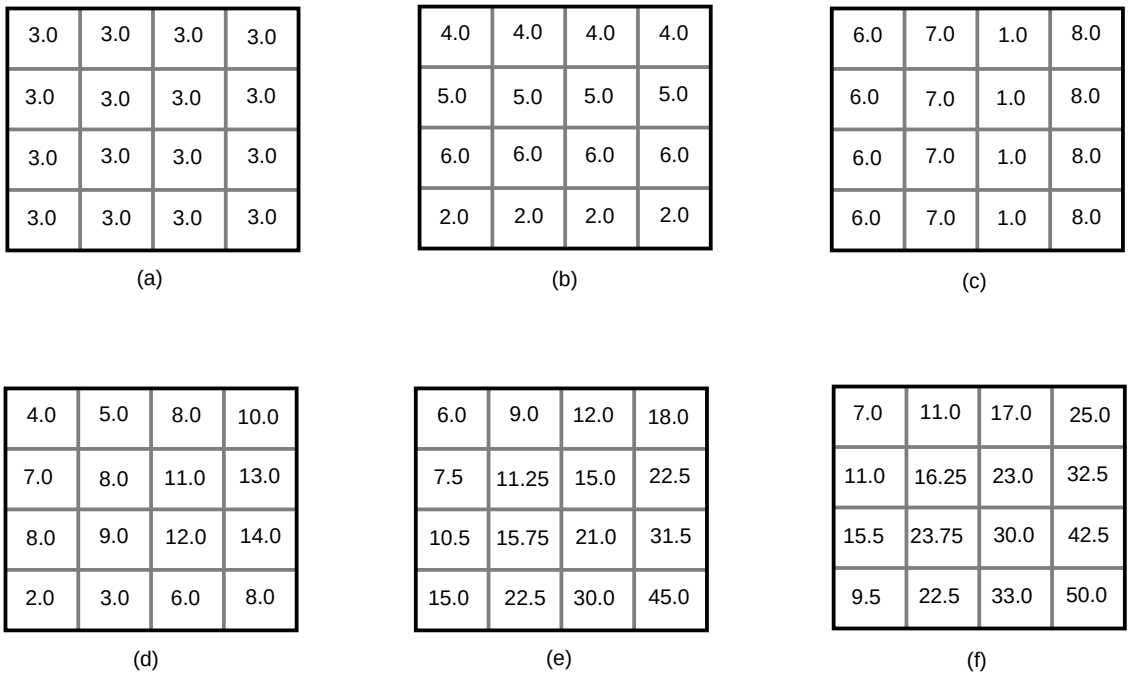


Figure 2.1: Example of different types of biclusters. (a) Constant, (b) Row-constant, (c) Column-constant, (d) Additive, (e) Multiplicative, and (f) Additive-multiplicative patterns.

(i) **Row exhaustive:** Every gene should reside in at least one bicluster.

(ii) **Column exhaustive:** Every condition should belong to at least one bicluster.

(iii) **Non-exhaustive:** Genes and conditions may not be a member of any bicluster.

(iv) **Row exclusive:** Each gene in the data matrix may belong to atmost one bicluster (Figure 2.2 (c)).

(v) **Column exclusive:** Each condition in the data matrix may belong to atmost one bicluster (Figure 2.2 (d)).

(vi) **Non-exclusive:** In this category the obtained biclusters can be overlapped i.e., more than one biclusters share genes or conditions.

An ideal structure of a bicluster should be non-exclusive or non-exhaustive. Beside these major structures biclusters can be further divided into single bicluster (Figure 2.2 (a)), non-overlapping biclusters with tree (Figure 2.2 (e)), checkerboard structure (Figure 2.2 (f)), non-overlapping biclusters non-exclusive group of biclusters (Figure 2.2 (g)), overlapping biclusters having hierarchical structure (Figure 2.2 (h)), and arbitrary positioned biclusters (Figure 2.2 (i)) [227].

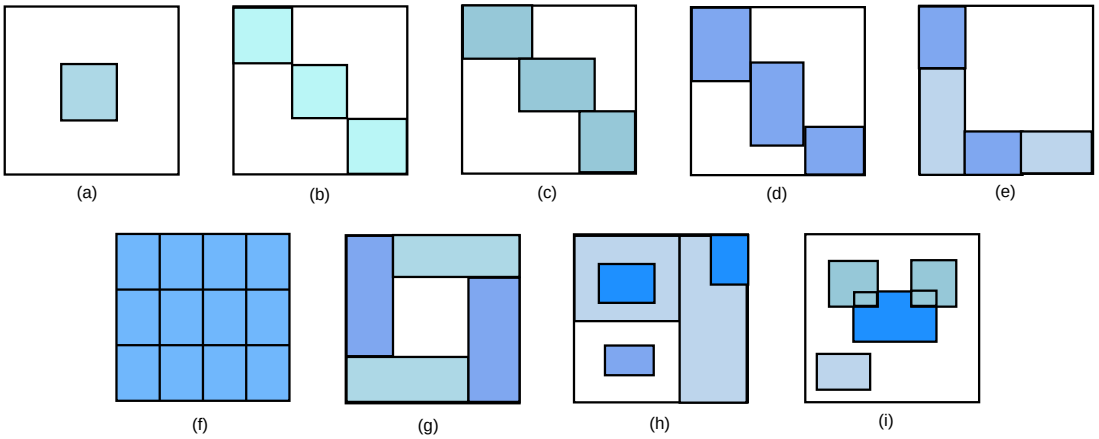


Figure 2.2: The structure of biclusters (a) Single bicluster, (b) Exclusive rows and columns, (c) Exclusive rows, (d) Exclusive columns, (e) Non-overlapping with tree structure, (f) Checkerboard structure, (g) Non-exclusive non-overlapping, (h) Overlapping biclusters with hierarchical structure, and (i) Arbitrary positioned overlapped biclusters.

2.3.3 Biclustering algorithms

Biclustering is an optimization problem. Most of the algorithms try to search the patterns using different heuristic strategies or searching criteria instead of exhaustive algorithms. Therefore, it has been proved to be an NP-hard problem. The term biclustering is first introduced by Hartigan [129]. Morgan and Sonquist [248] and Hartigan [129] have proposed the concept of data partitioning into submatrices with approximately constant values. Cheng and Church (C&C) [64] coined the concept of biclustering algorithm in the context of gene expression data in the year 2000. After that, a good number of biclustering algorithms have been developed which can be found in several survey papers [99, 112, 199, 227, 251, 262, 277, 287, 315]. However, all biclustering algorithms are not equally effective.

Most authors categorize the biclustering algorithms considering various criteria like the type of biclusters generated, bicluster patterns, the structure of biclusters, searching methods used [227], a mathematical formula used to identify different types of biclusters [262], evaluation measures applied [277] and type of input matrices [110]. Two kinds of surveys have been reported, one based on the theoretical perspective [112, 227, 251, 277, 287, 315] and the other based on empirical study or quantitative comparison [66, 99, 199, 262, 266, 278]. From the vivid description of biclustering, these algorithms can be broadly categorized into five distinct types, (I) Greedy iterative search (GIS), (II) Graph-based (GB), (III) Divide and conquer (DAC), (IV) Linear algebraic, and (V) Distribution

parameter identification (DPI) approach.

(I) **Greedy iterative search:** The basic idea behind the GIS approach is to create biclusters by addition/removal of rows(genes)/columns(conditions) to/from the data matrix using certain optimization function [227]. This strategy is well suited for large scale data [251] and gives results within a satisfactory amount of time. The notable drawback of this approach is to get stuck into the local optima due to the beginning configuration. This type of heuristics never ensures to get a global optimum solution. So, it may miss some of the good biclusters. C&C [64], FLOC (Flexible Overlapped biClustering) [353], OPSM (Order-Preserving Sub-Matrices) [32], xMOTIFs (Conserved gene expression Motifs) [252], Maximum Similarity Bicluster (MSB) [217], S4VD (Sparse Singular Stability Selection Value Decomposition) [302], and UniBic [337] biclustering algorithms follow GIS approach.

C&C [64] seeks to find a list of K number of biclusters with low variance from a given expression matrix by evaluating the quality of biclusters using Mean Square Residue (MSR) [64]. This coherence measure of a bicluster $\beta(\mathcal{I}, \mathcal{J})$ is defined by Equation 2.3.9, where $ge_{i\mathcal{J}} = \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} ge_{ij}$, $ge_{\mathcal{I}j} = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} ge_{ij}$, and $ge_{\mathcal{I}\mathcal{J}} = \frac{1}{|\mathcal{I}|\times|\mathcal{J}|}\sum_{i\in\mathcal{I},j\in\mathcal{J}} ge_{ij} = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} ge_{i\mathcal{J}} = \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} ge_{\mathcal{I}j}$ is column subset average, row subset average, and submatrix average, respectively.

$$MSR(\beta) = \frac{1}{|\mathcal{I}| \times |\mathcal{J}|} \sum_{i\in\mathcal{I},j\in\mathcal{J}} (ge_{ij} - ge_{\mathcal{I}j} - ge_{i\mathcal{J}} + ge_{\mathcal{I}\mathcal{J}})^2 \tag{2.3.9}$$

MSR metric successfully finds constant, row-constant, column-constant, and coherent additive patterns only. It is not suitable for additive-multiplicative patterns [3, 47]. The algorithm begins with the whole matrix and is done with two phases: deletion phase and addition phase to get a single bicluster. In the first phase, the matrix starts removing rows and columns in order to keep the MSR value less than δ (threshold) i.e., to get the high residue of the matrix. In the next step, the rows or columns are added back to the matrix such that the MSR value does not increase. The values of the recently identified bicluster are substituted with random values to prevent the overlapping among the biclusters and this matrix is again used in C&C algorithm. To get K biclusters the procedure is repeated for K times. The drawback of C&C algorithm is masking with random values after finding one bicluster, the threshold value (depends on the dataset), discovering one bicluster at a time. A bicluster is called δ-bicluster if $MSR \leq \delta$, where $\delta \geq 0$. Yang et al. [351, 353] propose a δ-cluster which captures the strong coherence between a subset of genes across a subset of conditions rather than

considering the physical closeness via minimizing the residue value. It allows the missing values and uses random masking after discovering a bicluster using C&C algorithm.

Authors also contribute a move-based algorithm called FLOC to identify high accuracy biclusters. An improved version of the FLOC algorithm can be found in [352] using probabilistic FLOC algorithm. FLOC follows the same algorithmic strategy as C&C. It starts with a set of initial biclusters and iteratively improves the quality of biclusters in terms of lower MSR value by addition and removal of row and column at a time. The algorithm stops when no change occurs which improves the overall quality of the biclusters. This algorithm prefers bigger biclusters.

Ben Dor et al. in [32] searches for OPSM i.e., a subset of genes having the same order for a subset of experimental conditions. In other words, a bicluster is defined as a set of expression values having the same linear ordering across a subset of columns. A greedy method is applied to find highly statistically significant and large biclusters.

Murali and Kasif [252] propose a representation of gene expression data named xMOTIFs where a subset of genes are simultaneously conserved under a subset of conditions in the discretized data [252]. A gene expression level is said to be conserved under a set of samples when the expression level is in the same state across all the samples. The main goal of xMOTIFs is to find genes having the same state under the samples of the seeds (selection of columns randomly) and the discriminating set (selection of a set of columns randomly). For every gene, a list of intervals representing the states is determined using the statistical significance of intervals in accordance with uniform distribution. The algorithm adds some factors to its definition i.e., size, conservation, and maximality in order to avoid too large or too small biclusters.

Liu and Wang [217] propose a polynomial time algorithm which can detect additive and constant optimal biclusters with maximum similarity score, known as MSB. The algorithm contributes several advantages- (i) no requirement of discretization of the original dataset, (ii) good performance on overlapping clusters, and (iii) doing well in additive biclusters. MSB starts with the whole data matrix as a bicluster and iteratively removes row or column in order to find biclusters with a maximum similarity score. The process continues until one element is left in the bicluster. The limitation of the MSB algorithm is that it works only with square biclusters. To overcome this drawback, the authors presented another algorithm known as RMSBE (Randomized Maximum Similar-

ity Bicluster Extension) [217] which is relatively faster than MSB. The algorithm calculates the average similarity score via multiple scans of data for identifying biclusters as well as the selection of reference genes.

S4VD is proposed by Sill et al. [302], where stability selection method was incorporated in Sparse Singular Value Decomposition (SVD) approach to improve the quality of biclusters. Stability selection is actually a variable section based on sub-sampling which controls type I error rate. The sub-sampling method helps to obtain stable biclusters and estimates the selection probability of genes and conditions to be present in the biclusters.

The key principle of the UniBic [337] algorithm is to apply Longest Common Subsequence (LCS) between a selected pair of rows over an indexed matrix which is a result of permutation of columns on input matrix for locating the seed of each bicluster. Then, the algorithm expands the bicluster by adding columns/rows to get approximately trend preserving bicluster. The number of biclusters is specified by the user.

(II) **Graph-based:** Graph-based algorithms normally represent the input data as a bipartite graph consisting of two sets of vertex conditions and genes, respectively. The edge of a graph denotes the over-expression or under-expression of a gene under certain conditions. The problem can be formulated to find the subgraph from a bipartite graph. Before applying an algorithm the dataset must be discretized. It also doesn't work well for large datasets. In this section, we describe different graph-based biclustering algorithms, SAMBA (Statistical-Algorithmic Method for Bicluster Analysis) [314], QUBIC (QUalitative BI-Clustering algorithm) [196], and HOCCLUS2 (Hierarchical Overlapping Co-CLUStering2) [273].

A polynomial-time algorithm SAMBA [314] is based on a graph-theoretic approach coupled with statistical considerations and guaranteed to find a subgroup of genes jointly responding across a subgroup of conditions. In the SAMBA framework, input gene expression data is modeled as a bipartite graph, where one set of vertices is responsible for genes, another set of vertices corresponding to the conditions and edges are referred to as significant expression changes. Two statistical models, a simple model and a refined model is presented here for the resulting graph. The weights of the vertex pairs are assigned according to the aforementioned probabilistic model, which assures finding the maximum weight subgraph corresponding to a significant bicluster with maximum likelihood.

QUBIC [196] can solve a biclustering algorithm utilizing both qualitative

(or semi-qualitative) measures of expression data and a combinatorial optimization technique. QUBIC [196] can identify statistically significant biclusters and both positive as well as negatively correlated expression patterns. In this method, the gene expression data is represented in an integer-valued matrix. The algorithmic framework starts by constructing a weighted graph. The weight of an edge is stated as the number of columns where two genes have an identical nonzero integer. The goal of the QUBIC algorithm is to find the maximal bicluster with a higher consistency level. The consistency level is interpreted as the minimum ratio between the number of identical nonzero integers in a column and the total number of rows in the submatrix. The biclusters are identified from the recently constructed weighted graph one by one. The algorithm begins with the initial bicluster based on seeds which is initially a set of the sorted list of edges and then go on expanding iteratively in both the horizontal and vertical directions without violating the consistency level.

HOCCLUS2 [273] can be described as the extraction of initial bicliques. The steps involved in this algorithm are (i) extraction of non hierarchically organized bicluster, (ii) overlap identification and merging, and (iii) providing ranks to the extracted biclusters.

(III) **Divide and Conquer:** The strategy of DAC is to divide the whole problem into smaller subproblems with a similar structure to the actual problem until the subproblems are sufficiently small enough. The solutions of the subproblems are merged to get the result of the original problem. The advantage of the DAC approach is very fast but the problem regarding this approach is that it can miss some significantly good biclusters when it splits the data before it can be identified. One of the leading examples of DAC is BiMax [278]. Here, before applying the main algorithm, the input matrix needs to be binarized. It finds the bicluster as a rectangle of 1's from the binary matrix. BiMax initiates the whole matrix and starts dividing the matrix into checkerboard format.

A novel Bit-Pattern Biclustering algorithm (BiBit) has been proposed to extract biclusters from a binary matrix [284]. The algorithmic strategy of BiBit is to apply the AND operator over all possible row pairs so that it can find maximal biclusters.

(IV) **Linear algebraic:** ISA (Iterative Signature Algorithm) [152] and BicSPAM (Biclustering based on Sequential PAttern Mining) [134] utilize linear algebra with the help of vector space and linear mapping to find the correlated submatrix from the given input.

Bergmann et al. [152] propose a biclustering algorithm ISA and define a significant bicluster as a transcription module identified from the given gene expression data matrix. A set of co-regulated genes and a set of experimental conditions makes a self-consistent regulatory unit called a transcription module. The algorithm uses two thresholds: the average value of each column of a bicluster should be above T_C and the average value of each row of a bicluster should be above T_G, where T_C and T_G are two thresholds. The signature algorithm, Singular Value Decomposition (SVD) is employed to identify the transcription module. The algorithm starts with a randomly selected set of rows or columns as an initial seed bicluster. Iteratively it keeps on updating the rows and columns until it meets the definition of the transcription module. ISA can identify one bicluster per iteration. It can find overlapped, down-regulated, and up-regulated biclusters.

Recently, Henriques and Madeira have proposed pattern-based algorithms namely BicSPAM [134]. The algorithm identifies flexible structures for order-preserving biclusters which allow for symmetries and is robust towards the noise. The basic steps of both the algorithms are mapping, mining (preprocessing), and closing (post processing).

(V) **Distribution parameter identification:** DPI approach assumes a statistical model for identifying the distribution of parameters which is used to generate data by minimizing some criteria iteratively. Biclustering algorithms which adopt DPI technique are Plaid model [321], FABIA (Factor Analysis for BIcluster Acquisition) [141], and iBBiG (iterative Binary Bi-clustering of Gene sets) [124].

The plaid model is proposed by Lazzeroni and Own [188], is a tool for exploratory analysis of multivariate data. Turner et al. [321] propose an improvement version of plaid model mentioned in [188]. In the plaid model, the expression value of the data model is considered to be the sum of terms known as layers (biclusters). The expression level of gene i under sample j, ge_{ij} is defined by Equation 2.3.10, where K is the number of biclusters, k is the layer index, θ_{ij0} indicates background layer, ϵ_{ij} is residual error, θ_{ijk} denotes the sum of mean, sample and gene which effects in k layer, ρ_{ik} is a binary cluster membership parameter, κ_{jk} denotes cluster membership sample j. ρ_{ik} is 1 if i^{th} gene belongs to k^{th} bicluster otherwise it is 0.

$$ge_{ij} = \theta_{ij0} + \sum_{k=1}^{K} \theta_{ijk}\rho_{ik}K_{jk} + \epsilon_{ij} \tag{2.3.10}$$

In order to find K biclusters, the authors propose a greedy algorithm that adds one layer at a time. The algorithm iteratively updates each of the parameters of the model to minimize the MSR value between the true data and data matrix to the model. It assumes that the residual becomes unstructured noise when each layer is removed from the data. The authors propose a very simple rule to terminate the algorithm that is a small number of extra layers can be extracted when the data is reduced to noise.

FABIA is proposed by Hochreiter et al. [141]. The algorithm uses a multiplicative model which captures linear dependencies among a subset of genes and a subset of conditions. In the FABIA model, data matrix X is a sum of p biclusters and additive noise Υ is represented by Equation 2.3.11, where each bicluster is a product of λ row and z column vectors. The dimension of these parameters are $X \in \mathbb{R}^{m \times n}$, $\Upsilon \in \mathbb{R}^{m \times n}$, $\lambda_i \in \mathbb{R}^m$, and $z_i \in \mathbb{R}^n$. $\Lambda \in \mathbb{R}^{m \times p}$ is the sparse prototype matrix and $Z \in \mathbb{R}^{p \times n}$ is the sparse factor matrix.

$$X = \sum_{i=1}^{p} \lambda_i z_i^T + \Upsilon = \Lambda Z + \Upsilon \tag{2.3.11}$$

Gusenleitner et al. [124] propose a genetic algorithm based biclustering iBBiG, which works on binary gene set profile and extracts modules or biclusters maximizing the size and entropy of each bicluster. The identified modules from the iBBiG algorithm are ranked by information score and the gene set of each of the modules are ranked by fitness score measuring the weight in the module. The advantage of iBBiG is that it does not require the prior information of the number or the size of biclusters and discovers overlapping biclusters even in the presence of noisy data. The algorithm consists of three main components: (i) fitness score for a module, (ii) a heuristic search algorithm for module identification and growth in a high dimensional search space, and (iii) an iterative extraction method for masking the signal of modules which are discovered previously.

Shi et al. [297] introduce an algorithm, known as LinCoh which identifies linear coherent patterns from microarray gene expression data. The concept behind this algorithm is to identify a non-trivial sample set for each pair of genes, called the outer sample set. It has been seen that two genes can be up/down co-regulated under a non-trivial subset of samples. In the next step, a finer bicluster is produced by removing genes and inner samples for each outer sample set of biclusters. A line detection version of LinCoh [297] algorithm can be found in [296]. Shi et al. [298] develop another biclustering algorithm called,

Sparse Learning based Linear Coherent Bi-clustering (SLLB). This algorithm discovers linear coherent patterns using sparse learning based method. Deodhar et al. [82] propose a scalable, robust algorithm for discovering extremely dense cluster, named Robust Overlapping CoClustering (ROCC).

2.3.4 Bicluster evaluation methods

In Section 2.3.3, we have reviewed some of the existing biclustering algorithms. These algorithms discover different bicluster types, structures considering different objective functions. Therefore, it is important to validate biclusters which remains a challenging task till today. Unlike full-space clustering algorithms, biclustering algorithms are evaluated based on only external or biological validation instead of internal or statistical validation. To assess the quality of a biclustering algorithm, we broadly divide the bicluster evaluation into two main classes statistical validation and biological validation.

(I) **Statistical significance:** Statistical significance is useful for validation of synthetic datasets and uses information of data to judge the quality of biclusters [287]. The several aspects of bicluster validation are given below.

(a) **Separation:** The measure quantifies the degree of separation between biclusters i.e., how well biclusters are separated from each other. The separation between two biclusters β_1 and β_2 can be defined by Equation 2.3.12.

$$Sep(\beta_1, \beta_2) = 1 - \frac{\beta_1 \cap \beta_2}{\beta_1 \cup \beta_2} \tag{2.3.12}$$

(b) **Compactness:** Cluster homogeneity with intra-cluster validation signifies bicluster compactness. Intra-bicluster defines the degree of coherence of each bicluster [112]. Several intra-bicluster evaluation function can be found in the literature. Consider a bicluster $\beta(\mathcal{I}, \mathcal{J})$ containing $\mathcal{I}$ rows and $\mathcal{J}$ columns, where $i \in \mathcal{I}$ and $j \in \mathcal{J}$. Next, we present some of the existing evaluation functions.

VARiance: Hartigan [129] has proposed the coherence measure VARiance (VAR) in order to minimize the sum of bicluster variances. It is noteworthy that VAR is used to detect constant biclusters. Mathematically, it is defined in Equation 2.3.13.

$$VAR(\mathcal{I}, \mathcal{J}) = \sum_{i=1}^{|\mathcal{I}|} \sum_{j=1}^{|\mathcal{J}|} (ge_{ij} - ge_{\mathcal{I}\mathcal{J}})^2 \tag{2.3.13}$$

AVerage Similarity Score: The AVerage Similarity Score (AVSS) of a bicluster is proposed by Liu and Wang [217] which is calculated by Equation

2.3.14, where s_{ij} is the similarity measure of i^{th} row and j^{th} column with all other objects belong to that bicluster.

$$AVSS(\mathcal{I},\mathcal{J}) = \frac{\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} s_{ij}}{|\mathcal{I}|.|\mathcal{J}|} \tag{2.3.14}$$

Mean Square Residue: The most well-known intra-bicluster measure is Mean Square Residue (MSR) which is proposed by Cheng and Church [64]. MSR measures the coherence among genes and columns are defined as in Equation 2.3.9. Lower (close to 0) the MSR value (resp. higher), exhibits the stronger (resp. weaker) coherence in the bicluster. For a perfect bicluster, the MSR value is 0 which means genes fluctuate similarly under a subset of conditions. Various algorithms adopt MSR [14, 63, 87, 351]. Mean square residue is good for capturing the constant and the shifting patterns from the input data.

Scaling Mean Square Residue: Nevertheless, MSR is widely used in literature but it is unable to recognize scaling patterns in the bicluster. To overcome this problem, Mukhopadhay et al. [250] have proposed a metric named Scaling Mean Square Residue (SMSR) as shown in Equation 2.3.15. This is invariant to the local and global scaling pattern of the data.

$$SMSR(\mathcal{I},\mathcal{J}) = \frac{1}{|\mathcal{I}|\times|\mathcal{J}|}\sum_{i=1}^{|\mathcal{I}|}\sum_{j=1}^{|\mathcal{J}|}\frac{(ge_{i\mathcal{J}}\times ge_{\mathcal{I}j} - ge_{ij}\times ge_{\mathcal{I}\mathcal{J}})^2}{ge_{i\mathcal{J}}^2\times ge_{\mathcal{I}j}^2} \tag{2.3.15}$$

Average Row Variance: Average Row Variance (ARV) is proposed by Anguilli et al. [14] as shown in Equation 2.3.16. It is expected to have large row variance if a bicluster contains rows with large changes for different columns. This measure guarantees to capture the coherent trends of a subset of rows under some experimental conditions.

$$ARV(\mathcal{I},\mathcal{J}) = \frac{\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}(ge_{ij} - ge_{i\mathcal{J}})^2}{|\mathcal{I}|\times|\mathcal{J}|} \tag{2.3.16}$$

SUb-MAtrix Correlation Score: Yang et al. [355] have proposed a PCC based measure, SUb-MAtrix Correlation Score (SCS) with prior assumption that a perfect correlated pattern fulfills perfect linear correlation on row and column vector. Equation 2.3.17 and 2.3.18 are the form of correlation degrees on columns and rows, respectively of a bicluster where $corr(x_{\mathcal{I}j_1}, x_{\mathcal{I}j_2})$ and $corr(x_{i_1\mathcal{J}}, x_{i_2\mathcal{J}})$ represent the PCC between any pair of columns or rows in a

bicluster, respectively.

$$SCS_{col}(\mathcal{I},\mathcal{J}) = min_{j_1\in\mathcal{J}}(1 - \frac{1}{|\mathcal{J}|-1}\sum_{j_2\neq j_1, j_2\in\mathcal{J}}|corr(x_{\mathcal{I}j_1}, x_{\mathcal{I}j_2})|) \quad (2.3.17)$$

$$SCS_{row}(\mathcal{I},\mathcal{J}) = min_{i_1\in\mathcal{I}}(1 - \frac{1}{|\mathcal{I}|-1}\sum_{i_2\neq i_1, i_2\in\mathcal{I}}|corr(x_{i_1\mathcal{J}}, x_{i_2\mathcal{J}})|) \quad (2.3.18)$$

Both the measures SCS_{row} and SCS_{col} are asymmetric. The former one reflects the degree on the rows of the bicluster whereas the latter one shows the degree on the columns of the bicluster. The Submatrix correlation score of a submatrix is defined as Equation 2.3.19.

$$SCS(\mathcal{I},\mathcal{J}) = min(S_{row}(\mathcal{I},\mathcal{J}), S_{col}(\mathcal{I},\mathcal{J})) \quad (2.3.19)$$

Moreover, Yang et al. [355] have also demonstrated δ-corbicluster if the value of SCS is less than some threshold δ where $\delta > 0$. The lower (resp. high) SCS provides a better correlation (resp. weaker) of the rows and columns. If $S(\mathcal{I},\mathcal{J}) = 0$, then it shows the perfect bicluster where rows and columns of the bicluster are linearly correlated.

Volume: Anguilli et al. [14] have proposed Volume (V) as presented in Equation 2.3.20 to maximize the size of a bicluster.

$$V(\mathcal{I},\mathcal{J}) = |\mathcal{I}| \times |\mathcal{J}| \quad (2.3.20)$$

Average Correlation Value: For evaluation of homogeneity of a bicluster, Teng and Chan [318] have developed Average Correlation Value (ACV) in the following way.

$$ACV(\mathcal{I},\mathcal{J}) = max\{\frac{\sum_{i_1\in\mathcal{I}}\sum_{i_2\in\mathcal{I}}|corr_{i_1i_2}| - |\mathcal{I}|}{|\mathcal{I}| \times (|\mathcal{I}|-1)}, \frac{\sum_{j_1\in\mathcal{J}}\sum_{j_2\in\mathcal{J}}|corr_{j_1j_2}| - |\mathcal{J}|}{|\mathcal{J}| \times (|\mathcal{J}|-1)}\} \quad (2.3.21)$$

Here, $corr_{i_1i_2}$ and $corr_{j_1j_2}$ are the Pearson correlation between any pair of rows i_1, i_2 and columns j_1, j_2, respectively. The ACV value ranges from 0 to 1, where 1 signifies that rows and columns in a bicluster are highly co-expressed whereas 0 means none of the genes and conditions is co-expressed. Therefore, it is always preferred to have a higher ACV. It always gives desirable values for additive and multiplicative models in contrast to MSR. ACV is sensitive to errors.

Average Spearman's Rho: Ayadi et al. [20] have proposed an evaluation metric named Average Spearman's Rho (ASR) based on Spearman's cor-

relation coefficient as shown in Equation 2.3.22.

$$ASR(\mathcal{I},\mathcal{J}) = 2 \times max\{\frac{\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{I},j\geq i+1}\rho_{ij}}{|\mathcal{I}|(|\mathcal{I}|-1)}, \frac{\sum_{k\in\mathcal{J}}\sum_{l\in\mathcal{J},l\geq k+1}\rho_{kl}}{|\mathcal{J}|(|\mathcal{J}|-1)}\} \quad (2.3.22)$$

Here, ρ_{ij} and ρ_{kl} refer the Spearman's correlation of pair of rows and columns, respectively. The value of ASR value lies in the interval -1 to 1. A high/low value i.e., close to 1/-1 corresponds that the rows and columns of a bicluster strongly correlated either positively or negatively, respectively. Spearman's rank correlation is robust in the presence of prominent outliers in the data and normalization of the input matrix is not required at all.

Spearman's Biclustering Measure: Spearman's Biclustering Measure (SBM) has been proposed by Flores et al. [110] highlighting non-linear correlation among genes and conditions. SBM has the capability to identify the complex coherence pattern in the biclusters, such as shifting, scaling, and negative correlation. To compute the SBM, the data matrix is converted into rank according to Spearman's coefficient $r_{x,y}$ between any pairs of genes and conditions (x, y). The proposed SBM is defined as follows.

$$SBM(\mathcal{I},\mathcal{J}) = \alpha(B_{ij}) \times r^{\bar{G}}_{B_{\mathcal{I}\mathcal{J}}} \times \beta(B_{ij}) \times r^{\bar{C}}_{B_{\mathcal{I}\mathcal{J}}} \quad (2.3.23)$$

$r^{\bar{G}}_{B_{\mathcal{I}\mathcal{J}}}$ (Equation 2.3.24) and $r^{\bar{C}}_{B_{\mathcal{I}\mathcal{J}}}$ (Equation 2.3.25) express the summarized expression of the trends observed in the genes and conditions of the bicluster, respectively.

$$r^{\bar{G}}_{B_{\mathcal{I}\mathcal{J}}} = \frac{2}{|\mathcal{I}| \times (|\mathcal{I}|-1)} \times \sum_{i=1}^{|\mathcal{I}|}\sum_{i'=i+1}^{|\mathcal{I}|} |r^{G}_{ii'}| \quad (2.3.24)$$

$$r^{\bar{C}}_{B_{\mathcal{I}\mathcal{J}}} = \frac{2}{|\mathcal{J}| \times (|\mathcal{J}|-1)} \times \sum_{j=1}^{|\mathcal{J}|}\sum_{j'=j+1}^{|\mathcal{J}|} |r^{C}_{jj'}| \quad (2.3.25)$$

$|r^{G}_{ii'}|$ and $|r^{C}_{jj'}|$ correspond the absolute value of the Spearman's nonparametric correlation coefficient between genes i, $i^{'}$ and conditions j, $j^{'}$, respectively. Two terms $\alpha(B_{IJ})$ and $\beta(B_{IJ})$ are the reliability coefficients for weighing the influence of the patterns found in the rows and columns, respectively. β has high reliability and is computed from hundreds of genes and whereas α has low reliability and is computed from a small number of samples. The range of SBM is not specified, it may vary from one dataset to another dataset. If the value of SBM increases this means it increases the coherence between the genes and conditions, in other words, it increases the quality of a bicluster.

(c) **Connectedness:** Connectedness assesses how well the given cluster groups data with the nearest neighbor together, in data space.

(d) **Coverage:** This statistical measure is classified into three types viz, Gene coverage (Gc), Condition coverage (Cc), and Matrix coverage (Mc) as shown below. Gene/Condition/Matrix coverage is the ratio of genes/conditions/cells that are assigned to any extracted bicluster to the total number of genes/conditions/cells in an input dataset.

$$\text{Gc} = \frac{\text{Total number of genes covered by discovered biclusters}}{\text{Total number of genes present in the matrix}} \tag{2.3.26}$$

$$\text{Cc} = \frac{\text{Total number of conditions covered by discovered biclusters}}{\text{Total number of conditions present in the matrix}} \tag{2.3.27}$$

$$\text{Mc} = \frac{\text{Total number of cells covered by discovered biclusters}}{\text{Total number of cells present in the matrix}} \tag{2.3.28}$$

Further, bicluster evaluation methods can be classified into two main classes: intra- and inter-biclusters evaluation methods. Intra-bicluster methods quantify the quality of bicluster rather we can say the coherence degree of biclusters. This has been elaborated in the previous 'compactness' point. On the other hand, inter-biclusters evaluation function measures the quality of a group of biclusters. Mainly, it is used in synthetic datasets, where we know the true hidden biclusters in the data matrix. Several inter-biclusters measures are available in the literature.

The validation of identified biclusters is performed using Jaccard Coefficient (JC) on the basis of known biclusters [278]. Let us consider two biclusters, say $\beta^1_{\mathcal{I}_1 \times \mathcal{J}_1}$ and $\beta^2_{\mathcal{I}_2 \times \mathcal{J}_2}$, the JC is measured as in Equation (2.3.29).

$$JC(\beta^1, \beta^2) = \frac{|(\mathcal{I}_1 \cup \mathcal{J}_1) \cap (\mathcal{I}_2 \cup \mathcal{J}_2)|}{(|\mathcal{I}_1 \cup \mathcal{J}_1) \cup (\mathcal{I}_2 \cup \mathcal{J}_2)|} \tag{2.3.29}$$

Let us again consider two sets of biclusters, discovered $D = \{D_1, D_2, ..., D_l\}$ and original $O = \{O_1, O_2, ..., O_k\}$. The match score ($MS$) between the two sets of biclusters is defined in Equation (2.3.30). How well the implanted biclusters are recovered is given by the recovery score $MS(O, D)$. On the other hand, the $MS(D, O)$ score reflects what extent the discovered biclusters match the true biclusters known as the relevance score. Relevance and recovery scores range from 0 to 1. A relevance score of 1 signifies that all the identified biclusters are expected whereas a recovery score of 1 denotes that all the implanted biclusters

have been found.

$$MS(O,D) = \frac{1}{|O|} \sum_{(\mathcal{I}_1,\mathcal{J}_1)\in O} max_{(\mathcal{I}_2,\mathcal{J}_2)\in D} JC(\beta^1,\beta^2) \tag{2.3.30}$$

(II) **Biological validation:** The biological assessment of identified biclusters from real datasets is carried out using functional enrichment analysis as mentioned before. The key goal is to determine whether the genes of each generated bicluster are significantly enriched or not with respect to the GO annotations. Due to the unavailability of the true biclusters, we use GO enrichment analysis to evaluate the biclusters, demonstrating how well genes can match with different GO categories. A bicluster is considered to be enriched if p-values of all the annotation terms are less than the significance cut-off value. Moreover, if one of the annotation terms is from any one of the GO categories such as BP, MF, and CC, it is said to be enriched with BP, MF, or CC.

2.4 Triclustering

Triclustering algorithm overcomes the limitations of both full-space clustering and biclustering. In triclustering the third dimension i.e., time point is added to the dataset besides genes and samples. In this section, we present formal definitions of tricluster and its different types. Subsequently, we provide a comprehensive survey of different triclustering algorithmic strategies relevant to biological data and quality measures for triclusters.

Definition 2.4.1 *Let $\mathcal{D}_{G\times S\times T}$ be the three dimensional gene expression or GST data of size $m\times n\times v$ which consists of m number of genes, $G = \{g_1, g_2, \ldots g_m\}$, n number of samples or experimental conditions $S = \{s_1, s_2, \ldots, s_n\}$, and v number of time points $T = \{t_1, t_2, \ldots, t_v\}$. Each cell value ($d_{xyz}$) of the matrix represents the expression level of x^{th} gene, y^{th} experimental condition at z^{th} time point. The 3D matrix can also be referred as $\mathcal{D} = \{G, S, T\}$.*

In this sequel, we represent the expression of gene g_i in time t_z across all samples denoted by a row vector $F_i(St_z)$ [316]. Thus, we can view 3D gene expression data as 2D gene expression data $G \times (S \times T)$ in horizontal plane shown in Equation

2.4.1 and depicted in Figure 2.3.

$$\mathcal{D} = \begin{bmatrix} F_1(St) \\ F_2(St) \\ F_3(St) \\ \dots \\ F_m(St) \end{bmatrix} \tag{2.4.1}$$

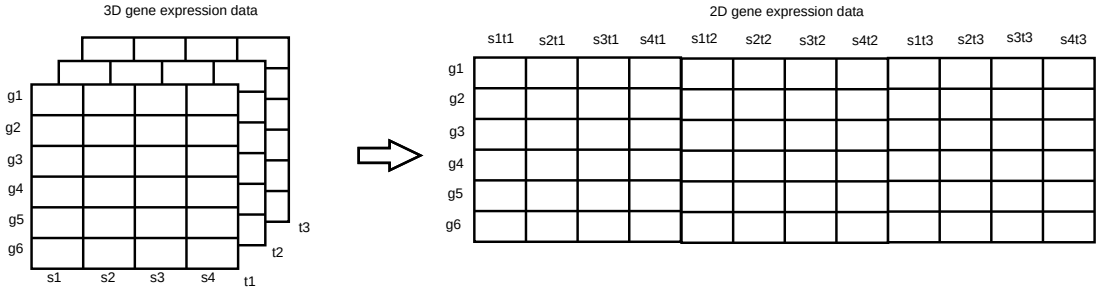


Figure 2.3: $\mathcal{D}$ gene expression data can be viewed as 2D gene expression data.

Definition 2.4.2 *A tricluster $\mathcal{T}(X, Y, Z) = \{d_{xyz}\}$ is defined as a submatrix $\mathcal{T}$, where $x \in X$, $y \in Y$, and $z \in Z$. The submatrix $\mathcal{T}$ represents subset of genes $X \subseteq G$ that are co-expressed under subset of experimental conditions or samples $Y \subseteq S$ over a subset of time-points $Z \subseteq T$.*

Triclustering algorithm aims to discover a set of triclusters $\{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_K\}$, such that each tricluster $\mathcal{T}_i$ satisfies some sort of homogeneity criteria and statistical significance. Considering the definition 2.4.2, it can still be maintained the restrictions on the locality of subspace [136]. In terms of this, two types of clustering definitions are defined: full cluster and partial cluster.

Definition 2.4.3 *A full cluster is a subspace consisting of a subset of objects from any one dimension and all the objects from the remaining dimensions.*

Full clusters are defined as $\mathcal{F}_\mathcal{C} = (X, S, T)$, $\mathcal{F}_\mathcal{C} = (G, Y, T)$, or $\mathcal{F}_\mathcal{C} = (G, S, Z)$, where $X \subseteq G, Y \subseteq S$, and $Z \subseteq T$. For example, in the first case, $\mathcal{F}_\mathcal{C} = (X, S, T)$ the clusters are a subset of genes across all conditions and time points and so on.

Definition 2.4.4 *A partial cluster is a subspace defined by subsets of objects from any two dimensions and all objects from the remaining dimension.*

Partial clusters are defined as $\mathcal{P}_\mathcal{C} = (X, Y, T)$, $\mathcal{P}_\mathcal{C} = (X, S, Z)$, or $\mathcal{P}_\mathcal{C} = (G, Y, Z)$, where $X \subseteq G, Y \subseteq S$, and $Z \subseteq T$. Both the types of clusters with varying locality are illustrated in Figure 2.4.

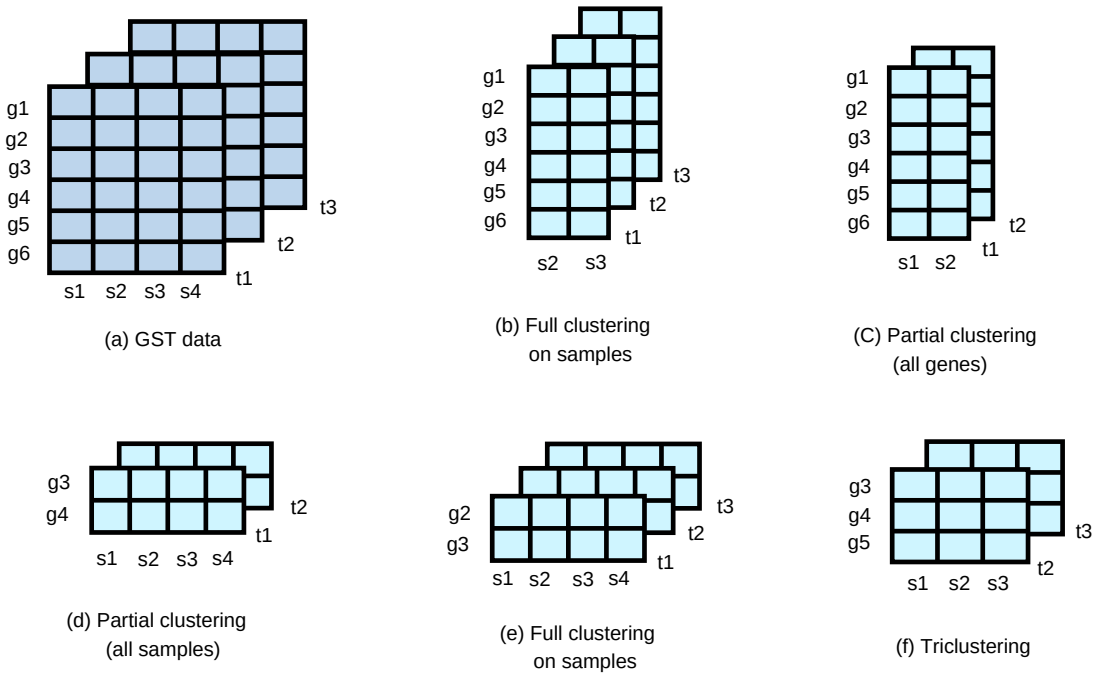


Figure 2.4: Different subspace clustering of 3D data with varying locality criteria.

2.4.1 Tricluster types

Like bicluster, tricluster has several types such as additive, multiplicative, and additive-multiplicative triclusters which are explained next.

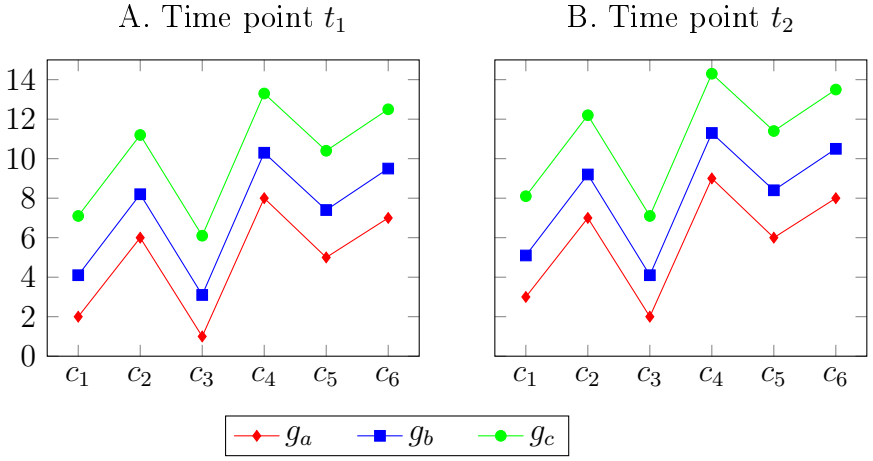


Figure 2.5: Additive patterns for different time points.

Definition 2.4.5 *A tricluster* $\mathcal{T}(X, Y, Z) = \{d_{xyz}\}$, *where* $x \in X$, $y \in Y$, *and* $z \in Z$ *is said to be additive tricluster if each element of the tricluster satisfies the following equation 2.4.2, where* α_x, β_y, *and* γ_z *are the additive factors of* x^{th} *gene,* y^{th} *experimental conditions, and* z^{th} *time point, respectively.*

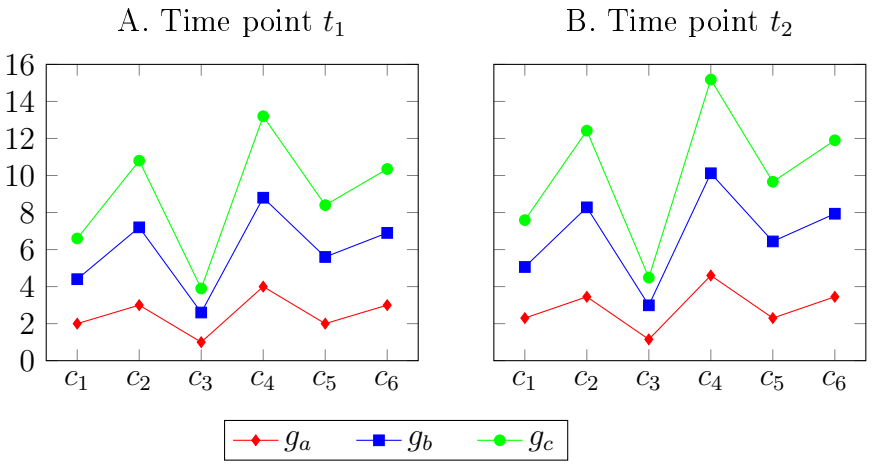


Figure 2.6: Multiplicative patterns for different time points.

$$d_{xyz} = const + \alpha_x + \beta_y + \gamma_z \tag{2.4.2}$$

Figure 2.5 illustrates the additive patterns of three genes where g_a is considered to be the base pattern and others are shifted with additive factors according to Equation 2.4.2. The additive factors for g_b, g_c, c_1, c_2, c_3, c_4, c_5, c_6, t_1, and t_2 are 2, 3, 0.1, 0.2, 0.1, 0.3, 0.4, 0.5, 0, and 1, respectively.

Definition 2.4.6 *A tricluster $\mathcal{T}(X, Y, Z) = \{d_{xyz}\}$, where $x \in X$, $y \in Y$, and $z \in Z$ is said to be multiplicative tricluster if each element of the tricluster satisfies the equation 2.4.3, where τ_x, η_y, and ζ_z are the multiplicative factors of x^{th} gene, y^{th} experimental condition, and z^{th} time point, respectively.*

$$d_{xyz} = const \times \tau_x \times \eta_y \times \zeta_z \tag{2.4.3}$$

The multiplicative patterns of three genes are shown in Figure 2.6 where g_a is considered to be the base pattern and others are scaled with multiplicative factors according to Equation 2.4.3. The multiplicative factors for g_b, g_c, c_1, c_2, c_3, c_4, c_5, c_6, t_1, and t_2 are 2, 3, 1.1, 1.2, 1.3, 1.1, 1.4, 1.15, 1, and 1.15, respectively.

Definition 2.4.7 *A tricluster $\mathcal{T}(X, Y, Z) = \{d_{xyz}\}$, where $x \in X$, $y \in Y$, and $z \in Z$ is said to be additive-multiplicative tricluster if each element of the tricluster satisfies the following equation 2.4.4.*

$$d_{xyz} = const \times \tau_x \times \eta_y \times \zeta_z + \alpha_x + \beta_y + \gamma_z \tag{2.4.4}$$

Equation 2.4.4 is the generalized form of both Equation 2.4.2 and Equation 2.4.3 where the multiplicative and additive factors are 1 and 0, respectively. From the Figure 2.7, we can observe the additive-multiplicative patterns of three genes. We include additive and multiplicative factors to the base gene g_a to get the

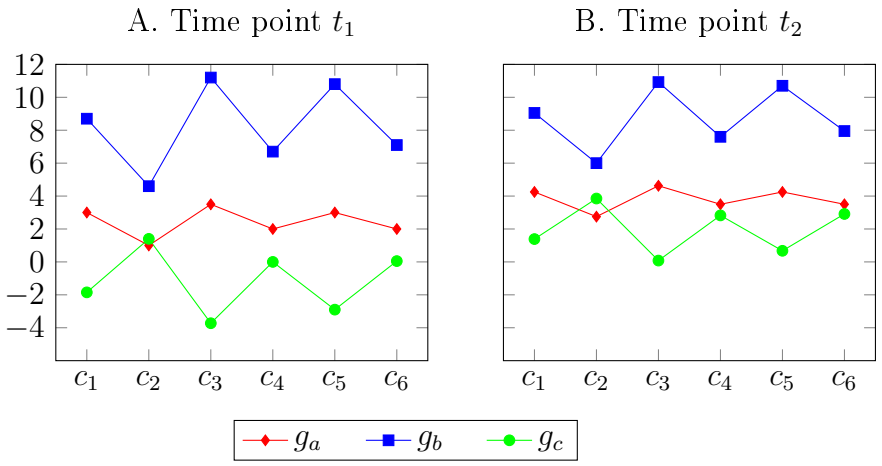


Figure 2.7: Additive-multiplicative patterns for different time points.

patterns of g_b and g_c. The additive factors for g_b, g_c, c_1, c_2, c_3, c_4, c_5, c_6, t_1, and t_2 are 2, 3, 0.1, 0.2, 0.1, 0.3, 0.4, 0.5, 0, and 2, respectively. The multiplicative factors for g_b, g_c, c_1, c_2, c_3, c_4, c_5, c_6, t_1, and t_2 are 2, -1.5, 1.1, 1.2, 1.3, 1.1, 1.4, 1.15, 1, and 0.75, respectively.

2.4.2 Triclustering algorithms

An enormous number of triclustering algorithms have been developed in the last decade [136, 228]. According to Henriques and Madeira [136], existing triclustering algorithms can be divided into six types, (I) Greedy, (II) Stochastic, (III) Exhaustive, (IV) Biclustering-based, (V) Pattern-based, and (VI) Evolutionary-based approaches. The applications of these algorithms are not just restricted to biological data. Next, we discuss them in detail.

(I) **Greedy methods:** Based on the 3D coherence function, the greedy approach adds and removes objects in an iterative manner from candidate subspace in order to maximize or minimize the 3D objective function. Bhar et al. [37] have proposed a greedy algorithm, δ-TRIMAX to identify triclusters with lower (less than δ threshold) Mean Square Residue (MSR) value in 3D data. For a perfect shifting tricluster MSR value is 0 [38]. Therefore, a lower MSR score signifies a better quality tricluster.

Definition 2.4.8 *The Mean Square Residue MSR_{3D} of a tricluster $\mathcal{T}(X, Y, Z)$ can be modeled using Equation 2.4.5 where d_{xYZ} is the mean of x^{th} gene ($d_{xYZ} = \frac{1}{|Y||Z|}\sum_{y\in Y, z\in Z} d_{xyz}$), d_{XyZ} is the mean of y^{th} sample ($d_{XyZ} = \frac{1}{|X||Z|}\sum_{x\in Y, z\in Z} d_{xyz}$), d_{XYz} is the mean of z^{th} time point*

($d_{XYz} = \frac{1}{|X||Y|}\sum_{x\in Y, y\in Y} d_{xyz}$), and d_{XYZ} is the mean tricluster ($d_{XYZ} = \frac{1}{|X|*|Y|*|Z|}\sum_{x\in Y, y\in Y, z\in Z} d_{xyz}$).

$$MSR_{3D} = \frac{1}{|X| * |Y| * |Z|} \sum_{x\in X, y\in Y, z\in Z} (d_{xyz} - d_{xYZ} - d_{XyZ} - d_{XYz} + 2d_{XYZ})^2 \tag{2.4.5}$$

The δ-TRIMAX performs in two major steps: i) starting from the 3D GST input, the algorithm iteratively removes genes, samples, and time points until the MSR_{3D} becomes less than predefined threshold δ and (ii) gene, samples, and time points are added to the tricluster if it satisfies MSR value to be less than δ. Many authors have adopted this principle in their work such as Three-Way Clustering (TriWClustering) [79] and TriClust tool [80] to identify triclusters with lower MSR values.

Xu et al. [345] have proposed a novel tricluster model LagMiner to find S^2D^3 cluster where S^2 reflects additive-multiplicative patterns and D^3 is three dimensional data. The goal of LagMiner is to find triclusters satisfying: (i) shifting-and-scaling interplane coherence using S^2Score for each gene-sample plane and gene-time plane, $\forall_{z_k\in Z} S^2Score(X, Y, z_k) \leq \delta$ and $\forall_{x_i\in X} S^2Score(K, J, x_i) \leq \delta$ (δ is a coherence strength) and (ii) an order-preserving interplane for each gene-time plane. Let, sample triplet order is $\rho(y_j) = y_{j1} \prec y_{j2} \prec y_{j3}$ and coherence measure S^2Score can be defined using Equation 2.4.6.

$$S^2Score(X, Y, z_k) = \max_{x_i\in X \rho(y_j)\subseteq Y} \frac{d_{xy_2z} - d_{xy_1z}}{d_{xy_3z} - d_{xy_1z}} - \min_{x_i\in X \rho(y_j)} \frac{d_{xy_2z} - d_{xy_1z}}{d_{xy_3z} - d_{xy_1z}} \tag{2.4.6}$$

The algorithm initiates with a subspace having sample triplets and as many genes as possible. Next, samples and times are included in the subspace if it satisfies the above-mentioned coherence.

(II) **Stochastic approaches:** Some proposed triclustering algorithms rely on stochastic approaches. TWIGS (Three-Way module Inference via Gibbs Sampling) is proposed by Amar et al. [10] based on both hierarchical Bayesian data model and Gibbs sampling to find large triclusters from 3D time course data. The method uses Bernoulli-β and the Normal-γ assumption for binary 3D data and real data, respectively in the case of hierarchical Bayesian data model. Initially, the algorithm uses biclustering solutions and iteratively improves the solution using the Gibbs sampling procedure.

Triclustering 3D plaid framework referred to as 3D-Plaid is proposed by Mankad and Michailidis [230]. This 3D-plaid is an extended version of the model

proposed in [188] for 2D. In the 3D-plaid model, data can be represented as a sum of q triclusters, as presented in Equations 2.4.7 and 2.4.8.

$$d_{xyz} = \mu_0 + \sum_{i=0}^{q} \theta_{xyzi} \rho_{xi} \kappa_{yi} \tau_{zi} \tag{2.4.7}$$

$$\theta_{xyzi} = \mu_i + \alpha_{xi} + \beta_{yi} + \gamma_{zi} + \eta_{xyzi} \tag{2.4.8}$$

θ_{xyzi} is the contributions of each tricluster. In a tricluster $\mathcal{T}_i$, boolean variables ρ_{xi}, κ_{yi}, and τ_{zi} specifies the membership of genes, samples, and time points. Tricluster can be obtained by minimizing the merit function as given in Equation 2.4.9, assuming η_{xyz} approximately Gaussian.

$$\sum_{x=1}^{m} \sum_{y=1}^{n} \sum_{x=1}^{l} (d_{xyz} - \theta_{xyz0} - \sum_{i=0}^{K} \theta_{xyzi} \rho_{xi} \kappa_{yi} \tau_{zi})^2 \tag{2.4.9}$$

The key concept of this framework is to find subspaces exhibiting strong deviations and then estimate their dependence over the whole data array. Different strategies like pruning, backfitting, and other heuristic are also applied in this model.

Guigourès et al. [122] introduce a novel technique that formulates the triclustering problem into a clustering tripartite graph. The algorithm is built upon MODL as mentioned in [46]. The work presents maximum a posteriori (MAP) for estimating the parameters i.e., the number of clusters is chosen automatically for co-clustering of 3D temporal data.

Recently, Wu et al. [343] have proposed BCAT_I (Bregman cuboid average triclustering algorithm with I-divergence) to analyze hidden patterns from georeferenced time series (GTS). BCAT_I groups regular triclusters in such a way that it minimizes the loss of mutual information. It then subsequently refines triclusters using K-means to capture spatiotemporal patterns in the data.

(III) **Exhaustive approaches:** This section surveys those methods which are based on exhaustive approaches. Jiang et al. [161] have developed an algorithm to mine coherent gene clusters from temporal 3D data. Each cluster consists of a subset of genes over a subset of samples along with the time points. PCC is used to extract maximal coherent sample sets for each gene or maximal coherent gene sets for sample sets. Two mining methods have been proposed: Sample-Gene Search and Gene-Sample Search. To get all possible combinations of genes or samples, the algorithm uses an enumeration tree and utilizes a pruning operation.

Hereafter, for each subset of samples or genes, the algorithm finds a subset of genes or samples relying on an efficient solution of the inverted list of maximal coherent sample sets of all genes.

In the year 1994, the triclustering of 3D binary data known as a triadic formal concept has been proposed by Krolak-Schwerdt et al. [181]. Lehmann and Wille [193] have introduced the theoretical concepts of Triadic Formal Concept Analysis. Many machine learning methods are reformulated algebraically by formal concept analysis. Ignatov et al. [148] present formal definitions of "optimal patterns" in triadic data and have experimentally shown comparative results for five triclustering algorithms. They are object-attribute-condition triclustering (OAC) Box [150], TRIBOX [237], SPECTRIC [151], TRIAS (Triadic Formal Concept Analysis) [155], and OAC-Prime. Initially, the OAC algorithm is proposed by Ignatov and Kuznetsov [149]. Extensive work has been done based on this particular algorithm, such as TRIAS, TRIBOX, SPECTRIC, OAC-Prime, and greedy-OAC [117] to improve the quality of triclusters.

(IV) **Biclustering-based approaches:** For each plane of 3D data, a set of biclusters are identified first and then a set of triclusters are discovered. The biclustering-based approach uses intraplane coherence for inferring triclusters from the identified biclusters and can be assigned for more than one dimension. One point can be noted that these approaches are dependent on the algorithmic strategy like greedy, stochastic, or others applied on data to get biclusters.

In 2005, pioneering work on triclustering algorithm named TriCluster has been proposed by Zhao and Zaki based on graph-based approach [364]. TriCluster algorithm performs in four basic steps. i) The GST data is sliced according to time points and for each time point matrix ($G \times S$), it constructs a multigraph to store a valid ratio for all pairs of samples. (ii) The maximal clique is searched from these multigraphs to find a set of biclusters by performing Depth First Search (DFS). (iii) From the mined biclusters, a graph is constructed to get maximal triclusters. (iv) To the end, deletion and merging of clusters are performed if only overlapping criteria is satisfied. The algorithm ignores intertemporal coherence (which means genes or samples do not coherently vary over time points and may appear in a slice of tricluster) and is highly dependent on parameters. Here, Pearson correlation is used as intraplane coherence.

Following the approach of TriCluster, several versions were designed to mine clusters. Jiang et al. [165] have proposed a more generalized 3D model (gTRICLUSTER) to overcome the problem of intertemporal coherence and pa-

rameter dependency of TriCluster. The algorithm uses Spearman for interplane coherence instead of Pearson correlation across time points to capture more flexible clusters. It focuses to identify biologically meaningful coherent clusters from noisy data. Despite having advantages of gTRICLUSTER, it suffers from inter-gene coherency and is biased towards sample and gene size. It can not detect two similar patterns which may belong to the different time ranges. Shortly after, Araújo et al. [16] have designed a parallel version of TriCluster algorithm (ParTriCluster) using filer-labeled-stream which is supported by Anthill parallel programming environment. ParTriCluster reduces the computational complexity and handles scalability issues nicely.

Ahmed et al. [4] have proposed a technique Intersected Coexpressed Subcube Miner (ICSM) to find order-preserving submatrix taking into account the inter-gene and inter-temporal coherence. It also finds time-latent triclusters. In association with triclustering algorithm, the authors propose a planar similarity measure (PMRS) to detect shifting correlations between two planes. The PMRS between two planes x and y with the size of each matrix of $m \times n$, can be calculated using Equation 2.4.10, where, μ_a and μ_b are the mean of a and b, respectively.

$$PMRS(a,b) = \frac{\sum_{i=1}^{a}\sum_{j=1}^{b} \text{abs}(a(i,j) - \mu_a - b(i,j) + \mu_b)}{2 \times \max(\sum_{i=1}^{a}\sum_{j=1}^{b} \text{abs}(a(i,j) - \mu_a), \sum_{i=1}^{a}\sum_{j=1}^{b} \text{abs}(b(i,j) - \mu_b))} \tag{2.4.10}$$

Tchagang et al. [316] have developed an Order Preserving Triclustering (OPTricluster) for short time series 3D data. In sample direction, OPTricluster uses a combinatorial approach and an order-preserving approach in the time direction to get triclusters with coherent evolution. The dimensions of time and sample can be swapped according to the goal [136]. In such a scenario, OPTricluster is capable of capturing groups of genes under a subset of conditions for specific time points. The algorithm consists of five major steps: (i) quantization of data, (ii) ranking of gene expression across time points, (iii) identifying order-preserving biclusters, (iv) inferencing triclusters from biclusters, and (v) statistical significant assessment. OPTicluster efficiently mines the triclusters for small time series data and identifies the cluster having constant or coherent patterns.

Kakati et al. [171] have presented a distributed triclustering algorithm Shifting-and-Scaling Similarity Triclustering (SSSimTri) in order to identify shift and/or scale patterns from GST data. The algorithm is seed-growth and ex-

tracts biclusters from each time slice in a parallel fashion. The work uses a fast biclustering algorithm and a shared-nothing client-server architecture to analyze 3D data.

(V) **Pattern-based approaches:** In this approach, triclusters are discovered with well-defined patterns consisting of a subset of any objects. Ji et al. [160] have proposed the algorithm frequent closed cube (FCC) which elaborates the concept of 2D frequent closed pattern in the context of 3D. The authors of this work have mentioned two novel algorithms namely Representative Slice Mining (RSM) and CubeMiner. The former mines transformed 2D dataset from 3D exploiting the existing 2D FCP mining algorithm and prune cubes that are not closed. The latter method i.e., CubeMiner directly operates on 3D data to mine FCC.

More recently, a pattern-based algorithm TimesVector is proposed by Jung et al. [169] to seek groups of genes which exhibit similar and differentially expressed patterns from GST data. The main idea of TimesVector can be understood in three basic steps. (i) Data is concatenated along condition and time dimensions so that clustering data can be reduced into two-dimensional space, (ii) Spherical K-means algorithm can be applied to the data. The resulting clusters are then classified to detect similar and distinct patterns and (iii) some genes remained unclassified.

(VI) **Multiobjective optimization approaches:** In this context, multiple objective functions such as volume and homogeneity of triclusters are optimized simultaneously. Liu et al. [212] have proposed a novel multiobjective evolutionary 3D clustering algorithm termed MOGA3C to find one or more significant clusters with the maximum size. This algorithm is considered to be one first proposed triclustering algorithms in this category. It simultaneously satisfies three objective functions by maximizing tricluster size, minimizing MSR value, and maximizing gene-dimension variance. The x^{th} gene-dimension variance in a tricluster $\mathcal{T}(X, Y, Z)$ and overall gene-dimension variance can be defined by Equations 2.4.11 and 2.4.12, respectively.

$$GVAR(x, Y, Z) = \frac{1}{|Y||Z|} \sum_{y \in S, z \in Z} (d_{xyz} - d_{xSZ})^2 \tag{2.4.11}$$

$$GVAR(X, Y, Z) = \frac{1}{|X||Y||Z|} \sum_{x \in X, y \in Y, z \in Z} (d_{xyz} - d_{xYZ})^2 \tag{2.4.12}$$

Gutiérrez-Avilés et al. [126] have proposed an evolutionary heuristic,

genetic algorithm TriGen (Triclustering-Genetic based). The TriGen discovers triclusters minimizing both 3D MSR value and correlation measure, least square approximation (LSL). After that, Gutiérrez-Avilés and Rubio-Escudero have extended the work of TriGen by embedding a new evaluation measure MSL (Multi Slope Measure) [125] to judge the quality of tricluster. MSL measures the similarity among angles of the slopes for all pairs of genes, conditions, and times of the given subspace.

A modified version of δ-TRIMAX, termed as Evolutionary Multi-objective Optimization Algorithm for δ-TRIMAX (EMOA-δ-TRIMAX) has been developed by Bhar et al. [38] using Non-dominated Sorting Genetic Algorithm (NSGA-II). The problems associated with Trigen [126] and EMOA-δ-TRIMAX are fine-tuning of input parameters and it is not as fast as a greedy algorithm.

2.4.3 Tricluster evaluation methods

Like full-space clustering and biclustering, triclustering also requires the assessment of discovered triclusters. However, evaluation of triclustering algorithm is a very difficult job in two perspectives as mentioned in [136]. Synthetic data generation procedure for tricluster is biased towards the particular homogeneity and absence of consensual similarity metric. Moreover, for real datasets no ground truth is present. Accordingly, triclustering solutions can be evaluated in two ways: statistical significance and biological significance.

(I) **Statistical significance:** The triclustering solutions can be assessed in several ways.

(a) **Homogeneity:** Merit functions are not only used to guide triclustering process, it is also used to evaluate triclustering results. Diverse merit functions are listed in [136]. In general, evaluating clustering results with merit functions is biased towards specific homogeneity criteria. Therefore, for fair comparison, it is advisable to combine results of more than one merit function. Most commonly used merit function is MSR value as defined in Equation 2.4.5. Merit function as shown in Equation 2.4.12, is used to identify constant tricluster. This merit function calculates the variance of a tricluster which finds a larger tricluster with some allowable level of threshold.

Ahmed et al. [4] have proposed two internal evaluation measures, intra-temporal and inter-temporal homogeneity as defined mathematically in Equation 2.4.13 and 2.4.14, respectively, where $mr(z)$ is the mean of genes in z time plane with samples present in Y, $corr(x, mr(z))$ is the Pearson correlation between

gene x and $mr(z)$, and mmr is the mean of the means of all time planes belongs to the tricluster.

$$\lambda_g = \frac{\sum_{z \in Z} \frac{\sum_{x \in X}(corr(x, mr(z)))}{|X|}}{|Z|} \tag{2.4.13}$$

$$\lambda_z = \frac{\sum_{z \in Z} corr(mr(z), mmr)}{|Z|} \tag{2.4.14}$$

Another merit function is mentioned in Equation 2.4.9 based on the quadratic error, which is to be minimized in order to get triclusters. Others are PRMS [4], LSL [126], and MSL [125].

(b) **Accuracy based:** Accuracy based metrics are formulated to evaluate the synthetic datasets based on true solutions as triclusters are implanted in the background matrix. Researchers have derived several similarity metrics to evaluate triclusters. The most popular measure is Jaccard based scores known as match score [278]. Let, $\mathcal{H} = \{H_1, H_2, \ldots, H_r\}$ are the hidden or planted triclusters and $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$ are the set of discovered triclusters. Henriques et al. [136] have revised match score as shown in Equation 2.4.15 to add a penalty of non-matched volume triclusters.

$$RMS3(\mathcal{F}, \mathcal{H}) = \frac{1}{|\mathcal{F}|} \sum_{\mathcal{T}_1 \in \mathcal{F}, \text{maxarg}\{Jac(\mathcal{T}_1, \mathcal{T}_2) | \mathcal{T}_2 \in \mathcal{H}\}} \sqrt[3]{\frac{|X_1 \cap X_2|}{|X_1 \cup X_2|} \frac{|Y_1 \cap Y_2|}{|Y_1 \cup Y_2|} \frac{|Z_1 \cap Z_2|}{|Z_1 \cup Z_2|}} \tag{2.4.15}$$

Bhar et al. [36] have proposed similarity score $MS_X(\mathcal{T}_1, \mathcal{T}_2) \times MS_Y(\mathcal{T}_1, \mathcal{T}_2) \times MS_Z(\mathcal{T}_1, \mathcal{T}_2)$, which computes the product of match score in different dimensions. Another Jaccard based metric is proposed by Amar et al. [10] in Equation 2.4.16 which considers weighted all pairwise match score.

$$\frac{1}{|\mathcal{F}| + |\mathcal{H}|}\left(\sum_{\mathcal{T}_1 \in \mathcal{F}} \max_{\mathcal{T}_2 \in \mathcal{H}} MS(\mathcal{T}_1, \mathcal{T}_2) + \sum_{\mathcal{T}_2 \in \mathcal{H}} max_{\mathcal{T}_1 \in \mathcal{F}} MS(\mathcal{T}_2, \mathcal{T}_1)\right) \tag{2.4.16}$$

(c) **Coverage:** Gene/Sample/Time coverage can be quantified by unique genes/samples/times discovered in all the triclusters to the total genes/samples/times present in the dataset. On the other hand, Coverage is defined by Equation 2.4.17, where X^A, Y^A, and Z^A are discovered unique genes, samples, and times by an algorithm A and G, S, and T represent the total number of genes, samples, and times present in the dataset.

$$Coverage = \frac{X^A \times Y^A \times^A}{G \times S \times T} \times 100 \tag{2.4.17}$$

(II) **Biological significance:** Triclustering algorithms are assessed based on real biological datasets. But, one point we have to remember is that there is a dearth of biological information of GST data which may be considered as ground truth. In this context, we evaluate triclustering solutions against background knowledge such as GO, as we have done for clustering and biclustering. Enrichment analysis is one such method that helps to establish the significance of found clusters. We exploit the information provided by GO and KEGG pathways to assess the significance of our results biologically.

2.5 Discussion

In this chapter, we have done a widespread survey of all three types of clustering algorithms with validation which aims to examine the different algorithms or different evaluation measures for gene expression data. Despite a large number of algorithms, clustering remains an extremely challenging task. There is no basic guideline for choosing an appropriate clustering algorithm and biologically validating the clustering results. Therefore, it is hard to define the universal consensus on the definition of cluster and the most effective one. Moreover, no algorithm exists which is considered to be the best performer throughout all clustering problems. On the other hand, clustering algorithm largely depends on the input parameters and the properties of data. Hence, scientists are actively participating in either developing new clustering algorithms or designing new validation techniques, as a consequence to evaluate discovered clustering solutions biologically.

Although clustering algorithms have been successfully applied to biological data, there still exists an intrinsic problem in clustering because of its unsupervised learning nature. Full-space, biclustering, and triclustering algorithms that we have mentioned earlier ignore known gene functions during clustering. Most clustering algorithms cluster objects without exploiting any biological constraints as an input to the clustering. Basically, in most biological problems, domain-specific hypotheses or knowledge are present. Therefore, it is hard for biologists or researchers to manually examine the information to interpret the derived result and come to a conclusion. Using only classical methods it may not be possible to figure out all potential relationships among genes. On realizing this fact, researchers have started giving importance to incorporating gene functional or domain knowledge for developing an algorithm that will automatically reveal biologically more reliable clusters. With this as our guiding principle, we

have focused on steering our work from unsupervised to semi-supervised learning, where we embed prior biological knowledge to guide clustering algorithms.

In the field of data mining or machine learning, semi-supervised learning algorithms are gaining a lot of popularity. A semi-supervised clustering approach can be achieved in two different ways, either by modifying similarity measures or by directly enhancing clustering algorithms. Handful numbers of semi-supervised algorithms are proposed to cover the full-space clustering area and even in the case of biclustering algorithms, the number is much less. From the literature, a similar scenario can be observed for the triclustering algorithm i.e., scarcity of semi-supervised triclustering algorithm. In succeeding chapters, we have studied semi-supervised full-space, biclustering, and triclustering algorithms in depth.

3

Full-space Cluster Analysis of Cancer Gene Expression Data

Cluster analysis helps researchers to formulate a new hypothesis to detect the relationship between genes and is effectively used to predict the function of unknown genes based on the genes of known functions with which it is co-expressed [242]. In other words, it is based on the assumption that similar expression patterns may exhibit a strong correlation with their functions in the biological activities [214]. A number of full-space clustering algorithms have been introduced in Chapter 2. The remainder of the chapter is organized as follows. We start by briefing the introduction in Section 3.1. Section 3.2 reviews related work in this area. Next, we provide a clear motivation for the work in Section 3.3. Two major types of contributions of this particular domain are provided. In the first part of Section 3.4, the proposed unsupervised full-space clustering is presented whereas the second part elaborates two semi-supervised clustering algorithms incorporating GO as external knowledge which surpass the drawbacks of an unsupervised algorithm. Section 3.5 analyzes the time complexity of all three proposed methods. The proposed unsupervised method is performed with both synthetic and real cancer gene expression datasets while semi-supervised methods are successfully employed in real datasets only, which are given in Sec-

tion 3.6. Thereafter, the validation of clustering results is discussed with the help of internal and external measures as mentioned in Chapter 2 with existing algorithms. Further in Section 3.7, we propose a biomarker identification method utilizing clustering results as an application towards cancer datasets. Finally, in Section 3.8, we discuss all three proposed algorithms and their performances with respect to other existing methods.

3.1 Introduction

Recalling from Chapter 1, a traditional full-space clustering algorithm is typically unsupervised. In a true sense, no labeling is provided to gain insights into the underlying structure of input data. It is basically an optimization problem. The fundamental goal of clustering is maximizing the intra-cluster distance and minimizing the inter-cluster distance. Cluster analysis has four key steps - preprocessing the gene expression data to be clustered, choosing an appropriate proximity measure, applying clustering technique to data, and the final step is to perform cluster validation. A vast majority of literature in this area has been covered, that formally aim to identify co-expressed genes from expression data. To date, clustering leads to be an active and rich area of research [100, 357].

In practice, K-means [226], SOM [313], and HC [95] are widely applied in the context of gene expression data clustering. However, these approaches attempt to group all input genes into some sort of a finite number of clusters. Thus, genes that are not co-expressed are also assigned to their "best-fitting" cluster and as a result, co-expressed and non-co-expressed genes come under the same cluster [1]. This outcome violates the basic property of biological clusters that no two clusters should have identical expression profiles rather it should form a cluster only with co-expressed genes. To address this issue, we propose an unsupervised algorithm, named **G**raph **A**ttraction **Clust**ering (GAClust) algorithm. In fact, very little effort has been made for partial clustering that avoids force clustering of the complete set of input data [1, 293, 317]. The aim of clustering should be the extraction of data and not a data partitioning problem that is necessarily co-expressed under conditions.

Unsupervised clustering algorithms are built on the presumption that co-expressed genes are likely to have common biological functions. However, it is seen that most of the algorithms miss the gene functional prediction at the time of clustering. The algorithms mentioned in Section 2.2.2 use only gene expression data so far for proximity measure between two genes and ignore the gene

function in clustering. Using only proximity measure of gene expression data, it may be possible that two functionally unrelated genes can come under the same cluster due to their similar expression value. So, it is essential to find the GO-based similarity in association with gene expression profile similarity. If two GO terms share more common information it means that they are more similar to each other. To extract biologically meaningful clusters from the dataset it is necessary to incorporate biological knowledge at the time of clustering. This has motivated us to shift from unsupervised to semi-supervised clustering by incorporating Gene Ontology (GO) knowledge in the clustering process. GO is the fundamental database of bioinformatics, that specifically gives the annotations for gene products with consistent and structured vocabularies [192]. In this regard, we have modified our GAClust algorithm incorporating GO information to **S**emi-supervised **G**raph **A**ttraction **Clust**ering (SGAClust) algorithm.

Co-regulated genes can be of two types: positively co-regulated and negatively co-regulated genes [159].

Definition 3.1.1 *Two genes, g_a and g_b are said to be positively co-expressed if the expression values of g_a show an increasing (up-regulated) or decreasing (down-regulated) trend over all conditions for which g_b also shows the same trend. If g_b shows an opposite trend corresponding to g_a, then they are negatively co-expressed.*

Figure 3.1 (A) and (B) give an illustration of the positively and negatively co-expressed patterns. In a real scenario, researchers have explored the emerging need to discover co-regulated genes which include both positive and negative co-regulated genes. An important interpretation of co-regulated genes is co-expression, which simultaneously show the rise and fall of expression values [368]. If two genes reside far apart from each other, still there can be a strong correlation between them. Traditional distance-based methods cannot capture correlation for the analysis of co-expression. To address this issue, we have proposed another full-space clustering algorithm named **S**emi-supervised **D**ensity-based **C**lustering (SDC). SDC algorithm uses density information in association with pattern-based approach and incorporates GO knowledge from gene ontology consortium.

3.2 Related work

We broadly classify the clustering of gene expression data into two parts, viz, (i) unsupervised and (ii) semi-supervised full-space clustering algorithms. In Section 2.2.2, a vivid description of unsupervised full-space clustering algorithms

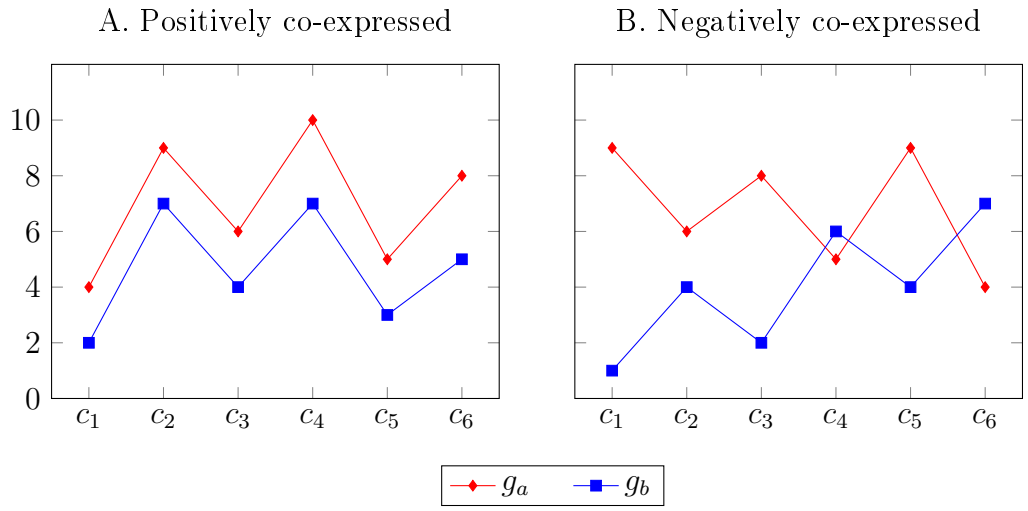


Figure 3.1: An illustration of different types of co-expression patterns. The x-axis denotes the conditions and the y-axis represents the expression values. A. Patterns g_a and g_b are positively co-expressed with respect to each other. B. Patterns g_a and g_b is negatively co-expressed with respect to each other.

has been provided. The review aims to examine different GO-based similarity measures and thereafter semi-supervised clustering algorithms.

Conventional clustering algorithms find sets of genes depending on their proximity ((dis)similarity) measure. In contrast, expression-based measures may not find the potential relationships among the genes as these measures are unable to capture the potential functional relationships among genes. Therefore, it is important to adopt ontologies for annotations while comparing entities. Semantic similarity (SS) allows the comparison of GO terms or GO annotated gene products by leveraging the hierarchical structure of the GO graph. SS calculates the closeness between them which in turn reflects numerical value. SS measure is the key technique to incorporate the knowledge of known genes from gene ontology and gene annotation files. A wide variety of SS measures can be found in [269, 270].

At first Lord et al. [218] have successfully applied SS in biology. Since then, several SS measures have been developed. We present a short survey of SS in the context of GO. To compare GO terms, there are two major approaches: edge-based and node-based. Edge-based approaches are dependent on the number of edges present in between GO terms. *Distance* (average of all paths or shortest path) and *common path* (the lowest common ancestor of two terms to root) are two popularly used techniques to calculate SS. On the other hand, node-based approaches rely on the comparison of the properties of the terms, their ascendants or their descendants. These semantic similarities are built on

the information theory which means how much information they commonly share. Information content of a term $\mathbb{T}$ is quantified as IC in a specific corpus and is described by negative log likelihood $IC = -\log(P(\mathbb{T}))$, where $P(\mathbb{T})$ represents the probability of occurrence of $\mathbb{T}$ in a specific corpus. Another way to determine IC is to calculate the number of children in GO which is not used commonly. To determine the SS between two terms that is how much information they share, IC can be applied to the common ancestors of both terms. To do this, two main approaches are used: the Most Informative Common Ancestor (MICA) and Disjoint Common Ancestor (DCA). MICA means common ancestor having highest IC [282] and DCA represents all common ancestors that do not subsume any other common ancestor [72]. Alternatively, node-based approaches also can be calculated by the number of shared annotations, number of gene-annotated products, number of shared ancestors, node depth, node-link density etc. While comparing gene products, often it can be done pairwise or groupwise. To quantify the pairwise similarity between two gene products, SS between their terms are combined. In this regard, often maximum, sum, and average are used for combining. Groupwise approaches are directly calculated by set, graph, or vectors which is different from the former one.

Here, we report some of the well known semantic similarities. Resnik similarity between two terms $\mathbb{T}_i$ and $\mathbb{T}_j$ is calculated by Equation 3.2.1, which is simply IC of their MICA. The lower bound of Resnik measure is 0 and it has no upper limit.

$$SS_{Res}(\mathbb{T}_i, \mathbb{T}_j) = IC(MICA) \tag{3.2.1}$$

Resnik measure does not consider the distance from both the terms to their lowest common ancestors. Hence, distance is taken into consideration in Lin's, and Jiang and Conrath's. Lin [207] similarity between two terms say $\mathbb{T}_i$ and $\mathbb{T}_j$ and given by Equation 3.2.2. SS_{Lin} gives the IC between two terms by considering the IC of each individual term and the IC of MICA. The obtained value of semantic similarity lies between 0 and 1.

$$SS_{Lin}(\mathbb{T}_i, \mathbb{T}_j) = \frac{2 \times IC(MICA)}{IC(\mathbb{T}_i) + IC(\mathbb{T}_j)} \tag{3.2.2}$$

Jiang and Conrath's [166] have proposed an IC-based measure as shown in Equation 3.2.3. The lowest and highest value of this measure is 0 and 1, respectively.

$$SS_{JCSS} = 1 - IC(\mathbb{T}_i) + IC(\mathbb{T}_j) - 2 \times IC(MICA) \tag{3.2.3}$$

These three node-based measures determine the similarity between two GO terms, which in turn can be extended for comparison of gene products, which have several GO terms. Wang et al. [332] have proposed a SS as a pairwise measure that is applied as edge-based. Let, a GO term $\mathbb{T}_i$ can be defined by a graph $G_{\mathbb{T}_i} = (\mathbb{T}_i, A_{\mathbb{T}_i}, E_{\mathbb{T}_i})$, where $A_{\mathbb{T}_i}$ is a set of GO terms in $G_{\mathbb{T}_i}$ including $\mathbb{T}_i$ and all ancestors of the term $\mathbb{T}_i$ and $E_{\mathbb{T}_i}$ is the set of edges or semantic relations. To do a quantitative comparison in between two GO terms, GO term is encoded by $\mathbb{T}_i$ as the aggregated contribution of all terms in $G_{\mathbb{T}_i}$. Therefore, S-value is used to define the contribution of GO terms $\mathbb{T}_i$. For any term $\mathbb{T}$ in $G_{\mathbb{T}_i}$, the S value of $\mathbb{T}_i$ is represented by Equation 3.2.4.

$$\begin{aligned} S_{\mathbb{T}_i}(\mathbb{T}_i) &= 1 \\ S_{\mathbb{T}_i}(\mathbb{T}) &= \max\{w_e \times S_{\mathbb{T}_i}(\mathbb{T}')|\mathbb{T}' \in \text{children of}(\mathbb{T}) \text{ if } \mathbb{T} \neq \mathbb{T}_i\} \end{aligned} \tag{3.2.4}$$

Here, w_e is the contribution factor of the edge between $\mathbb{T}_i$ and its children $\mathbb{T}'$ and $0 < w_e < 1$. After calculating the S-values for all the terms present in $G_{\mathbb{T}_i}$, semantic value $SV(\mathbb{T}_i)$ is obtained by Equation 3.2.5.

$$SV(\mathbb{T}_i) = \sum_{\mathbb{T} \in A_{\mathbb{T}_i}} S_{\mathbb{T}_i}(\mathbb{T}) \tag{3.2.5}$$

Considering the GO hierarchy as mentioned in Chapter 1, w_e for *is_ a* is 0.8 and *part_ of* is 0.6. Given two graphs say, $G_{\mathbb{T}_i}$ and $G_{\mathbb{T}_j}$ for two GO terms $\mathbb{T}_i$ and $\mathbb{T}_j$, semantic similarity between two terms can be represented by Equation 3.2.6.

$$S(\mathbb{T}_i, \mathbb{T}_j) = \frac{\sum_{\mathbb{T} \in A_{\mathbb{T}_i} \cap A_{\mathbb{T}_j}} (S_{\mathbb{T}_i}(\mathbb{T}) + S_{\mathbb{T}_j}(\mathbb{T}))}{SV(\mathbb{T}_i) + SV(\mathbb{T}_j)} \tag{3.2.6}$$

Nowadays, knowledge-based clustering algorithms have become an integral part of the research. However, the number of semi-supervised full-space clustering algorithms is much lesser than the number of unsupervised full-space clustering algorithms. Next, we present a brief survey on semi-supervised algorithms. Adryan and Schuh [2] have developed a GO-Cluster program that incorporates the hierarchy structure of the GO database as a model for cluster analysis and also gives the visualization of gene expression data at any level of the gene ontology tree. Huang and Pan [143] have included the gene function in distance metric and showed the advantage of using it over K-medoids (partitional) and hierarchical algorithms. In [264], a fast gene ontology-based clustering has been built which demonstrates hierarchical clustering and a heatmap visualization

with the help of gene expression data and GO annotations. It helps to identify rapidly the biologically related genes. Verbank et al. [326] have incorporated external biological knowledge (GO) to measure the distance between genes and applied it to the K-means algorithm, which gives biologically significant homogeneous co-expressed clusters. Speer et al. [306], Srivastava et al. [307], Macintyre et al. [225], Mitra and Ghosh [244] have incorporated the GO in clustering process for gene expression data. Hang et al. [128] have proposed an algorithm using two information such as gene density function and biological knowledge and the proposed one gave a better result than the standard algorithm.

Zhou et al. [370] also have developed an algorithm incorporating density of data and gene ontology in the distance-based clustering algorithm. Both the algorithms do not address the issue of identifying the positive and negative co-regulated genes. An algorithm that finds clusters comprised of co-regulated genes is being proposed by Ji and Tan [159]. To identify interesting partial negative, positive co-regulated gene clusters, Koch et al. [346] have proposed an algorithm that also discovers overlapping clusters.

3.3 Motivation

There are various unsupervised full-space clustering algorithms targeted to analyze gene expression data, as discussed in Chapter 2. Due to wet lab experiments, often gene expression data are noisy, therefore partial clustering is more suitable and appreciable in such cases. In partial clustering, full-space algorithm will not allow some of the genes (noise) to be present in a well-defined cluster that impacts the quality of the cluster. Additionally, a clustering algorithm should be designed in an automated framework such that, the algorithm either is free from parameters or is calculated dynamically. Here, we use the graph-theoretic approach to find potential solutions for discovering clusters from noisy data, which does not require the number of clusters explicitly. We propose the GAClust algorithm, which shares some features (clique finding) of an existing graph-theoretic approach, CAST [33]. The clustering result obtained by CAST highly depends upon the fine-tuning of the threshold value. Our proposed method dynamically estimates the parameters based on the dataset used. Moreover, the graph construction method is accomplished, by focusing only on groups of genes or common-neighborhood concept (grounded on proximity measure) rather than absolute measure between two genes.

Most of the researchers have the tendency to use Euclidean distance or

Pearson correlation in the traditional clustering process. Earlier, it is mentioned that external domain knowledge is the necessary pillar for guiding the clustering algorithms. The majority of the existing algorithms ignore external knowledge to get more biologically relevant clusters. Density-based methods can detect arbitrary shaped clusters but suffer from user input [15, 101, 283]. In this chapter, we propose two algorithms: SDC and SGAClust. SDC algorithm particularly focuses on the density-based algorithm and SGAClust is the modification of GAClust which holds all the properties of GAClust. The key features of the SDC algorithm are it (i) handles noise efficiently, (ii) discovers clusters automatically, and (iii) identifies both positively and negatively expressed genes. The algorithm gives a nice guideline to estimate parameters that vary from dataset to dataset.

3.4 Proposed methods

The section is divided into two major sub-sections. In Section 3.4.1, we describe a simple heuristic clustering algorithm i.e., GAClust in detail. After discussing the unsupervised method, the second part Section 3.4.2, focuses on two semi-supervised clustering algorithms, SDC and SGAClust.

3.4.1 Unsupervised full-space clustering algorithm

Given neighborhood distance threshold Υ (user specified parameter), GAClust proceeds in three steps, producing K number of clusters $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$ from input gene expression data $ED_{m \times n}$. The number of clusters and their size is highly influenced by the parameter Υ. GAClust is a graph-theoretic clustering algorithm, based on the clique graph and divisive approach. The divisive approach follows a top-down analysis. It initiates with a large cluster and gradually splits into small clusters until each cluster contains a single piece of data. The fundamental assumption of this model is the true biological partition of genes rely on certain functionality of the genes.

The similarity between all pairs of expression patterns can be represented by a similarity matrix $Sim_{m \times m}$, where $Sim(g_x, g_y)$ denotes the similarity in between gene g_x and g_y. This can be easily computed by proximity measure (similarity or dissimilarity). Further, the similarity matrix can be represented by a weighted graph $\mathcal{G}^*(V, E)$, where vertices V denote genes and E represents the edge set $E = \{(g_x, g_y)$ for $g_x, g_y \in V$ and $x \neq y\}$. The weight of an edge is defined by the similarity between two genes. A graph is said to be a clique graph if it consists of a disjoint complete graph [30]. In this context, the clique

graph is composed of clusters of genes where the similarity of each gene within the clique is higher than the genes belonging to other cliques. A clique graph $\mathcal{H}$ is formed by genes (vertices) $G = \{g_1, g_2, \ldots, g_m\}$, such that each clique $cq_i \in \mathcal{H}$ contains an edge between every two genes $g_i, g_p \in cq_i$. Additionally, there are no edges between genes g_i and g_k, where $g_i \in cq_i$ and $g_k \in G \setminus cq_i$. Mathematically a clique $\mathcal{H}$ graph for a given graph $\mathcal{G}^*(V, E)$ is defined in such a way that (i) each vertex of $\mathcal{H}$ presents a maximal clique of $\mathcal{G}^*$ and (ii) two distinct vertices of $\mathcal{H}$ are adjacent. Gene expression data is noisy in nature, hence an ideal clique graph is never possible. In expression data, contamination errors are introduced resulting in weighted graph $C(\mathcal{H})$ which is not a clique graph. Therefore, the clustering problem can be modeled as restoring clique graph $\mathcal{H}$ using edge modification problem from corrupted clique graph where the error is introduced. The implementation of GAClust is described next stepwise and the algorithm is shown in Algorithm 1.

(i) **Graph construction:** We compute an $m \times m$, $R_{m\times m}$ similarity matrix from expression data to construct a graph. The edge between two genes g_x and g_y has been given a weight using Equation 3.4.1 defined as similarity R, where $\mathcal{N}(g_x)$ is neighbors of gene g_x and $\mathcal{CN}(g_x, g_y)$ is common neighborhood of g_x and g_y. The R is stated as the similarity between two genes, where $a = |\mathcal{N}(g_x)|$, $b = |\mathcal{N}(g_y)|$, and $c = |\mathcal{CN}(g_x, g_y)|$.

$$R(g_x, g_y) = \begin{cases} 1 & \text{if } (g_x = g_y) \\ \frac{c}{a+b-c} & \text{if } (|\mathcal{CN}(g_x, g_y)| \neq 0) \\ 0 & \text{if } (|\mathcal{CN}(g_x, g_y)| == 0) \end{cases} \tag{3.4.1}$$

Definition 3.4.1 *Neighborhood of a gene, $\mathcal{N}(g_x)$ is described by the genes g_z, residing within its user-defined radius Υ.*

$$\mathcal{N}(g_x) = \{g_z | z \in G, Dist_{Euc}(g_x, g_z) \leq \Upsilon\} \tag{3.4.2}$$

$\mathcal{N}(g_x)$ is defined in Equation 3.4.2, where $Dist(g_x, g_z)$ is determined by Euclidean distance shown in Equation 3.4.3.

$$Dist_{Euc}(g_x, g_z) = \sqrt{(\sum_{j=1}^{n} (g_{xj}, g_{zj})^2)} \tag{3.4.3}$$

Algorithm 1: GAClust algorithm

Input : $ED_{m\times n}$ with a set of genes $G = \{g_1, g_2, \ldots, g_m\}$ and a set of samples $C = \{c_1, c_2, \ldots, c_n\}$, Υ, η
Output: $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$

$\mathcal{C} = \phi$
Compute R matrix with the help of Equation 3.4.1
while $(G \neq \phi)$ **do**
  $\mathcal{C}_{now} = \phi$, $\mathcal{A}(G) = 0$
  Select $g_x \in G$ such that $R(g_x, g_y) = \max\{R(g_w, g_y) | g_w, g_y \in G\}$
  $\mathcal{C}_{now} = \mathcal{C}_{now} \cup g_x$
  $G = G \setminus g_x$
  $\forall g_y \in G$, $\mathcal{A}(g_y) = \mathcal{A}(g_y) + R(g_y, g_x)$
  while *(Changes in* C_{now}*)* **do**
    while $(max\{\mathcal{A}(g_z) | g_z \in G\} \geq \eta |\mathcal{C}_{now}|)$ **do**
      Select $g_a \in G$ with maximum attraction such that $\mathcal{A}(g_a) = \max\{\mathcal{A}(g_w) | g_w \in G\}$
      $\mathcal{C}_{now} = \mathcal{C}_{now} \cup \{g_a\}$
      $G = G \setminus \{g_a\}$
      $\forall g_b \in G \cup \mathcal{C}_{now}$, $\mathcal{A}(g_b) = \mathcal{A}(g_b) + R(g_b, g_a)$
    end
    while $(min\{\mathcal{A}(g_z) | g_z \in \mathcal{C}_{now}\} < \eta |\mathcal{C}_{now}|)$ **do**
      Select $g_a \in G$ with minimum attraction such that $\mathcal{A}(g_a) = \min\{\mathcal{A}(g_w) | g_w \in G\}$
      $C_{now} = C_{now} \setminus \{g_a\}$
      $G = G \cup \{g_a\}$
      $\forall g_b \in G \cup C_{now}$, $\mathcal{A}(g_b) = \mathcal{A}(g_b) - R(g_b, g_a)$
    end
  end
  $\mathcal{C} = \mathcal{C} \cup \mathcal{C}_{now}$
end

Definition 3.4.2 *Common neighborhood between two genes* g_x *and* g_y *are the genes* $\{g_1, g_2, \ldots, g_q\}$ *which belong to the neighborhood of both genes,* g_x *and* g_y *with respect to* Υ *and is given by Equation 3.4.4.*

$$\mathcal{CN}(g_x, g_y) = \{g_k \in \mathcal{N}(g_x) \cap \mathcal{N}(g_y)\}, k = 1, 2, \ldots, q \tag{3.4.4}$$

The concept of common neighborhood is shown in Figure 3.2. The measure $R \in [0, 1]$ is symmetrical i.e., $R(g_x, g_y) = R(g_y, g_x)$. The value lies between 0 and 1, $0 \leq R \leq 1$. 0 means genes are not connected, 1 means the neighbors of g_x is overlapped with neighbors of g_y. Higher $R(g_x, g_y)$ i.e., values closer to one, indicates that the two neighbors are closely connected.

(ii) **Node addition:** The key step of GAClust is to compute the average at-

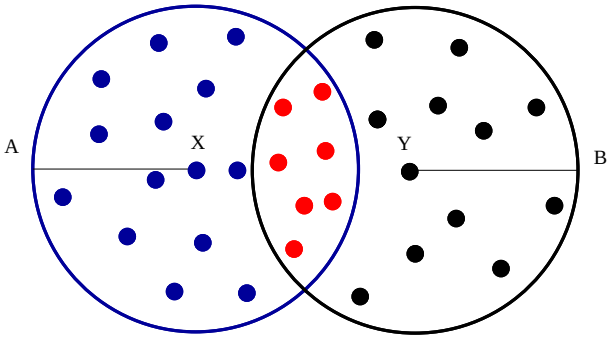


Figure 3.2: Schematic diagram of the common neighborhood between two objects. The blue and black colored circle represents the neighborhood of X and Y objects, respectively within its Υ distance. Red colored solid circles represent the common neighbor objects of both X and Y within Υ distance.

traction ($\mathcal{A}$) between unclustered data to its present cluster to make further decisions. Clusters are generated one at a time.

Definition 3.4.3 *The attraction ($\mathcal{A}$) of a gene g_x with respect to a cluster $\mathcal{C}_{now}$ is the sum of relation between g_x and all genes in $\mathcal{C}_{now}$.*

$$\mathcal{A}(g_x) = \sum_{g_y \in \mathcal{C}_{now}} R(g_x, g_y) \tag{3.4.5}$$

Definition 3.4.4 *A gene g_x is said to have high connectivity to be included in the current cluster $\mathcal{C}_{now}$ if it satisfies the condition $\mathcal{A}(g_x) \geq \eta|\mathcal{C}_{now}|$ where η is an attraction threshold.*

We initiate the current cluster by denoting $\mathcal{C}_{now} = \phi$. Clusters are formed by adding high connectivity genes one at a time to $\mathcal{C}_{now}$ until no changes have been found. It is important to note the relation of g_x with the genes presenting $\mathcal{C}_{now}$ is considered to compute $\mathcal{A}(g_x)$ and it is assumed at any given iteration of the algorithm the $|\mathcal{C}_{now}|$ is much less than the total number of genes m in the dataset.

The crucial task of the algorithm is parameter estimation of η and Υ. Unlike CAST [33], GAClust calculates threshold η dynamically with the help of Equations 3.4.6 and 3.4.7, where $deg(g_x)$ indicates the degree of a vertex V or gene g_x and $R(g_x, g_y)$ must be greater than 0.5. Here, we have taken $R(g_x, g_y) >= 0.5$ which will signify that at least 50% of neighbors of g_x and g_y are common.

$$deg(g_x) = \begin{cases} \sum_{y=1}^{m} 1 & \text{if } R(g_x, g_y) \geq 0.5 \text{ and } x \neq y \\ 0 & \text{otherwise} \end{cases} \tag{3.4.6}$$

$$\eta = 0.5 \times \frac{\sum_{x=1}^{m} \sum_{y=1, x \neq y}^{m} R(g_x, g_y)}{\sum_{i=1}^{m} deg(g_i)} \tag{3.4.7}$$

Definition 3.4.5 *A gene g_x is said to be low connectivity if it satisfies the following condition $\mathcal{A}(g_x) < \eta |\mathcal{C}_{now}|$.*

(iii) **Node deletion:** After the addition step, low connectivity genes are removed from the current cluster $\mathcal{C}_{now}$. We keep on removing genes from $\mathcal{C}_{now}$ until it gets stabilized to form a single cluster. Repeating the node addition and removal steps further, we get K number of clusters. All singleton clusters are considered as outliers. A gene is said to be an outlier if it behaves in a significantly different manner from the rest of the genes in a particular dataset. We have chosen the neighborhood distance Υ of a gene as sufficiently large (+0.5) from the graph of sorted K-Nearest Neighbor (KNN) distance from each gene. This K value is determined by taking the square root of the total number of genes present in an input dataset.

3.4.2 Semi-supervised full-space clustering algorithms

This section elaborates two semi-supervised algorithms SDC and SGAClust in detail. First, we discuss the SDC algorithm then the SGAClust algorithm. SDC is a density-based clustering algorithm that works in two phases: (i) preprocessing and (ii) clustering phase.

(i) **Preprocessing:** This step is initiated by normalizing (standard deviation (σ) 1 and mean (μ) 0) the gene expression data. Then, a discretization process discretizes the gene expression data and the discretized data (ED_{disct}) is fed as input to the clustering algorithm.

$$ED_{disct}(g_{i,1}) = \begin{cases} 2 & \text{if } ge_{i1} < 0 \\ 0 & \text{if } ge_{i1} = 0 \\ 1 & \text{if } ge_{i1} > 0 \end{cases} \tag{3.4.8}$$

$$ED_{disct}(g_{i,j}) = \begin{cases} 2 & \text{if } ge_{ij} < ge_{i(j-1)} \\ 0 & \text{if } ge_{ij} = ge_{i(j-1)} \\ 1 & \text{if } ge_{ij} > ge_{i(j-1)} \end{cases} \tag{3.4.9}$$

In discretization, each cell ge_{ij}, (where $j = 1$) of the gene expression data (ED) for the first condition is discretized by using Equation 3.4.8 and for the other conditions $(n - j_1)$, each cell ge_{ij} (where $j = 2, 3, \ldots, n$) is computed using Equation 3.4.9. Each gene in ED_{disct} now has a pattern of regulation values of 0^s, 1^s, and 2^s across condition known as regulation pattern. After the computation of each gene's regulation pattern, the next job is to calculate the match (M) between genes g_x and g_y stated in Equation 3.4.10.

Definition 3.4.6 ***Match:*** *Match (M) gives the number of common regulation values according to the conditions except the first one, which signifies how similar the two patterns are with respect to their expression values.*

If $M = n-1$, it can be said that the two patterns are almost similar. The match between g_x and g_y is calculated as below.

$$Pat_j^{g_x,g_y} = \begin{cases} 1 & \text{if } ED_{disct}(g_{x,j}) = ED_{disct}(g_{y,j}), \text{ where } j = 2, \ldots, n \\ 0 & \text{otherwise} \end{cases} \tag{3.4.10}$$

$$M(g_x, g_y) = \text{number of } 1^s \text{ in } Pat_j^{g_x,g_y} \tag{3.4.11}$$

Definition 3.4.7 ***Maximal Match:*** *If match between g_x and g_y is equal or greater than the minimum matching threshold value i.e., δ, ($M(g_x, g_y) >= \delta$) and no other gene exists whose M with respect to g_x is greater than g_y, then g_x has a Maximal Match (MM) with another g_y ($g_x \neq g_y$).*

Definition 3.4.8 ***Maximally Matched Regulation Pattern:*** *For genes g_x and g_y, let g_x be maximally matched with g_y, then the Maximally Matched Regulation Pattern ($MMRP$) is computed using Equation 3.4.12 by considering the subset (two gene profiles may not match throughout $n-1$ conditions) of conditions where they maximally matched based on δ.*

$$MMRP(g_{x,j}) = MMRP(g_{y,j}) = \begin{cases} 2 & \text{if } ED_{disct}(g_{x,j}) = 2 = ED_{disct}(g_{y,j}) \\ 0 & \text{if } ED_{disct}(g_{x,j}) = 0 = ED_{disct}(g_{y,j}) \\ 1 & \text{if } ED_{disct}(g_{x,j}) = 1 = ED_{disct}(g_{y,j}) \\ X & \text{otherwise} \end{cases} \tag{3.4.12}$$

Here, $j = 2, 3, \ldots, n$. Therefore, for the whole set of j conditions, we obtain an $MMRP$ pattern of 0^s, 1^s, 2^s, and X^s.

Definition 3.4.9 ***Negative Maximally Matched Regulation Pattern:*** *The Negative Maximally Matched Regulation Pattern (NMMRP) of g_y is determined by comparing the MMRP of g_x as stated in Equation 3.4.13.*

$$NMMRP(g_{y,j}) = \begin{cases} 2 & \text{if } MMRP(g_{x,j}) = 1 \\ 1 & \text{if } MMRP(g_{x,j}) = 2 \\ 0 & \text{if } MMRP(g_{x,j}) = 0 \\ X & \text{if } MMRP(g_{x,j}) = X \end{cases} \tag{3.4.13}$$

Therefore, we obtain a $NMMRP$ pattern for j conditions ($j = 2, 3, \ldots, n$).

Definition 3.4.10 ***Rank:*** *Rank gives the ascending order of expression levels of a gene across conditions.*

Rank is measured by giving a ranked value starting from 1 to all the expression values in the $MMRP$ pattern except for those conditions having a X value. The working examples of the computation of discretized data, M, MM, $MMRP$, $NMMRP$, and *Rank* are available in `http://agnigarh.tezu.ernet.in/~rosy8/workingexampleSDC.pdf`.

(ii) **Clustering:** The second phase of SDC is based on some of the fundamental concepts of DBSCAN. The following definitions are trivial to the clustering process.

Definition 3.4.11 ***ε-neighbor:*** *ε-neighbors with respect to $g_i \in G$, are those genes $g_k \in G$, which have more similarity than the user-defined threshold (ε) as shown in Equation 3.4.14. Here we have used combined similarity which is mentioned in Equation 3.4.15.*

$$\varepsilon - neighbors(g_i) = \{g_k | \text{where} g_k \in G \text{ and } Com\_sim(g_i, g_k) >= \varepsilon\} \tag{3.4.14}$$

In this method, we combine similarity measure (Sim) and semantic similarity (SS) to improve clustering result. We find the combined similarity (Com_sim) between two genes g_i and g_k given next.

$$Com\_sim(g_i, g_k) = w1 * Sim(g_i, g_k) + w2 * SS(g_i, g_k) \tag{3.4.15}$$

Here, $w1 + w2 = 1$ and $0 \leq w2 \leq 1$ [192]. Weight factors $w1$ and $w2$ control the weights of two similarity measures. Most commonly used proximity measure is Euclidean distance which gives the dissimilarity between two genes as Equation 3.4.3 [164]. We first convert $Dist_{Euc}(g_i, g_k)$ into a similarity measure as, $Sim(g_i, g_k) = \frac{1}{1+Dist_{Euc}(g_i,g_k)}$. For SS, Lin's semantic similarity is taken under consideration. Next, we present the definition of core neighbors which has been extended from the definition of core neighbors given in [101].

Definition 3.4.12 ***Core neighbors:*** *Core neighbors (N_c) of a gene $g_i \in G$ is described by a set of genes $G^* \in G$ and should satisfy the following four criteria. A gene, say g_i is considered as core gene if.*

(a) $\forall g_y \in G^*, g_y \in \varepsilon - neighbors(g_i)$

(b) $MMRP(g_y) \approx MMRP(g_i)$

(c) $Rank(g_y) \approx Rank(g_i)$

(d) $|G^*| >= M_p$ *(minimum points: a user-defined threshold)*

To compute the N_c of a particular gene g_i, we check the above-mentioned four criteria for all the $n-1$ dimensions (except condition 1). If we do not get the N_c, we keep on checking the criteria by reducing the search space one condition at a time. At first we reduce the condition set by $n - \{j_l\}$ the last condition i.e., $n - 1 - 1 = n - 2$. If we still do not find the N_c of g_i, we further reduce the search space by $n - 3$ and so on. The following definitions have been extended from the concept of DBSCAN algorithm [101].

Definition 3.4.13 ***Direct density reachable:*** *g_x is direct density reachable with respect to g_y if it fulfills three basic principles.*

(a) *g_y must be a core-gene or g_y must have N_c.*

(b) $g_x \in \varepsilon - neighbors(g_y)$

(c) $MMRP(g_x) \approx MMRP(g_y)$

Definition 3.4.14 ***Density reachable:*** *Gene g_q is density reachable from g_p provided there is a chain of genes $g_1, g_2, g_3, \ldots, g_m$ such that $g_1 = g_p$ and $g_m = g_q$ and every g_{i+1} gene is directly density reachable from g_i^{th} gene.*

Definition 3.4.15 ***Connected:*** *Gene g_x is connected to g_y with respect to ε, provided g_x, g_y are reachable from a common gene say g_k.*

Definition 3.4.16 ***Cluster:*** *A cluster $\mathcal{C}_k$ ($|\mathcal{C}_k| >= M_p$) is a collection of reachable and connected genes. Say, a gene $g_x \in \mathcal{C}_k$ and the gene g_y is found to be reachable from g_x, then g_y must be in cluster $\mathcal{C}_k$. Similarly, if a gene $g_x \in \mathcal{C}_k$ and g_y is connected to g_x then g_y will be in the same $\mathcal{C}_k$ cluster.*

Definition 3.4.17 ***Noise:*** *A noise gene is a gene which does not belong to any cluster.*

In case of pairs of core genes, direct density reachable holds symmetric relation. Connected also holds symmetric property.

The SDC is explained in Algorithm 2. The algorithm proceeds by arbitrarily selecting an unclustered gene g_i. Thereafter, it finds $MMRP(g_i)$ and $Rank(g_i)$. According to definition 3.4.12, core-neighbors of g_i are considered which belong to the cluster (say $\mathcal{C}_k$). The cluster expansion of $\mathcal{C}_k$ is done by repeating the process of finding all connected and density reachable genes for each core neighboring point. Next, we find $NMMRP$ from $MMRP$ of the newly formed cluster to get negatively co-expressed genes. Genes G^u which match with $NMMRP$ are again considered for further processing. We also keep core neighbors of $g_x \in G^u$ in the same cluster, $\mathcal{C}_k$. Cluster $\mathcal{C}_k$ is expanded until density connected genes are found completely. The process then restarts with a new unclustered gene to form a new cluster. Genes that are not a member of any of the clusters are marked as noise.

SGAClust is the extended version of the GAClust algorithm. SGAClust takes four input parameters neighborhood similarity threshold Υ', attraction threshold η', w_1, and w_2. Here, we incorporate combined similarity (Com_sim) as shown in Equation 3.4.15 instead of only an expression-based distance measure to find the neighborhood of a gene. Here, we consider similarity measure Sim (considering Euclidean distance) and Wang's measure as SS to compute Com_sim. Wang's measure distinguishes the two relations (is_a and part_of) in GO hierarchy structure whereas Lin's measure does not differentiate between both the relations. We redefine the definition of neighborhood of a gene.

Definition 3.4.18 *Neighborhood of a gene $\mathcal{N}(g_i)$ is described by the genes g_x, residing within its user defined radius Υ'.*

$$\mathcal{N}(g_i) = \{g_x | x \in G, Com\_sim(g_i, g_x) \geq \Upsilon'\} \tag{3.4.16}$$

SGAClust algorithm considers Υ' and η' instead of Υ and η in GAClust. The parameters are calculated by the following Equations 3.4.17 and 3.4.18, where Υ

Algorithm 2: SDC algorithm

Input : $ED_{m\times n}$ with a set of genes $G = \{g_1, g_2, \ldots, g_m\}$ and a set of samples $C = \{c_1, c_2, \ldots, c_n\}$, w_1, w_2, δ, ε, M_p

Output: $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$

```
C = φ
Compute Com_sim between all pairs of genes using Equation 3.4.15
foreach g_a ∈ G do
    C_k = φ
    Start with an random unclustered gene say g_i
    Find the MMRP(g_i) and Rank(g_i)
    Find N_c^{g_i} of g_i using definition 3.4.12
    foreach g_c ∈ N_c^{g_i} do
        Identify all connected and reachable genes G^c with respect to g_c
        Add G^c to C_k
    end
    Find the NMMRP from MMRP of the newly formed C_k
    Find the unclustered genes G^u which matches the NMMRP
    foreach g_x ∈ G^u do
        Find the N_c^{g_x} of gene g_x using definition 3.4.12
        foreach g_q ∈ N_c^{g_x} do
            Identify all reachable and connected genes G^q with respect to
             g_q
            Add G^q to C_k
        end
    end
    Add C_k to C
end
All the unclustered genes are marked as noise
```

and η are estimated as mentioned in GAClust algorithm.

$$\Upsilon' = \frac{1}{1+\Upsilon} \tag{3.4.17}$$

$$\eta' = \frac{1}{1+\eta} \tag{3.4.18}$$

The main algorithmic approach is similar to GAClust. It consists of three major steps, i.e., graph construction, node addition, and node deletion. The graph construction is similar to the GAClust algorithm except for finding neighborhood of a gene. We compute an $m \times m$, $R_{m\times m}$ similarity matrix for a weighted graph. The weight of an edge between two genes g_i and g_k has given by $R(g_i, g_k)$ as mentioned in Equation 3.4.1. Each cluster is generated by adding high connectivity genes and removing low connectivity genes from the cluster.

3.5 Time complexity

GAClust algorithm takes $O(m^2)$ operations to compute $R_{m\times m}$ matrix. In the best case, node addition and node deletion takes much lesser time than the computation of the similarity matrix. The most crucial task of node addition and deletion is to compute the attraction of a gene with the currently formed cluster. Each gene is taken under consideration and sums of similarity between a gene to all other genes in a current cluster are compared with a threshold. In the worst case scenario, this operation may take $O(m^2)$ time, if all the genes come under a single cluster. Therefore, the overall running time of the GAClust algorithm is $O(m^2)$. The algorithmic approach of SGAClust is similar to GAClust except for the similarity matrix. For SGAClust, we compute proximity measure as well as semantic similarity measure. Hence, the creation of $R_{m\times m}$ matrix will take $O(2 \times m^2) \approx O(m^2)$ time. Therefore, the total running time of the SGAClust algorithm is $O(m^2)$.

The SDC algorithm computes Com_sim between every pairs of genes in $O(m^2)$ time, as mentioned above. In the worst case, the algorithm takes $O(m^2)$ time to check each ε-neighborhood of each data point. This is only for positively co-expressed pattern search. Next, it takes a maximum $O(m^2)$ time to find negatively co-expressed patterns. Therefore the total complexity of the SDC algorithm is $O(m^2)$.

In addition to our proposed algorithms, we also analyze the time complexity of other algorithms under study. For the CAST algorithm, let us consider that there is $\log(m)^r$ number of partitions, where r is constant. Each partition leads to a cluster consisting of all vertices V. $O(m\log(m))$ number of edges are considered in each partition. At most once, each gene is taken under consideration because sums of disjoint edges are compared with a threshold. Therefore, the distance operation requires $O(m^2)$ time in between the input graph and each of the cliques. The algorithm takes $O(m^2(\log(m))^r)$ time [33]. The time complexity of the K-means algorithm is $O(m \times K \times i)$, where K is the number of clusters and i is the number of iterations. The computational complexity of HC and CLICK algorithms are $O(Kn^2)$ and $O(mn^{2/3})$, respectively. Algorithm SOTA clusters data in approximately linear time.

3.6 Performance analysis

In order to provide a comparison of all three proposed algorithms, we select a suite of clustering algorithms K-means, HC, CAST [33], SOTA [138], and CLICK

[293] which are applied on synthetic data as well as real gene expression datasets. The performance of algorithms is established by the means of internal criteria and biological assessment. The internal measure is a pure indication of how many groups are really present in a dataset, i.e., how well the partition solution is produced by a clustering algorithm that captures the separation of data amongst different clusters. It is actually useful when we do not know the true clustering solutions. Each clustering result produced by different algorithms on several datasets is assessed with four commonly used cluster validation indices to judge the quality of clusters. They are MSE [1], DB [75], BH [26], and CI [146] which are explained in Chapter 2. Lower MSE, DB, BHI, and CI values suggest a good clustering result.

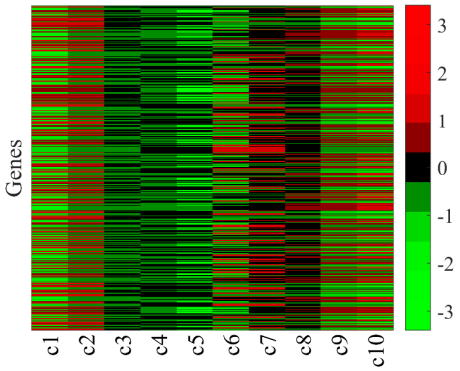
To generate clusters by CAST and SOTA, we have used the MultiExperiment Viewer (MeV) available at `http://mev.tm4.org/` algorithms with default parameter settings. CLICK algorithm is executed as a part of Expander software version 7.0 (`http://acgt.cs.tau.ac.il/expander/`) with default homogeneity value as mentioned in the software. K-means and HC average linkage is executed in MATLAB. Our own methods are also implemented in the MATLAB environment.

3.6.1 Results on synthetic datasets

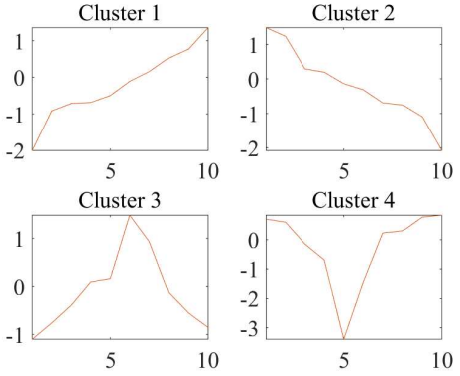
For a better explanation and to establish the effectiveness of GAClust, we firstly generate five synthetic datasets which can be visualized in Figure 3.3. Each dataset is comprised of 400 genes where we implant four clusters of 100 genes in each of them. At first, we create a background matrix of size 400 rows and 10 columns from a normal distribution of μ 0 and σ 1. We implant four different types of clusters where the first cluster has up-regulated patterns (Cluster 1), the second cluster has down-regulated patterns (Cluster 2), the third cluster has up-regulated then down-regulated patterns (Cluster 3), and the fourth one has down-regulated then up-regulated patterns (Cluster 4) as shown in the Figure 3.3. To create the up-regulated cluster, we randomly select one gene expression profile and sort the expression values in ascending order. Then we replicate the same expression profile for randomly other 99 genes. Similarly, we create down-regulated patterns except for the expression values which are necessarily in descending order. For Cluster 3 first half of the expression values are up-regulated for the first five columns and then down-regulated for the next five conditions and vice versa for cluster 4. Here, one point we need to keep in mind is that we create the clusters in non-overlapping manner. Thus we create a matrix

say D1. To make the datasets more realistic, next, we add random noise from a normal distribution with μ 0 and varying σ 0.25, 0.5, 0.75, and 1 with each of the cells of D1 to get matrices D2, D3, D4, and D5, respectively. Before applying clustering algorithms, we normalize the datasets by z-scores.

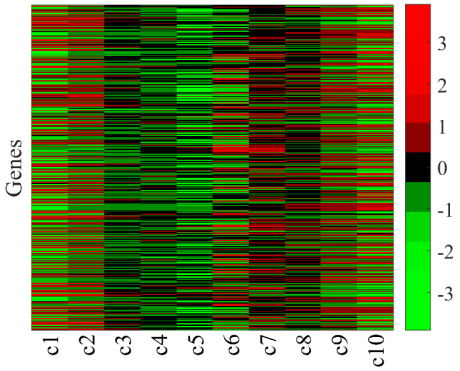
Running K-means and HC, a user-specified number of clusters K is re-



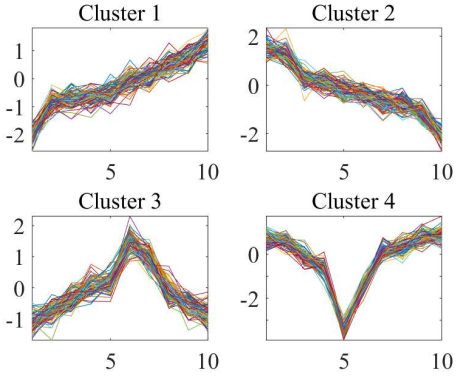
(a) Heatmap of all genes in simulated data D1.



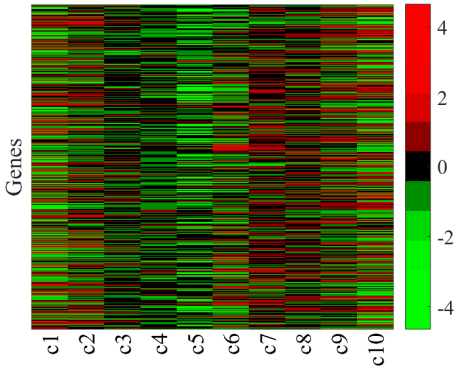
(b) Co-expression profiles of gene clusters for D1.



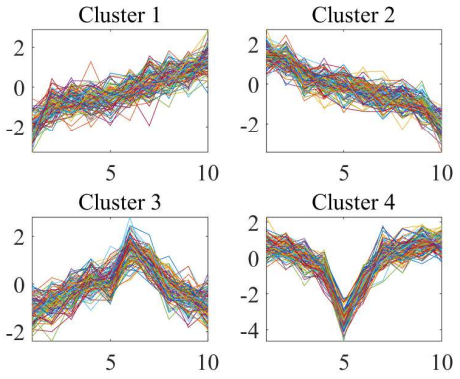
(c) Heatmap of all genes in simulated data with noise 0.25 D2.



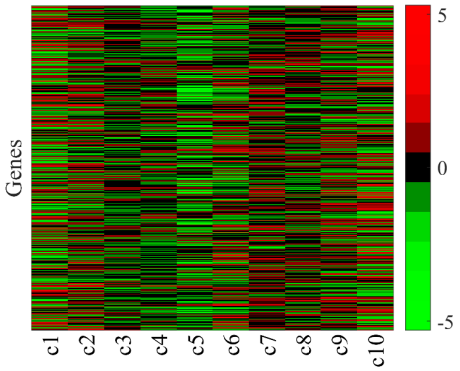
(d) Co-expression profiles of gene clusters for D2.



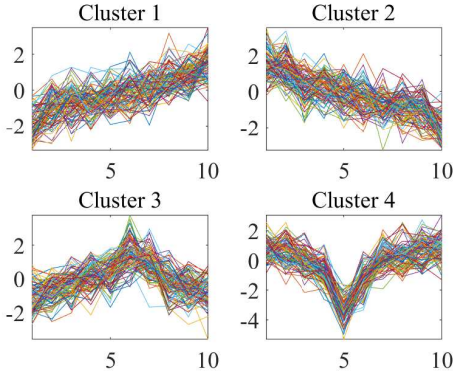
(e) Heatmap of all genes in simulated data with noise 0.5 D3.



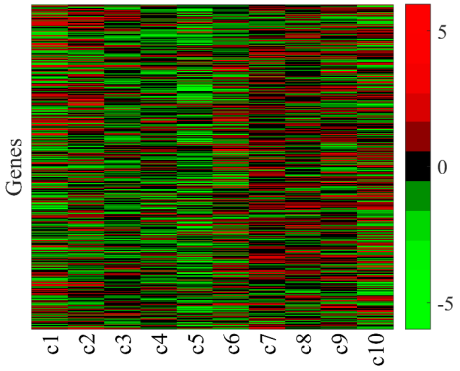
(f) Co-expression profiles of gene clusters for D3.



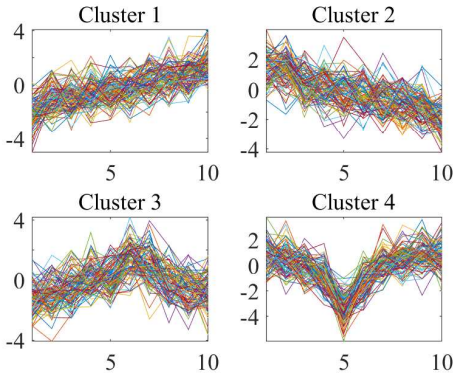
(g) Heatmap of all genes in simulated data with noise 0.75 D4.



(h) Co-expression profiles of gene clusters for D4.



(i) Heatmap of all genes in simulated data with noise 1 D5.

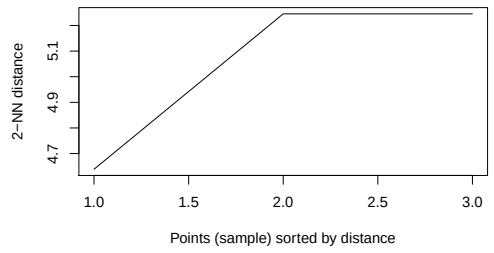


(j) Co-expression profiles of gene clusters for D5.

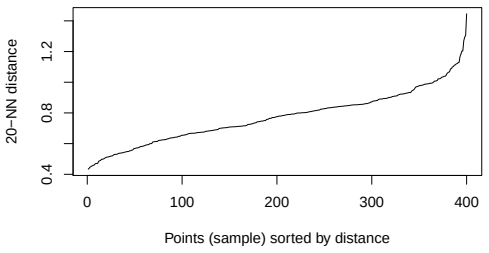
Figure 3.3: Synthetic gene expression data with 400 genes and 10 samples with and without noise are shown in the left column. The right column denotes the corresponding profiles of four gene clusters. The x-direction shows the samples or conditions and the y-direction denotes the genes.

quired. Therefore, we use implanted true number of clusters as K for synthetic datasets to obtain K number of clusters. CLICK algorithm returns partitions leaving some data unclustered. We have considered those data as a single cluster in order to compute all internal validation indices [89]. To determine the parameter Υ for each synthetic dataset we plot the graph of sorted KNN distance from each gene in Figure 3.4. The parameter Υ for GAClust is kept relatively large which is given in Table 3.1. The attraction threshold η is computed dynamically according to Equations 3.4.6 and 3.4.7 except for dataset D1 because it has a replication of data in four clusters. Hence, we consider $K = \sqrt{4} = 2$ of KNN for dataset D1. The η is calculated using Equation 3.4.7, where multiplication factor is 1 instead of 0.5, as well as Υ, is increased with 1. It is important to mention that in this experiment we have not considered semi-supervised algorithms for

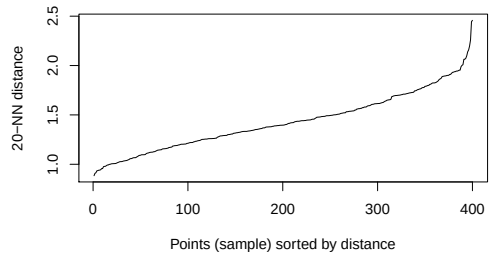
synthetic datasets as there is no GO information for synthetic data.



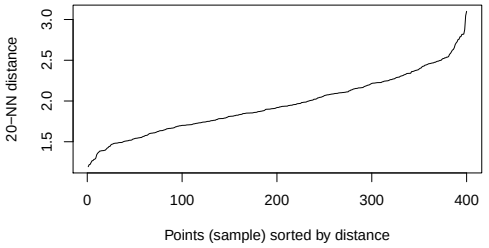
(a) Graph of sorted KNN distance for D1.



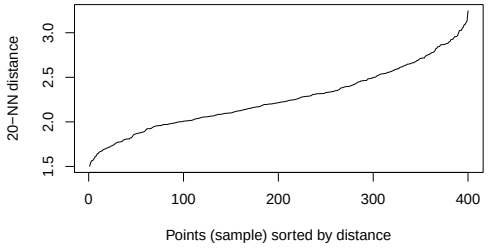
(b) Graph of sorted KNN distance for D2.



(c) Graph of sorted KNN distance for D3.



(d) Graph of sorted KNN distance for D4.



(e) Graph of sorted KNN distance for D5.

Figure 3.4: Determination of Υ of GAClust for synthetic data by the graphs of sorted KNN distance.

We now analyze the performance of GAClust with all other algorithms under consideration with respect to internal measures using MSE, DB, BH, and CI described in Section 2.2.3. Figure 3.5 shows the histograms of four cluster validation indices for comparing the performance of six clustering algorithms on synthetic datasets. In the diagram, the x-axis denotes the clustering algorithms while the y-axis denotes the metric values. Different colors are being used to differentiate different clustering algorithms. The graph demonstrates that GAClust is able to identify clusters in presence of a higher amount of noise. The internal

Table 3.1: Parameter settings of GAClust for synthetic datasets.

Dataset	Attraction threshold (η)	Neighborhood distance (Υ)
D1	0.6241	5.2 + 1
D2	0.4853	1.3 + 0.5
D3	0.4604	2.3 + 0.5
D4	0.3687	3 + 0.5
D5	0.3278	3.1 + 0.5

metric increases with the increasing noise. Now, if we look closely at the figure, then it can be understood that GAClust outperforms all the algorithms (For Dataset D3 CAST performs better than GAClust) in terms of DB score where a lower metric signifies better performance. While comparing GAClust with all other methods, it performs similar to CAST and is sometimes inferior for some datasets based on MSE and BH values. On the other hand, GAClust performs slightly lesser than CAST, K-means, and HC for datasets D4 and D5 on the basis of CI.

For better understanding, we summarize the values of metrics for all clustering algorithms on all five datasets in Table 3.2. MSE and BH scores give a similar rank for all the algorithms. CAST performs best followed by GAClust, SOTA, K-means, HC while the CLICK algorithm performs the worst. From the table, it is not too hard to recognize that CLICK holds the last position for DB, MSE, and BH and the second last position followed by SOTA for CI score. K-means is the second-best algorithm and SOTA and CLICK both are not performing well according to CI and DB scores. As for all the measures, K-means and HC are quite close in many circumstances. Based on Table 3.2, it appears that K-means is slightly superior to HC. It appears that the CAST algorithm has very good predictive power and GAClust is competitive in comparison with CAST as well as other state-of-the-art methods. It is important to note that CI is greatly influenced by the fact of producing optimal index values for the different number of clusters. Therefore, CI does not perform well to evaluate the clusters generated by different algorithms. The assessment of all three of our proposed algorithms for the real datasets is given next.

3.6.2 Results on real datasets

To examine the capability of clustering algorithms, we test all three algorithms on five different Affymetrix cancer gene expression datasets. The description of

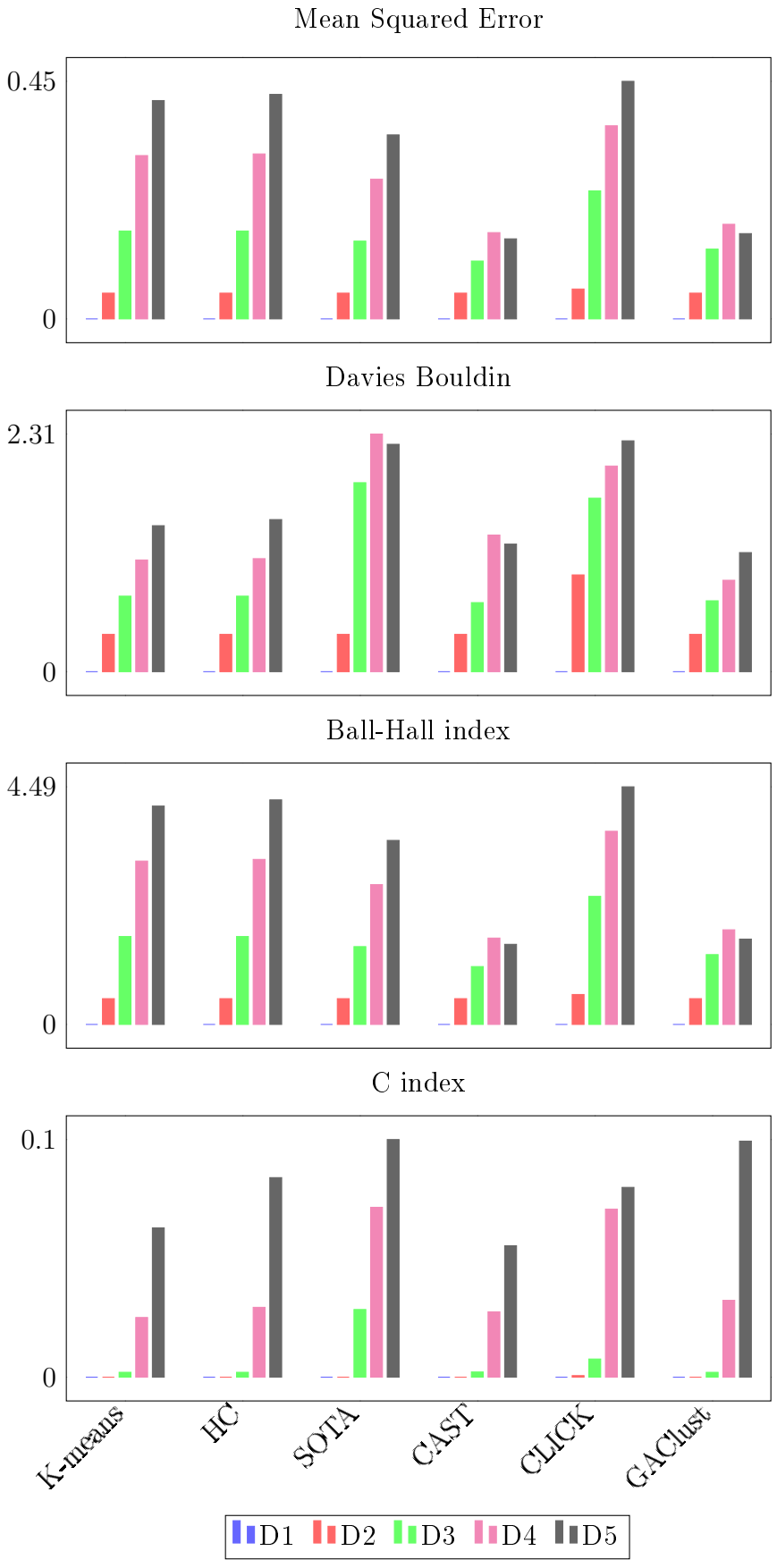


Figure 3.5: Histogram of different cluster validation indices on five synthetic datasets.

Table 3.2: Average internal measure on five synthetic datasets.

Algorithm	MSE	DB	BH	CI
K-means	0.1874	0.7211	1.8739	0.0182
HC	0.1904	0.7356	1.9036	0.0233
SOTA	0.1617	1.3454	1.6169	0.0404
CAST	0.0946	0.7212	0.9462	0.0172
CLICK	0.2227	1.3757	2.2270	0.0321
GAClust	0.1043	0.6204	1.0423	0.0270

the gene expression datasets is summarized in Table 3.3. The reported datasets are obtained from Affymetrix chips. The description of the table consists of the name of the dataset (first column), type of tissue (second column), number of genes (third column), number of samples (fourth column), and number of classes (fifth column). The microarray gene expression datasets which were already preprocessed by De Souto et al. [76] are taken from a website `https://schlieplab.org/Static/Supplements/CompCancer/`. Before applying clustering algorithms, we normalize all the datasets using z-score to μ 0 and σ 1. Next, we describe the datasets in detail.

Table 3.3: A brief description of cancer gene expression datasets.

Dataset	Tissue type	Genes	Samples	Classes	Ref
Armstrong-v2	Blood	2194	72	3	[18]
Bhattacharjee	Lung	1543	203	5	[39]
Laiho	Colon	2202	37	2	[184]
Ramaswamy	Multi-tissue	1363	190	14	[280]
Singh	Prostate	339	102	2	[303]

Armstrong-v2: The dataset is the gene expression profile of lymphoblastic leukemias of size 12582 $\times$ 72. After preprocessing the filtered dataset is of 2194 $\times$ 72 sizes. The leukemic samples are 28 acute myeloid leukemia (AML), 24 acute lymphoblastic leukemia (ALL), and 20 mixed lineage leukemia (MLL).

Bhattacharjee: One of the leading cause of death in USA and worldwide is lung carcinoma. The dataset is obtained from Oligonucleotide microarrays consisting of 12600 transcript sequences in 203 lung tumors. The size of the final dataset is 1543$\times$203. The commonly known type of lung cancer is small-cell lung carcinomas (SCLC) or non-small-cell lung carcinomas (NSCLC). NSCLC is further subdivided into adenocarcinomas (AD), squamous cell carcinomas, and large-cell carcinomas. Among all the samples 186 tumor lung samples including

139 adenocarcinomas (AD), 6 small cell lung carcinomas (SMCL), 21 squamous cell lung carcinoma (SQ), 20 pulmonary carcinoids (COID), and 17 normal lung specimens [39].

Laiho: This Affymetrix microarray gene expression data contains 222883 probe sets. There are a total of 37 samples out of 8 is serrated colorectal carcinomas (SCRC) and 29 is conventional CRC (CCRC). Dataset is preprocessed to get 2202 genes for further experiments. Serrated and conventional CRC both are different morphologically and biologically [184]. Clinically and pathologically it is suggested that serrated CRC can be more aggressive than a conventional one. Therefore, patients suffering from serrated CRC has a poor survival rate.

Ramaswamy: This dataset is a result of oligonucleotide microarrays with 16063 genes which is reduced to 1363 number of genes [280]. Ramaswamy dataset is composed of 190 multi-class samples which are categorized into fourteen different tumor types: 11 samples of breast adenocarcinoma, 10 samples of prostate adenocarcinoma, 11 samples of lung adenocarcinoma, 11 samples of colorectal adenocarcinoma, 22 samples of lymphoma, 10 samples of melanoma, 11 bladder transitional cell carcinoma, 10 samples of uterine adenocarcinoma, 30 samples of leukemia, 11 samples of renal carcinoma, 11 samples of pancreatic adenocarcinoma, 11 samples of ovarian adenocarcinoma, 11 samples of pleural mesothelioma, and 20 samples of the central nervous system (CNS).

Singh: Clinically and histologically one of the most heterogeneous cancer types is prostate cancer. Oligonucleotide microarray is used approximately 12600 probe id to acquire gene expression data. The dimensionality of this dataset is reduced to 339 genes. High-throughput technology derives data from 52 prostate tumor samples and 50 nontumor prostate samples.

To investigate the comparative performance of GAClust, SDC, and SGA-Clust on cancer gene expression datasets, we execute K-means, HC, SOTA, CAST, and CLICK as the competing methods. To obtain the optimal number of clusters for two widely used traditional clustering algorithms, K-means and HC, we execute these algorithms on real data with K values ranging from 2 to 50. Afterwards, K value is chosen in such a way, where the DB clustering index is minimized [1]. In practice, we can cut the dendrogram at any level to get the desired number of clusters. But, for a fair comparison, we have done this exhaustive experimentation. We plot DB scores for each of the clustering algorithms generated by K-means and HC in Figures 3.6 and 3.7, respectively.

We apply CAST, SOTA and CLICK with default parameter settings on real datasets.

As mentioned previously for synthetic datasets, we proceed in a similar manner for deciding the input parameter Υ of GAClust for real datasets. In this case, we again plot sorted KNN graph for every dataset depicted in Figure 3.8 [101]. With the help of this figure, visually we can predict the Υ value for GAClust. Here, K value of KNN graph is considered to be M_p and $\frac{1}{1+\Upsilon}$ as ε for SDC algorithm. We keep the value of δ as minimum as possible. The default value of δ is 3 for the SDC algorithm. We use MATLAB and R implementation for Lin's [350] and Wang semantic similarity measure [358], respectively. For Lin's measure, we download the gene ontology file (released on 2016-09-10) and annotation file of *Homo Sapience* from `www.geneontology.org`. We keep the values $w1 = 0.6$ and $w2 = 0.4$ for both SDC and SGAClust algorithms, as we want to give more weightage on proximity measure than semantic similarity measure. In SGAclust, ε is used as Υ'. Additionally, it is important to note that η is calculated dynamically for GAClust from where η' is estimated. The parameter settings of all three algorithms can be found in Table 3.4.

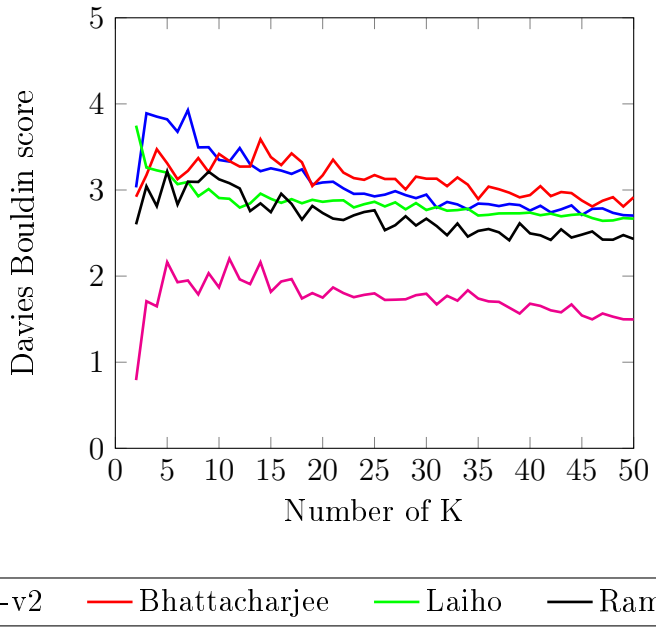


Figure 3.6: Selection of K for K-means algorithm with respect to Davies Bouldin score for cancer gene expression datasets.

Figure 3.9 are the results of these competing algorithms under various evaluation criteria on five cancer gene expression datasets. In addition to this, we summarize the results by taking the average across all datasets and reported

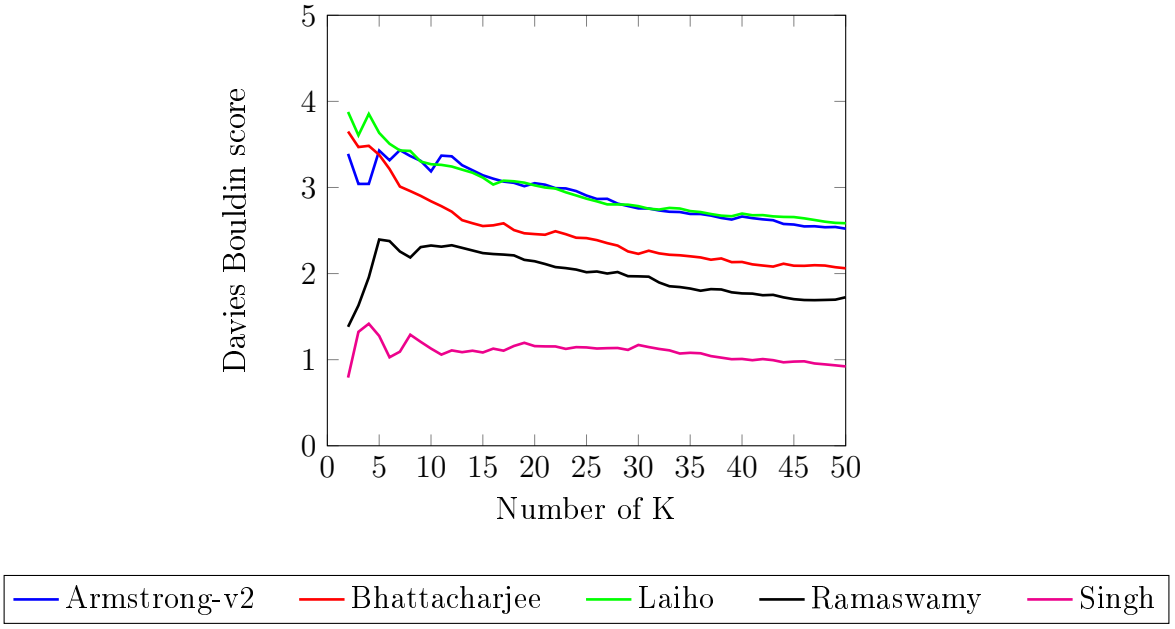
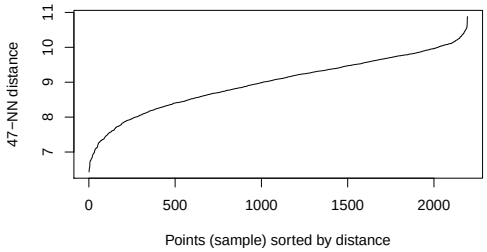


Figure 3.7: Selection of the number of clusters for hierarchical clustering with respect to Davies Bouldin score for real datasets.

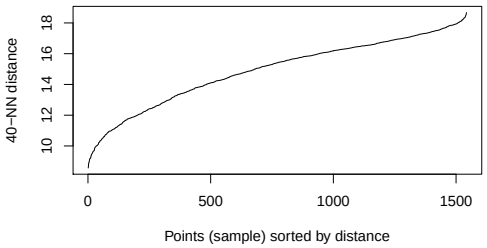
Table 3.4: Parameter settings of GAClust for cancer gene expression datasets.

Dataset	GAClust		SDC		SGAClust	
	η	Υ	M_p	ε	η'	Υ'
Armstrong-v2	0.2911	10.5 +0.5	47	0.08	0.8362	0.08
Bhattacharjee	0.3429	18 + 0.5	40	0.05	0.8794	0.05
Laiho	0.2945	6.8 + 0.5	47	0.12	0.7067	0.12
Ramaswamy	0.3523	18 + 0.5	37	0.05	0.7166	0.05
Singh	0.3622	12 + 0.5	19	0.07	0.8711	0.07

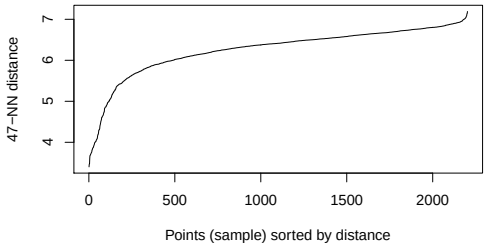
in Table 3.5. From Table 3.5, we can see that unsupervised clustering algorithm CAST achieves the best performance among all other methods for all five datasets together across two validation indices i.e., CI and DB. The possible reason for performing the best result is to identify more singleton clusters as outliers. On the other hand, SGAClust is considered to be the best performer with respect to MSE and BH indices. Although SDC provides the best result, still we have not considered it as the best one because of the 'NAN' value for the Bhattacharjee dataset. If we closely observe the figure, we can see that individually for each dataset SDC gives the lowest values for MSE and BH indices. GAClust is the second-best performer across all datasets with respect to CI and DB. With the help of CI, semi-supervised clustering algorithms do not perform well contrasting



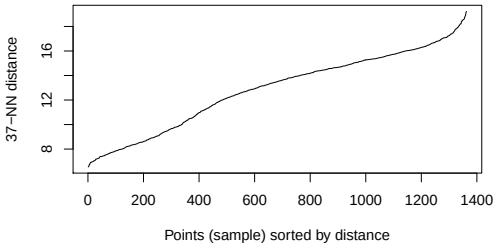
(a) Graph of sorted KNN distance for Armstrong-v2 data.



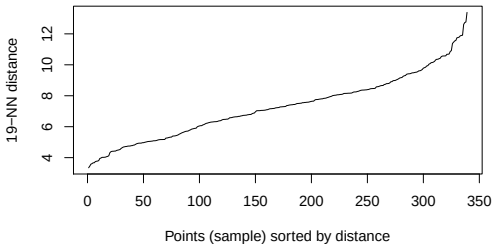
(b) Graph of sorted KNN distance for Bhattacharjee dataset.



(c) Graph of sorted KNN distance for Laiho dataset.



(d) Graph of sorted KNN distance for Ramaswamy dataset.



(e) Graph of sorted KNN distance for Singh dataset.

Figure 3.8: Determination of Υ of GAClust for real data by the graphs of sorted KNN distance.

with all unsupervised methods. Our proposed algorithm GAClust always take the immediate next position after the CAST algorithm for all four measures. In comparison to K-means and HC, both the algorithms perform very similarly mainly for MSE and BH, as can be observed from the table. In some of the datasets, K-means and HC perform very closely which can be easily observed from Figure 3.9. Regarding the clustering algorithms SOTA and CLICK, it also provide similar values for 2 indices (MSE and BH) out of 4 indices. Overall we

can say that CLICK and SOTA give similar types of results for all the datasets. We note that these two algorithms are not good for analyzing gene expression data as they degrade the cluster quality. In summary, these outputs suggest that GAClust is more advantageous than K-means, HC, CLICK, and SOTA and as good as the CAST algorithm. Semi-supervised algorithms SDC and SGAClust are better than any other unsupervised algorithms with reference to MSE and BH. For the other two measures, i.e., CI and DB the semi-supervised algorithms perform differently. It gives poor performance for CI and GAClust shows better results than semi-supervised algorithms for the DB index. It can be observed that the CAST algorithm generates a huge number of clusters whereas GAClust has less number of clusters than CAST. SGAClust also identifies less number of clusters than GAClust. Another important observation is to detect more singleton clusters in GAClust and SGAClust. This creates a difference in CI for CAST, GAClust, and SGAClust. It is noteworthy that the SGAClust algorithm is better than the SDC algorithm.

Table 3.5: Average internal measures on five cancer gene expression datasets.

Algorithm	MSE	DB	BH	CI
K-means	0.5317	2.2720	62.9152	0.1693
HC	0.5329	1.8682	62.3143	0.1259
SOTA	0.6597	3.2350	80.2281	0.1984
CAST	0.0599	0.8048	6.7760	0.0573
CLICK	0.6605	2.8258	81.6232	0.1281
GAClust	0.1182	1.0797	14.7863	0.1061
SDC	0.0266	1.3402	2.7707	0.5926
SGAClust	0.0544	1.2665	6.3043	0.3731

Enrichment analysis: Due to the biological complexity, enrichment analysis for co-expressed genes in real expression data is one of the most commonly used techniques for biological validation rather than statistical analysis. The best analytical decision can be made with the aid of biological knowledge, annotation database, resulting clusters and p-value acquired from statistical methods. In contrast, co-expressed genes in a cluster are expected to be enriched related to the biological role. More importantly, enrichment analysis is helpful to determine the over-representation of given input gene lists over the background set of genes. The background gene set is compiled from the GO database. The three categories of GO are BP, MF, and CC. A cluster is considered to be enriched if the p-values of all the annotation terms are less than the significance cut-off value. Moreover,

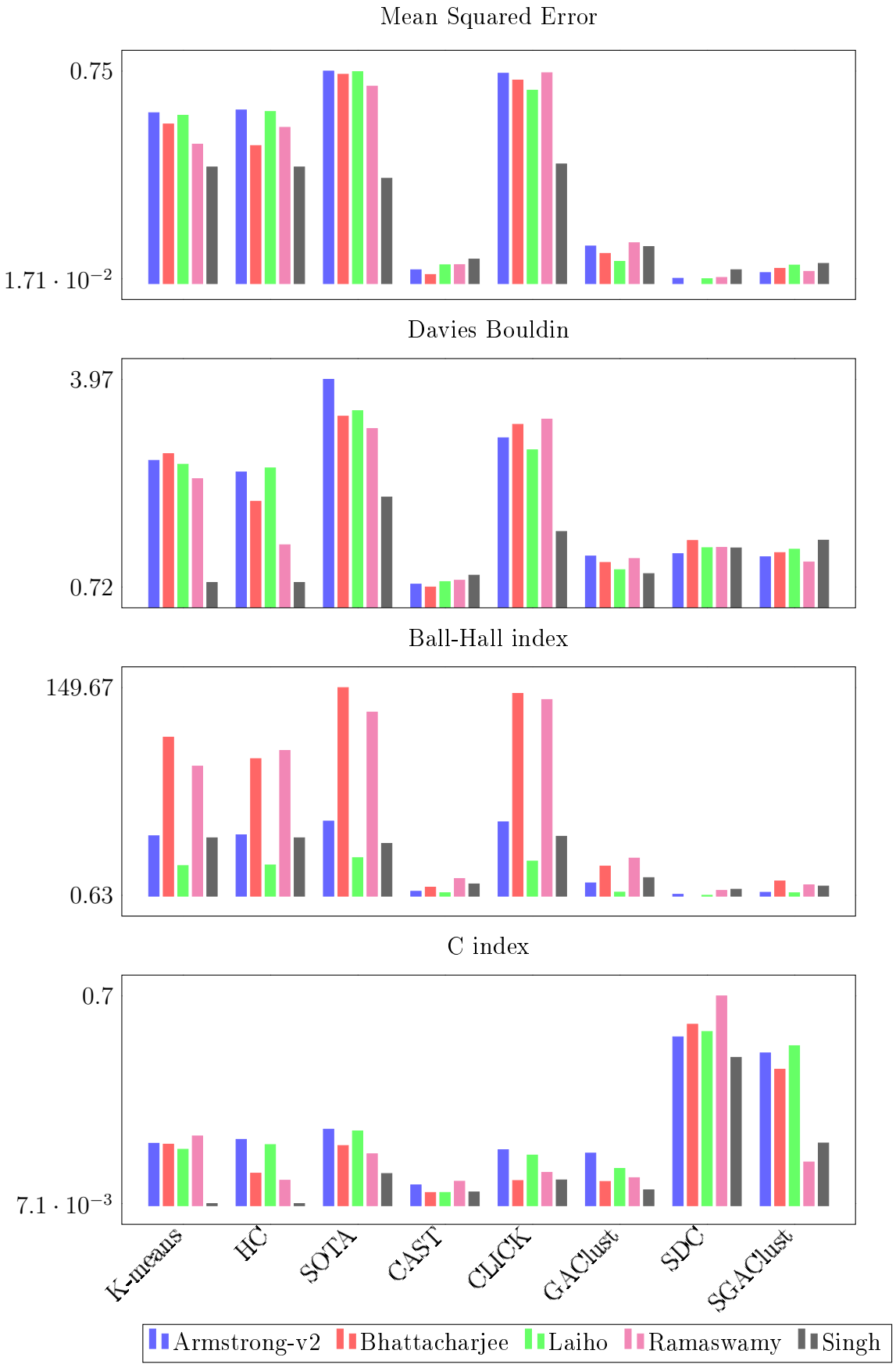


Figure 3.9: Different cluster validation indices for cancer datasets.

if one of the annotation terms is from any one of the GO categories, for instance, BP, it is said to be enriched.

Proposed algorithms discover several genes as outliers, now it is time to investigate whether reduction of genes really affects the enrichment analysis or not. Among the many available enrichment tools, we have used FuncAssociate [35] for calculating p-values. FuncAssociate uses Fisher's Exact Test to compute the hypergeometric functional score and adjusts the score for multiple testing using another method, named Westfall and Young procedure [35]. It is necessary to convert the gene list into Official IDs from Affymetrix id using web-based tool Database for Annotation, Visualization and Integrated Discovery (DAVID) [144]. The resulting clusters discovered from each of the methods are submitted into FuncAssociate 3.0 [35] one by one as a gene query list. Each of the methods is evaluated by their potentiality of identifying a total number of enriched GO terms with 5% significant cut-off for each dataset. Figure 3.10 depicts the number of significant GO terms for each method on each dataset. Considering all the datasets, we see different methods giving the enrichment result in different numbers of significant GO terms ranging from 3 to 1032.

Considering only unsupervised algorithms, we can observe that HC detects maximum numbers of enriched GO terms on the Bhattacharjee dataset and the K-means algorithm discovered the highest number of GO terms on the Singh dataset. It can also be noted that the CLICK algorithm outperforms among all unsupervised methods for the Amstrong-v2 dataset. Moreover, the CAST algorithm wins over Laiho and Ramaswamy datasets in this experiment. Although there is variation in the performance of different datasets for unsupervised algorithms, overall, taking all the five datasets together GAClust identifies 1553 GO terms which is definitely higher than any other unsupervised testing method. Based on unsupervised algorithms, SOTA is considered to be the worst performer by yielding 1229 numbers of GO terms. While comparing between semi-supervised and unsupervised methods it can be found that semi-supervised algorithm i.e., SGAClust outperforms all other methods including GAClust on every dataset whereas the SDC algorithm shows the worst performance in this case. SDC algorithm can only identify 384 significant GO terms which is much much lesser than GO terms identified by the SOTA algorithm.

Next, the central point of our discussion is p-values. For that, we summarize the lowest p-value corresponding to a GO term among all resulting clusters of each method on every dataset as given in Table 3.6. Lower p-values signify a better cluster. SGAClust outperforms all other clustering algorithms by giving

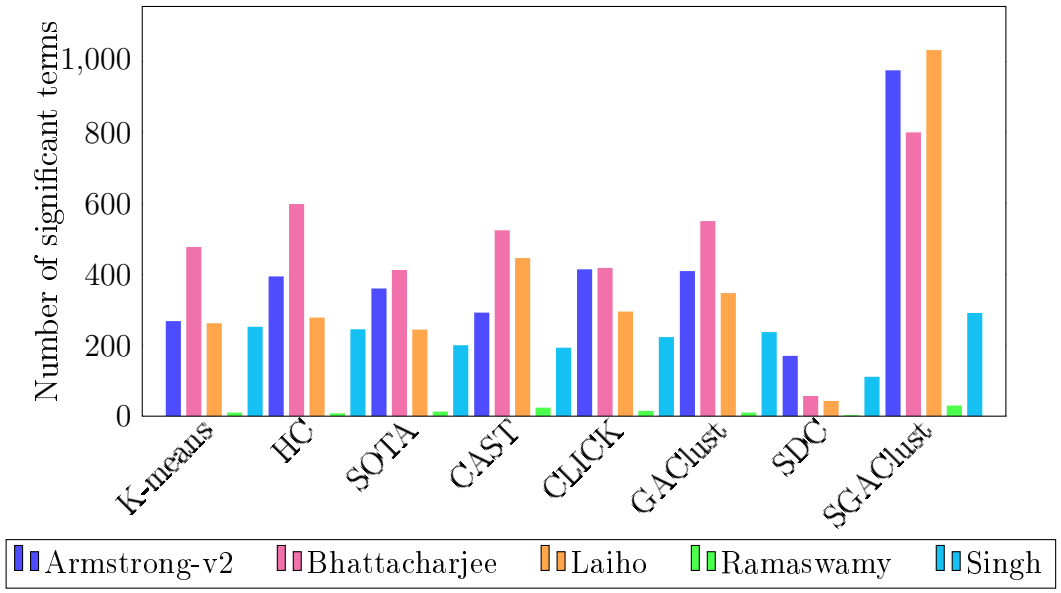


Figure 3.10: The number of enriched terms (shown in y-axis) by six different methods (shown in x-axis) for different datasets.

the lowest p-values for all datasets whereas SDC gives higher p-values for all datasets. Interestingly, SOTA yields the second-best p-value for two datasets, Bhattacharjee and Ramaswamy but it shows the worst performance in the previous experiment. GAClust also provides lower p-values for datasets Laiho and Singh among all unsupervised ones and second-best among all methods. CLICK algorithm also seems to achieve a lower p-value of 1.24E-39 for Laiho while comparing with unsupervised methods and is the second-best performer after SGAClust. Moreover, K-means algorithms give the second-best result in this regard for the Armstrong dataset. Surprisingly, with the reference to the previous discussion, we can see that the methods which provide a good number of GO terms, may not give the lowest p-value, for instance, the GAClust algorithm.

Table 3.6: Comparison of p-values among all datasets on various datasets.

Dataset	Algorithm	GO ID	GO Name	p-value
Armstrong-v2	K-means	GO:0000786	nucleosome	4.96E-24
	HC	GO:0043299	leukocyte degranulation	1.95E-23
	SOTA	GO:0031325	positive regulation of cellular metabolic process	1.05E-14
	CAST	GO:0000786	nucleosome	3.42E-21
	CLICK	GO:0007165	signal transduction	6.27E-23

Continuation of Table 3.6

Dataset	Algorithm	GO ID	GO Name	p-value
	GAClust	GO:0007165	signal transduction	2.39E-20
	SDC	GO:0032502	developmental process	8.18E-14
	SGAClust	GO:0007165	signal transduction	2.91E-91
Bhattacharjee	K-means	GO:0070268	cornification	1.59E-20
	HC	GO:0002376	immune system process	4.79E-22
	SOTA	GO:0002376	immune system process	1.29E-26
	CAST	GO:0000786	nucleosome	1.3E-16
	CLICK	GO:0030198	extracellular matrix organization	7.88E-22
	GAClust	GO:0070268	cornification	2.16E-19
	SDC	GO:0044421	extracellular region part	6.40E-08
	SGAClust	GO:0007165	signal transduction	2.52E-70
Laiho	K-means	GO:0031012	extracellular matrix	4.20E-32
	HC	GO:0031012	extracellular matrix	6.1E-33
	SOTA	GO:0031012	extracellular matrix	1.85E-37
	CAST	GO:0031012	extracellular matrix	2.49E-29
	CLICK	GO:0031012	extracellular matrix	1.24E-39
	GAClust	GO:0031012	extracellular matrix	4.14E-39
	SDC	GO:0032963	collagen metabolic process	1.69E-11
	SGAClust	GO:0065007	biological regulation	3.15E-72
Ramaswamy	K-means	GO:0002376	immune system process	6.48E-07
	HC	GO:0097458	neuron part	2.96E-07
	SOTA	GO:0043005	neuron projection	7.82E-09
	CAST	GO:0043005	neuron projection	5.89E-07
	CLICK	GO:0043005	neuron projection	4.69E-08
	GAClust	GO:0097458	neuron part	1.11E-08
	SDC	GO:0030425	dendrite	1.11E-06
	SGAClust	GO:0043005	neuron projection	2.14E-09

Continuation of Table 3.6

Dataset	Algorithm	GO ID	GO Name	p-value
Singh	K-means	GO:0006614	SRP-dependent co-translational protein targeting to membrane	1.8E-75
	HC	GO:0006614	SRP-dependent co-translational protein targeting to membrane	1.8E-75
	SOTA	GO:0006614	SRP-dependent co-translational protein targeting to membrane	2.34E-63
	CAST	GO:0006614	SRP-dependent co-translational protein targeting to membrane	7.48E-69
	CLICK	GO:0006614	SRP-dependent co-translational protein targeting to membrane	3.97E-67
	GAClust	GO:0006614	SRP-dependent co-translational protein targeting to membrane	1.8E-76
	SDC	GO:0006413	translational initiation	1.97E-15
	SGAClust	GO:0043005	neuron projection	5.88E-80

3.7 Potential biomarkers identification

In this section, we describe network-based biomarker identification techniques. The foundation of this technique is based on the clustering results. This technique uses two user-defined thresholds i.e., φ for the number of biomarkers and ψ for the number of clusters.

Network-based biomarker identification technique incorporates external

biological knowledge with clustering results. To compute the network-based biomarker for microarray data, first, we convert the Affymetrix gene id to the official gene symbol by DAVID 6.8[1] [144] for all clusters of each algorithm. To calculate the p-value of a cluster, the genes of the corresponding cluster is fed as input to the FuncAssociate [35] version 3.0 for gene microarrays considering 0.05 as a level of significance. The result obtained gives the p-value corresponding to GO IDs. This process is repeated for all the clusters. It is significant to mention here that the p-value may not be unique. Next, we sort clusters based on p-values we consider the ψ distinct p-valued clusters. If multiple clusters have the same p-values then all of them will be considered and the number of clusters in such case may be more than ψ. Next, we extract the network module(s) from each of the significant gene clusters for the prediction of functional interactions. To determine the biomarkers we find out the top (higher) φ degree unique genes from all the constructed networks. We set $\varphi = 2$ and $\psi = 10$. The reason for selecting the parametric values is given in Chapter 4.

For network construction we take the help of GeneMANIA Cytoscape version 3.3.0 plugin version 3.4.1 [247]. In addition to the genes present in a cluster, 20 additional related genes are taken into account to create an interaction network using an automatic weighing scheme and considering the source organism as *Homo Sapiens*. The relationship among the genes which are evaluated are co-expression, co-localization, genetic-interactions, pathway, physical-interaction, predicted, and shared protein domains (default parameter settings of GeneMANIA Cytoscape). The summary of network-based biomarker identification results for all the five different cancer datasets is given in Table 3.7. The identified gene biomarkers are validated through literature.

Among the potential biomarkers, Yu et al. [359] have reported that *APP* is highly expressed in AML. Therefore, clinically it has a significant impact on blood cancer. *SETBP1* is considered as oncogene and it defines the molecular characteristics of Leukemia [68]. The mutation of *SETBP1* gene is an important factor in cancer development. The gene *ILS1* plays an important key factor in many cancers including lung tumor [200]. Li et al. [202] have evaluated the function of gene *PBX1* in the proliferation of non-small-cell lung cancer. Lung cancer is a widely spread oncological disease. The study in [362], has reported that *COL1A2* is treated as tumor suppressor in a colorectal cancer cell and also provided a therapeutic approach to treat this disease. For colorectal cancer *FBN1* gene may be consider as a promising biomarker [203]. The gene *MAGI2*

[1] `https://david.ncifcrf.gov/`

Table 3.7: Potential biomarkers identification of different proposed full-space clustering algorithms using network-based method.

Algorithm	Dataset	Potential biomarkers
GAClust	Armstrong-v2	*APP*, *SETBP1*
	Bhattacharjee	*ISL1*, *PBX1*
	Laiho	*COL1A2*, *FBN1*
	Ramaswamy	*MAGI2*, *NFIA*
	Singh	*RPS23*, *RPS16*
SDC	Armstrong-v2	*MNDA*, *ELANE*
	Bhattacharjee	*EGFR*, *CPS1*
	Laiho	*FN1*, *PALLD*
	Ramaswamy	*MAGI2*, *TCF4*
	Singh	*RPS16*, *RPS5*
SGAClust	Armstrong-v2	*FN1*, *APP*
	Bhattacharjee	*ISL1*, *BMP2*
	Laiho	*FN1*, *COL1A2*
	Ramaswamy	*MAGI2*, *TCF4*
	Singh	*RPS23*, *RPS16*, *RPS3A*

is altered in 0.81% of all types of cancers such as lung, colon, and breast cancer[2]. The study [104] focuses on the vital role of gene *NFIA* in multiple cancer types. Ribosomal protein gene *RSP5* is a potential driver of cancer type [27].

Zhao et al. [367] have concluded that *ELANE* acts as an oncogene in Leukemia and it is a potential therapeutic target. Overexpression of the gene *EGFR* causes wide ranges of multiple cancers[3]. The study in [340] shows that the gene *CPS1* plays a vital role in the development and prognosis of lung cancer. The gene *FN1* is found to be dysregulated in multiple cancers such as colon cancer [198]. According to Cancer Genetics Web[4], *TCF4* plays a useful oncogene role in ovarian cancer. The gene *BMP2* is highly over-expressed in lung cancer tissue compared to normal tissue [23]. Hence, it enhances the growth of metastasis of tumor and develops cancer. In the study [267], it has been investigated that *RPS16* gene is useful on tumorigenesis and development of prostate cancer.

3.8 Discussion

In this chapter, we have proposed one unsupervised and two semi-supervised algorithms which we will discuss next. The proposed unsupervised GAClust is based on a graph-theoretic clustering algorithm. The main focus of the GAClust

[2] `https://www.mycancergenome.org`
[3] `http://www.cancerindex.org/`
[4] `www.cancer-genetics.org`

algorithm is to develop a user-defined parameter-less clustering and which makes the algorithm distinct from CAST. Moreover, the GAClust algorithm decides the threshold dynamically depending on the individual dataset whereas the CAST algorithm does not provide any guideline. Unlike CAST, GAClust obviates the need for a cleaning step due to this dynamic threshold as proposed in the original algorithm. The dynamic computation of thresholds gives very promising results on applying over different datasets. Further, theoretical analysis of determining threshold parameters may be one of the future research directions.

The performance of GAClust is compared with five state-of-the-art methods with respect to synthetic and cancer gene expression datasets using both internal and external measures as a validation criterion. Our algorithm is advantageous as it does not require the number of clusters a priori. We have also provided a guideline for the input parameters. The first striking conclusion we can draw is that no algorithm is superior throughout all the measures over all datasets synthetic as well as real. Indeed, in many cases, we have observed that one algorithm may give the best result for some metric and may also be worst considering another metric. From the study, we can say that GAClust is a well-suited algorithm in comparison with all other methods for synthetic datasets. In accordance with the real datasets, the GAClust algorithm outperforms all other comparing methods with respect to biological significance. Hence, GAClust is biologically more significant than other algorithms. Additionally, GAClust outperforms all algorithms except for CAST.

We have also proposed two semi-supervised algorithms SDC incorporating gene ontology in a density-based clustering algorithm and SGAClust which integrates GO with GAClust. The main advantage of both algorithms is that we do not have to give the number of clusters as an input. Among these two clusters, SGAClust outperforms all other competing methods. It is being observed that external domain knowledge gives reliable clusters. Additionally, all three proposed algorithms are equally effective in order to identifying potential cancer biomarkers. We have validated the found biomarkers from the literature.

We come to the conclusion that a semi-supervised algorithm provides significant clusters. Biologically it is proven that one gene may participate in many biological pathways, this allows it to belong to multiple clusters. Not only that, in the true sense subset of genes are active under certain conditions. To explore this concept, in the next chapter, we propose an order-preserving biclustering algorithm to identify biclusters that resembles the true biological scenario.

4

Bicluster Analysis of Cancer Transcriptomics Data

In Chapter 3, traditional clustering approaches have been successfully used to find global patterns from transcriptomics data. The inefficiency of classical methods in identifying local patterns has motivated us to shift towards biclustering algorithms. Chapter 2 presents an overview of biclustering, its type, structures, and lastly the evaluation procedures to be employed in the validation of obtained biclusters. In this chapter, we revisit biclustering and propose our own algorithm to detect biclusters from transcriptomics data. This chapter is structured as follows. Section 4.1 presents the introduction, Section 4.2 reviews related work in this domain, and Section 4.3 clearly states the motivation behind our proposed algorithm. The proposed method is described in Section 4.4 and the complexity is analyzed in Section 4.5. In Section 4.6, a collection of datasets including artificial and real cancer data (microarray and miRNA) are considered, to examine the overall performance of the proposed algorithm with other state-of-the-art methods. Next, the obtained biclusters are further explored by identifying potential biomarkers in Section 4.7. In this context, we present a frequency-based biomarker identification method. Finally, a discussion is provided in Section 4.8.

4.1 Introduction

Biclustering algorithms are based on the premise that a subset of genes participates in certain cellular processes active under some subsets of conditions [227]. The motivation behind the move from full-space clustering to biclustering algorithm is to investigate several genes which are responsible for diseases and which in turn expressed strong coherence under a subset of conditions. The need for biclustering algorithms can be understood by some clinical applications [56, 57]. It is possible that a patient can suffer from more than one disease so biclustering would be an appropriate technique to use in this scenario. It aids in finding the subset of biological factors which causes the disease along with the subset of genes [116]. Wang et al. [335] first applied biclustering algorithms to identify clinically significant modules in gene expression data to classify breast tumor.

An important biological fact states that co-regulated genes across a limited set of conditions are trend-preserving while expression values may be different in these experimental conditions [337]. An expression pattern (expression values of a row under certain conditions) is said to be trend-preserving when it is order-preserved. The study [337] claims that the biologically trend-preserving biclusters represent a generalized version of other types of biclusters which include constant, additive, multiplicative, and additive-multiplicative bicluster types. Among these, additive-multiplicative is considered the most challenging to identify. From studying the literature we understand that a particular biclustering algorithm work well for a particular bicluster type [266]. So, it has become of utmost importance to develop an efficient biclustering algorithm that can discover all types of bicluster models. Additionally, gene expression data has a huge amount of genes and samples. Therefore, we propose a parallel biclustering algorithm named **O**rder-**P**reserving **Bic**lustering (OPBic) which can discover biclusters with a maximum subset of rows under a maximum subset of conditions, which are order-preserved submatrices.

An order-preserved submatrix is a non-contiguous submatrix where expression values of each row under certain column permutations monotonically increases or decreases. The key advantage of the OPBic algorithm is that it does not require the number of biclusters as an input parameter. Our approach to solving the biclustering problem is based on a well-known string matching problem called Order-Preserving Pattern Matching (OPPM) [177]. OPPM states that a pattern P matches a substring of text say T, where the relative order completely matches P. In our work, we modify the algorithm by focusing on subsequences of T rather than substrings of T. Some definitions relevant to our algorithm is

given next.

Definition 4.1.1 *A substring of a given string is said to be a contiguous sequence of characters of a string.*

For example, if "gene" is a string, then the list of substrings of string "gene" are "" (empty string), "g", "e", "n", "ge", "en", "ne", "gen", "ene", and "gene".

Definition 4.1.2 *A subsequence of a given sequence is extracted by removing some of the elements or no elements without changing the original order of the sequence.*

Subsequence is a generalization of substring. Let us again consider the same word "gene" to demonstrate the concept of subsequence. In this case, the subsequences are "" (empty string), "g", "ge", "gn", "e", "en", "ee", "n", "ne", "e", "gen", "gee", "gne", and "ene".

The performance of the OPBic algorithm is tested using synthetic as well as real datasets. Our actual intention is to develop an algorithm for seeking smaller condition set biclusters which most of the algorithms ignore. The performance of the OPBic algorithm is compared with C&C, BicSPAM, BiBit, and UniBic on synthetic and real datasets. A significant advantage of our method to others except UniBic is to discover several types of biclusters. UniBic algorithm misses some small condition-specific biclusters whereas our algorithm considers that too.

Our study aims to examine the cancer gene and miRNA breast cancer expression profiles with the help of the proposed biclustering algorithm, OPBic. The major contributions of this work are:

- A order-preserving biclustering algorithm.
- A synthetic data generator to evaluate biclustering algorithms in an unbiased manner.
- Validation of the proposed OPBic algorithm using both synthetic and real data.
- Identification of some interesting potential biomarkers.

4.2 Related work

In this chapter, we focus on finding order-preserving submatrices. Ben-Dor et al. [32] have developed an effective OPSM algorithm to discover large order-preserving biclusters from gene expression data that exhibit similar trends over

some subsets of conditions, using a statistical approach. Prelic et al. [278] show that the OPSM is more capable of capturing biologically significant patterns compared to other competing methods. OPSM mining problem is gaining a lot of attention in this area of research [65, 67, 106, 107, 113]. However, in reality, the gene expression data is noisy by nature, and therefore the OPSM algorithm faced difficulty in locating identical trends. To address this issue, many noise-tolerant variants of the OPSM algorithm have been developed [105, 106, 363].

Xue et al. [348] have developed an exact algorithm based on frequent sequential pattern mining to identify all deep OPSMs by disclosing all common subsequences (ACS) between all pairs of rows. Cheung et al. [65] have extended the idea of OPSM by converting the problem into sequential pattern mining to enable the discovery of coherent patterns that exhibit rise and fall over subspaces. Jhang et al. [363] have devised a noise-tolerant algorithm named AOPC (Approximate Order-Preserving Cluster) which requires a pre-specified fraction of rows to induce identical attribute order instead of finding the same linear order of columns. Further relaxation of AOPC is considered in ROPSM (Relaxed Order-Preserving Submatrix) [105] which allows all the rows of a bicluster to have similar backbone order.

A robust OPSM method called OPSM-RM with repeated measurements [67] conducted multiple experiments to handle noise. The Bucket OPSM (BOPSM) model has discovered patterns by exploiting significant associations among rows and columns inducing linearity relaxation [106]. A generalized version of AOPC, ROPSM, and BOPSM are GeBOPSM which allows the rows to be induced with different integers than in OPSM [106]. The Probabilistic OPSM (POPSM) model aim at capturing similar local correlations among rows under columns from probabilistic matrices with data uncertainty [107]. Gao et al. [113] have proposed another sequential pattern mining approach referred to as deep OPSM, corresponding to long patterns with few supporting sequences. The KiWi framework uses two parameters i.e., k and w corresponding to bound for search space and computational resources for identifying deep OPSMs.

Recently, Pio G. et al. in [69] and [274] have proposed distributed clustering and classification approach for microarray gene expression data and significantly achieved better performance than the traditional approach.

4.3 Motivation

A wide variety of biclustering algorithms have been proposed in the context of biological data to uncover genetic relationships. Among all biclustering algorithms, pattern-based biclustering algorithms are widely used in analyzing gene expression data based on pattern similarity rather than distance similarity [348]. While comparing genes under different experimental conditions, it is assumed that relative expression levels of genes are more meaningful than absolute expression values [348]. Therefore, we concentrate on pattern-based subspace clustering to solve the biclustering problem.

Designing a new biclustering algorithm is actually a very challenging task. In the literature, more than fifty biclustering algorithms are present. Most of the algorithms are capable of identifying only one or two types of biclusters and ignore the rest. UniBic is capable of identifying six major types of biclusters, i.e., column-constant, row-constant, multiplicative, additive, additive-multiplicative, and trend-preserving. UniBic algorithm focuses on the local similarities therefore it loses global reference. It is able to identify the longest order-preserving biclusters and is not able to capture the narrow biclusters which have a lower number of rows with respect to a large number of samples. UniBic misses some small condition-specific biclusters whereas our algorithm considers that too.

A limited effort has been made to develop a parallel biclustering algorithms in [40] and [209] which improve the running time of biclustering algorithms. Both the algorithms are based on some scores therefore it does not employ the goodness of pattern-based biclustering. Through our proposed method, we tried to investigate almost all possible combinations for bicluster identification. It considers the important features of pattern-based subspace clustering and parallel computing for developing biclustering algorithms.

4.4 Proposed method

We aim to find all coherent biclusters which have high biological significance. In this section, we describe in detail the parallel OPBic algorithm, which is capable of identifying all types of bicluster models. The fundamental idea behind the algorithm is presented in Figure 4.1.

It takes four input parameters: the minimum number of rows R_{min}, the minimum number of conditions C_{min}, the number of workers W, and the maximum overlap allowed O_{max}. The algorithm starts by transforming the input matrix into order matrix which is horizontally partitioned equally and distributed

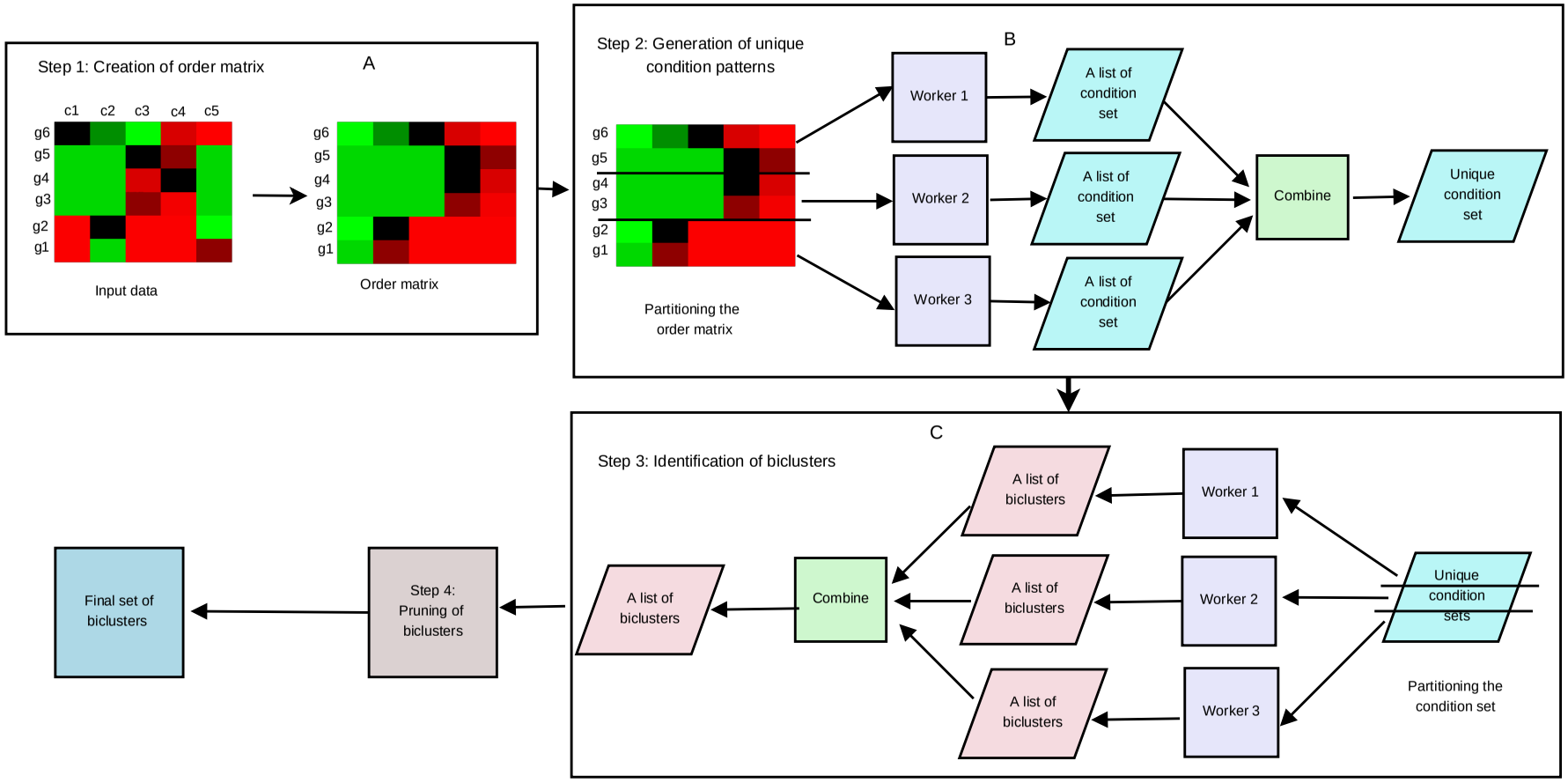


Figure 4.1: The overall workflow of the OPBic algorithm. The algorithm performs in four steps. (A) In the upper panel, we have an input data matrix where red and green color show the over and under expression values, respectively. The expression matrix is transformed into order matrix. (B) The order matrix is partitioned and distributed over W workers to generate the list of unique condition patterns. The length of condition pattern must be $\geq C_{min}$. (C) The final condition pattern is partitioned and given to the W workers to get the list of biclusters. A bicluster should have a minimum of R_{min} rows. Lastly, the biclusters which have higher overlaps (i.e., $O_{max} > 25\%$) are removed.

among multiple workers. Each worker computes the list of condition patterns by applying a modified version of the All Substrings Common Subsequence (ALCS) algorithm [9]. All generated unique condition patterns are stored in the master. Later, equal amounts of condition patterns are sent to multiple workers for parallel execution. Each worker produces a list of biclusters and returns the solution to the master to get a final list. Prior to finalizing the biclusters, we remove candidate biclusters that have high overlaps (i.e., $O_{max} > 25\%$). The value of O_{max} is assumed to be 25% as in [98, 266, 278]. The advantage of this algorithm is that it does not require the number of biclusters a priori.

4.4.1 Creation of order matrix

Definition 4.4.1 *Let,* $A = \{a_1, a_2, \ldots, a_n\}$ *and* $B = \{b_1, b_2, \ldots, b_n\} = asc\_sort(A)$ *be the two expression patterns. Consider a column index* $I = \{1, 2, \ldots, n\}$*. We define the order of an expression value* a_j *as* $\mathbb{O}(a_j) = \{r | r \in I\}$*, which satisfies* $a_j = b_r$ *and* $j = \{1, 2, \ldots, n\}$*. So the order of an expression pattern can be described as* $\mathbb{O}(A) = \{\mathbb{O}(a_1), \mathbb{O}(a_2), \ldots, \mathbb{O}(a_n)\}$*.*

Essentially, an order of an expression pattern is the permutation of the columns; in other words, it induces a linear re-ordering of the conditions based on the ascending ordered expression values. It represents the index of sorted values for an input pattern. If two values are identical, then ties are broken by assigning the lower value to the lower column index. An order matrix OM is obtained from the expression data ED for each pattern mentioned above.

To illustrate the concept of order, consider the expression matrix with three rows g_1, g_2, and g_3 over six conditions c_1, c_2, c_3, c_4, c_5, and c_6 demonstrated in Figure 4.2. The order of each row is $\mathbb{O}(g_1) = \{6, 3, 1, 2, 5, 4\}$ and $\mathbb{O}(g_2) = \{3, 1, 2, 6, 5, 4\}$. In case of tie, e.g. consider the expression values of g_3 where both c_1 and c_2 have same $(= -1)$ value, we apply the rule that smaller column index c_1 will get the higher order than c_2. Therefore, the order of g_3 is $\mathbb{O}(g_3) = \{3, 1, 2, 6, 5, 4\}$. The corresponding order matrix is also shown in 4.2.

4.4.2 Generation of unique condition patterns

Partitioning the order matrix: We divide OM horizontally into W number of equal partitions depending on available workers, where the last partition always gets either equal or slightly fewer instances. Each of the partitions is distributed across W workers for further computation.

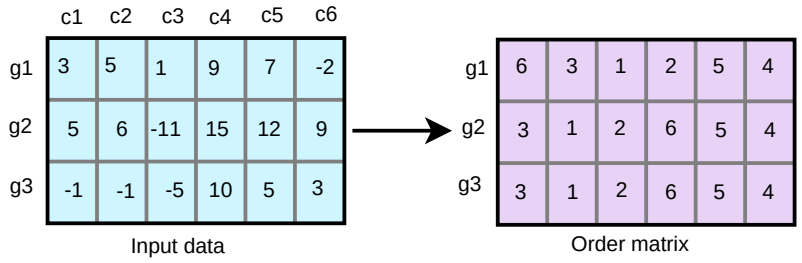


Figure 4.2: Illustration of an order matrix from input data. The input data consists of three rows and six columns where each entry of matrix shows the real expression values. The input data is transformed into an order matrix.

Applying mALCS: We apply mALCS (modified All Substrings Common Subsequence) [9] between every pair of rows for each worker to get the significant condition patterns.

Definition 4.4.2 *Given two sequences of patterns,* $P = \{p_1, p_2, \ldots, p_n\}$ *and* $Q = \{q_1, q_2, \ldots, q_n\}$*, the ALCS problem produces the length of longest string for every pair of* P *and substring* Q^s *of* Q *that is a subsequence of both* P *and* Q^s *[9].*

The LCS algorithm is used to find the Longest Common Subsequence between P and Q. ALCS is a generalization of LCS. We use mALCS, which identifies the condition patterns by LCS between every pair of P and restricted length substring Q_r^s of Q, where $|s| = n - l + 1, C_{min} \leq l \leq |Q|$ i.e., the minimum and maximum lengths of the substrings are C_{min} and $|Q|$, respectively. We consider the sequence rather than the length. We use the unique condition patterns for further computation. Every worker generates a set of condition patterns for the partition received by them.

Figure 4.3 illustrates the generation of unique condition patterns in details. Here, we consider two patterns, say $g_1 = \{6, 3, 1, 2, 5, 4\}$ and $g_2 = \{3, 1, 2, 6, 5, 4\}$. These two patterns are permutation of columns or conditions. Next, we consider the substrings of g_2 which is restricted to the minimum length as $C_{min} = 3$. Hence, we get substrings $\{3, 1, 2\}$, $\{1, 2, 6\}$, $\{2, 6, 5\}$, $\{6, 5, 4\}$, $\{3, 1, 2, 6\}$, $\{1, 2, 6, 5\}$, $\{2, 6, 5, 4\}$, $\{3, 1, 2, 6, 5\}$, $\{1, 2, 6, 5, 4\}$, and $\{3, 1, 2, 6, 5, 4\}$. Next, we identify the LCS between g_1 and all the substrings as shown in Figure 4.3-C. Now, we remove the obtained results whose length is $< C_{min}$. Thus we have $\{3, 1, 2\}$, $\{6, 5, 4\}$, $\{3, 1, 2\}$, $\{1, 2, 5\}$, $\{2, 5, 4\}$, $\{3, 1, 2, 5\}$, $\{1, 2, 5, 4\}$, and $\{3, 1, 2, 5, 4\}$ results. After that duplicates are removed to get distinct strings.

Final condition patterns: We obtain all solutions from the workers and remove duplicates to obtain the final condition patterns.

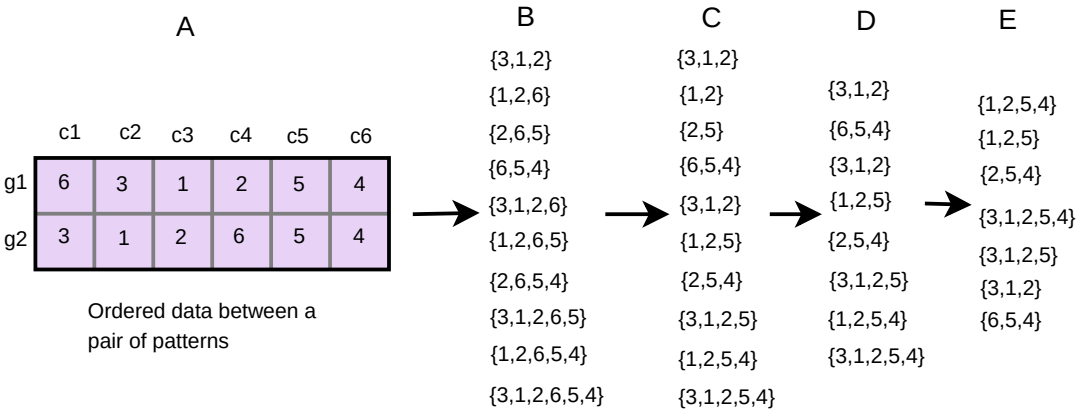


Figure 4.3: Identifying condition patterns. (A) It is an ordered data between g_1 and g_2. (B) Substrings of g_2 considering the minimum length as $C_{min} = 3$. (C) Result of LCS between g_1 and the substrings depicted in B. (D) Removing the strings which have the length $< C_{min}$. (E) Condition patterns after removing the duplicates.

4.4.3 Identification of biclusters

Partition condition patterns: This step is initiated by partitioning the final condition patterns equally and sending the partitions to the available workers to identify biclusters. The last worker gets a smaller or equal amount of data than the other workers.

Definition 4.4.3 *Two expression patterns,* $A = \{a_1, a_2, \ldots, a_n\}$ *and* $B = \{b_1, b_2, \ldots, b_n\}$ *are said to be order-preserved under subset of conditions,* $J = \{1, 2, 3, \ldots, y\}$*, if they satisfy the relation* $\mathbb{O}(A) = \mathbb{O}(B)$*, i.e., the order of each of the components of two patterns are equal.*

Formation of a bicluster: The core task of our algorithm is to identify an order-preserving submatrix that is strictly monotonically increasing using an OPPM algorithm on the basis of condition pattern. This problem is solved by locating a fragment of text which is order-isomorphic to the pattern. Let the two sequences X and pat of length n be order-isomorphic, and $pat[a] \leq pat[b]$ if and only if $X[a] \leq X[b]$ for all $a, b = \{1, 2, \ldots, n\}$ [182]. Next, we take a closer look into the definition 4.4.4 to understand the concept.

Definition 4.4.4 *Suppose, we have two sequences (orders)* $X = (x_1, x_2, \ldots, x_n)$ *and* $pat = (y_1, y_2, \ldots, y_{pt})$ *where* $|X| \geq |pat|$*. Given an integer* w *such that* $w = |pat|$*, an exact match consists of finding a sequence of size* w *from* X *which includes pattern pat as a subsequence i.e., the values* $y_1, y_2, \ldots, y_{pt}$ *occur in* X*, in the same order as in pat, but not necessarily consecutive (they might be interleaved*

with other values in the pattern).

For each worker, we take each condition pattern to form a single bicluster $\beta_{\mathcal{I}\times\mathcal{J}}$. We keep on adding OPPM rows for a condition set $\mathcal{J}$ where the number of rows $\geq R_{min}$, i.e., $|\mathcal{I}| \geq R_{min}$. In Figure 4.4, rows $\{g_1, g_2, g_3\}$ are order-preserved over three conditions $\{c_1, c_2, c_3\}$ as all the patterns follow same sequence. Therefore, $\beta_1 = (\{g_1, g_2, g_3\}, \{c_1, c_2, c_3\})$ forms a bicluster. The details of all other biclusters are given in Figure 4.4.

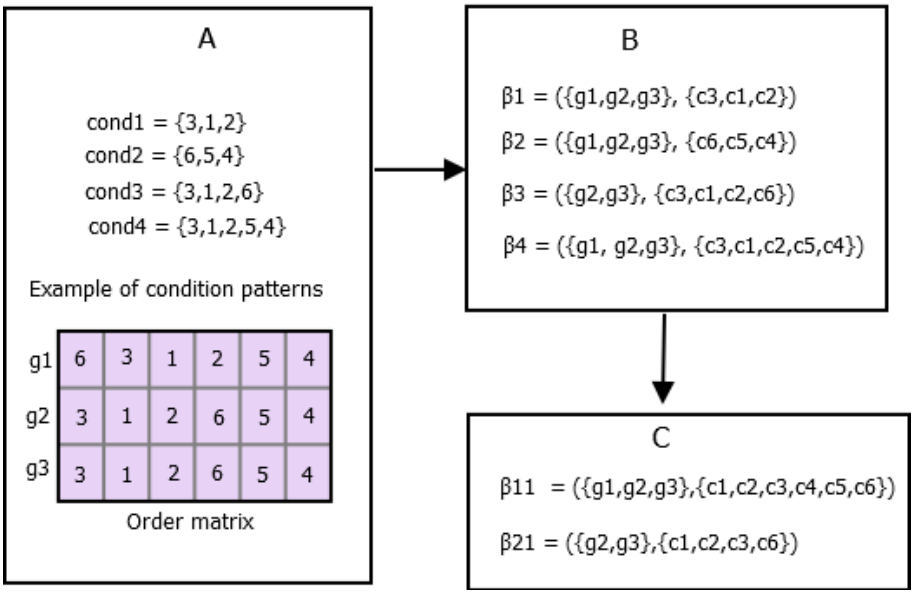


Figure 4.4: Identifying condition patterns. (A) The input parameters for bicluster identification are condition patterns, order matrix, and R_{min}. (B) Based on each condition pattern we identify the biclusters $\beta_1, \beta_2, \beta_3$, and β_4. Minimum number of rows are present in a single bicluster. (C) Merging of two biclusters are done depending on unique row sets. Here, we merge β_1, β_2, and β_4 to form a merged bicluster β_{11} and β_{21} represents the previous β_3 bicluster.

Merging of biclusters: We merge two biclusters if they have the same sets of rows. We add columns to a bicluster. Hence, we get approximately order-preserved biclusters. From Figure 4.4, we clearly observe that biclusters β_1, β_2, and β_4 have same set of rows. Therefore we merge them to form a bicluster say, $\beta_{11} = (\{g_1, g_2, g_3\}, \{c_1, c_2, c_3, c_4, c_5, c_6\})$.

Final set of biclusters: Biclusters identified by all the workers in parallel, are combined into a single list of biclusters. We follow the same procedure as above to merge the biclusters.

4.4.4 Pruning of biclusters

To get the final list of biclusters we remove those biclusters where the overlap between two biclusters is more than a user-defined threshold, i.e., O_{max}. To remove biclusters, we first determine the size of each bicluster i.e., $|\mathcal{I}| \times |\mathcal{J}|$ and calculate the overlap score between two biclusters, say β_1 and β_2 using Equation 4.4.1. We keep the large biclusters and remove the smaller biclusters which have large overlaps ($> O_{max}$).

$$O = \frac{|\beta_1 \cap \beta_2|}{min(|\beta_1|, |\beta_2|)} \tag{4.4.1}$$

4.5 Time complexity

The biclustering problem is intractable, therefore it is highly challenging to develop efficient and effective algorithms [337]. Biclustering problem is an NP-hard problem. We compute the algorithmic complexity in order to evaluate the efficiency of our algorithm. Let, $ED_{m \times n}$ be the expression matrix of m number of genes and n number of samples, W be the number of workers, and $\mathcal{S}$ be the number of substrings. OPBic takes $O(m(n \log n))$ time to compute order matrix. The fundamental step of OPBic is to find mALCS which essentially takes $O(\frac{m}{2}^{\frac{m}{W}} n^2 \mathcal{S})$ time. The number of substrings is dependent on $n - l + 1$ where the value of l ranges from C_{min} to n. The biclusters are generated from each of the condition patterns. In the next step we combine the result of each of the workers and find the unique conditions patterns. This step takes $O(W_2^{\frac{m}{W}} \mathcal{S})$ time. The cost of identifying bicluster identification task is $O(\frac{m}{2}^{\frac{m}{W}} \mathcal{S} m(n + n \log n))$. Combining all the biclusters from different workers and to eliminate the duplicate biclusters the algorithm takes $O(W_2^{\frac{m}{W}} \bar{r}\bar{s})$ time and $\bar{r}\bar{s}$ is the average size of the biclusters. Finally, the removal step takes $O((W_2^{\frac{m}{W}} \bar{r}\bar{s})^2)$ time. The overall worst case time complexity is $O(m(n \log n) + \frac{m}{2}^{\frac{m}{W}} n^2 \mathcal{S} + W_2^{\frac{m}{W}} \mathcal{S} + \frac{m}{2}^{\frac{m}{W}} \mathcal{S} m(n + n \log n) + W_2^{\frac{m}{W}} \bar{r}\bar{s} + (W_2^{\frac{m}{W}} \bar{r}\bar{s})^2)$. The resulting time complexity of OPBic approximately is $O(\frac{m}{2}^{\frac{m}{W}} (n^2 \mathcal{S} + \mathcal{S} mn + \frac{m}{2}^{\frac{m}{W}} (W \bar{r}\bar{s})^2))$.

To compare our algorithm, we use different biclustering algorithms and estimate the individual time complexity. Let, β be the total number of all inclusion-maximal biclusters in the input matrix, q be the separation percentage parameter, K the number of biclusters, $\bar{r}\bar{s}$ the average size of the biclusters, and $\wp$ the time to compute sequential pattern mining task. Then the time complexity of C&C, BiBit, BicSPAM, and UniBic algorithms are $O(mn)$, $O(m^3\beta)$, $\theta(min(\binom{m}{2}, n)\wp + \binom{K}{\frac{K}{2}}\bar{r}\bar{s})$, and $O(q^2 m^2 n^2)$. OPBic has high computational complexity which is one of the reasons that we have tried it with parallel computing. After using parallelism, it has been found that OPBic gives good quality biclus-

ters, which is our goal, even at the expense of computational cost.

4.6 Performance analysis

This section presents the performance of OPBic and shows its effectiveness compared to other state-of-the-art methods. OPBic is implemented in MATLAB 2016 on an Intel processor with 64 GB RAM. To assess the performance of the proposed contribution, we perform empirical assessment systematically. We analyze a total of 360 artificial datasets and three real datasets. The biological relevance of the biclusters identified by OPBic from miRNA expression datasets is assessed at the end of this section. For comparison, we take four recent and popular biclustering tools, C&C [64], BiBit [284], BicSPAM [134], and UniBic [337]. The reason behind selecting BicSPAM and UniBic is that all these algorithms aim to identify order-preserved submatrices. C&C is the pioneering biclustering algorithm and BiBit [284] performs well for up-regulated cases. For our experiment, we use the codes available as R packages, biclust [170] and BiBitR [78] for C&C (synthetic data) and BiBit algorithms, respectively. We also use Java-based tool available in Biclustering Analysis Toolbox (BicAT) Plus [5] for C&C (only for real data), Biclustering based on PAttern Mining Software (BicPAMS) for BicSPAM [132] and C implementation for UniBic [337]. Table 4.1, summarizes the codes and parameters used for the different biclustering algorithms.

4.6.1 Synthetic datasets generation

In general, a biclustering algorithm is developed either to identify a particular type of biclustering model or to make it perform better on various biclustering models. Therefore, it is essential to do a fair comparison by considering all bicluster types. In this study, we use three testing scenarios when generating synthetic data. They are (a) scenario 1 (single implanted bicluster [98]), (b) scenario 2 (variable sized background matrices with variable number of variable sized implanted biclusters [133]), and (c) scenario 3 (overlapped biclusters). Eight different biclustering models as given in Chapter 2, namely constant, row-constant, column-constant, constant, up-regulated, additive, multiplicative, additive-multiplicative, and trend-preserving are used to select the best suited model for a particular biclustering algorithm and to avoid misleading results.

At first, the background entries of synthetic datasets are chosen randomly from normal distribution $N(0, 1)$ (μ 0 and σ 1) and the values of smaller sized implanted submatrices are modified according to the eight different biclus-

Table 4.1: The meaning of parameters for different biclustering algorithms.

Method	Implementation used	Parameters	Meaning	Year
C&C [64]	R [170]	δ	Maximum mean square residue	2000
		α	A threshold for multiple node deletion	
		K	No. of biclusters	
BiBit [284]	R [78]	R_{min}	Minimum no. of rows	2011
		C_{min}	Minimum no. of columns	
BicSPAM [134]	BicPAMS [132]		Coherency assumption	2014
			Coherency strength	
			Quality	
			Normalization	
			Discretization	
			Noise handler	
			Symmetries	
			Missing handler	
			Remove elements	
			Stopping criteria (No. of biclusters before merging)	
			Minimum number of columns	
			Number of iterations	
			Coherency orientation	
			Pattern representation	
			Pattern miner	
			Scalability enhancer	
			Merging procedure	
			Filtering procedure (Dissimilar elements)	
UniBic [337]	C [337]	k	Minimum column width of the block	2016
		f	Filtering overlapping blocks	
		c	Consistency level	
		K	No. of biclusters	
		q	Quantile discretization	
		r	The number of ranks	

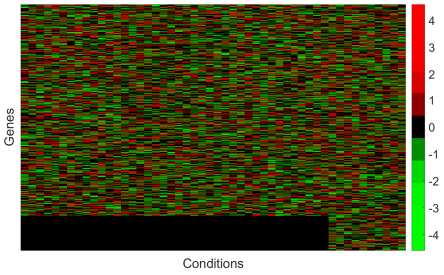
tering types. The expression values of constant and up-regulated biclusters are modified with 0 and 5, respectively [266]. A row or column-constant bicluster is generated by randomly selecting the base row or column, which is replicated in other rows and columns within the hidden bicluster. Each row of an additive-multiplicative bicluster model is modified by adding randomly generated additive and multiplicative factors using the normal distribution $N(0, 1)$ with a randomly selected base row within known biclusters. On the other hand, additive biclusters

are created in the same way, as mentioned in the previous model with scaling factor 1. Multiplicative biclusters are also generated in the same way, as mentioned in the additive-multiplicative model with additive factors 0. Biologically, trend-preserving is considered to be the most important type of bicluster, which is produced by selecting a random base row from the hidden bicluster and rearranging the other rows according to the same order of base rows within the submatrix [337].

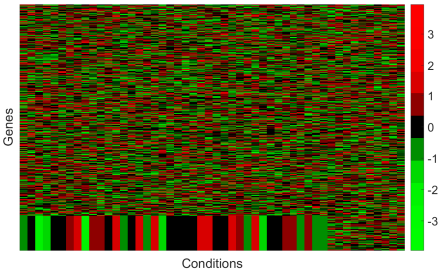
Usually, it is often observed from the original gene expression dataset that the number of genes is more than the number of conditions. This characteristic of expression data is reflected in our artificial datasets. For the first testing scenario, we consider the background matrix of size 500 rows and 50 columns with one implanted bicluster of size 70×40 which occupies 11.2% area of background matrix with no noise. For each bicluster model, we generate 10 instances of data matrices (in total 80) and results are summarized as an average across 10 matrices for the selection of the best-suited biclustering model for each algorithm. The heatmap of eight different biclustering models for scenario one is depicted in Figure 4.5 showing only one instance.

In the second scenario, the matrix size is varied from 100×30 to 1000×70. The number of implanted biclusters are also varied from 2 to 5 with the variation of hidden bicluster sizes without noise and overlap. An overview of the datasets is given in Table 4.2. For each background matrix, biclusters are implanted into it. There are lower and upper bound for both rows (referred to as R_L and R_U) and columns (referred to as C_L and C_U). Given a background matrix, suppose there are K implanted biclusters then the biclusters rows increment as $R_L + (i-1) * 5$ and columns increment as $C_L + (i-1) * 1$ where $i = \{1, 2, \ldots, K\}$. For example, for the second column of Table 4.2, there are 2 implanted biclusters of size (i) 10 rows 9 columns and (ii) 15 rows 10 columns. For the third column there are four implanted biclusters of size (i) 15 rows 10 columns, (ii) 20 rows 11 columns, (iii) 25 rows 12 columns, and (iv) 30 rows 13 columns. Therefore, we see the number of rows increase by 5 and columns increase by 1. Similarly, we find the five implanted biclusters of column fourth of the Table 4.2. The area covered by the biclusters is decreased by increasing background matrix size, as reported in [134]. For each setting, the data generation is repeated 10 times for each of the eight models (total 240) and results are presented as the average performance across these matrices. The heatmap for scenario 2 of one instance is presented in Figure 4.6.

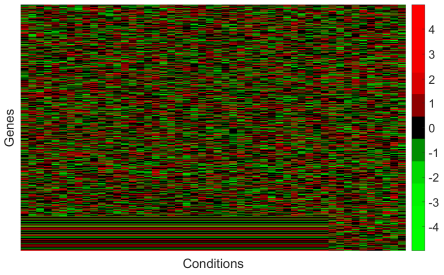
We also perform the experiment to check the capability of the biclus-



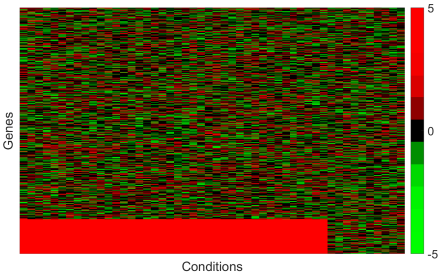
(a) Constant bicluster



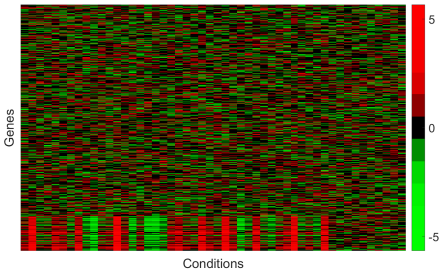
(b) Column-constant bicluster



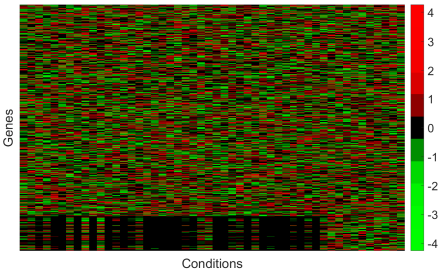
(c) Row-constant bicluster



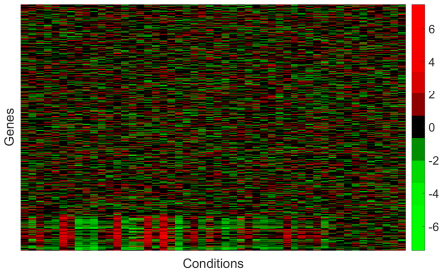
(d) Up-regulated bicluster



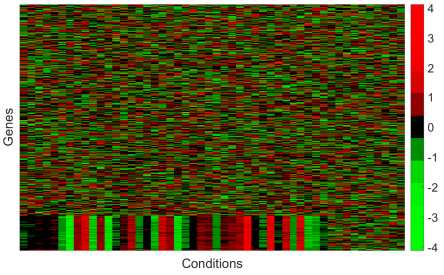
(e) Additive bicluster



(f) Multiplicative bicluster

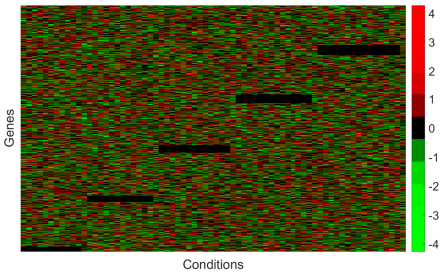


(g) Additive-multiplicative bicluster

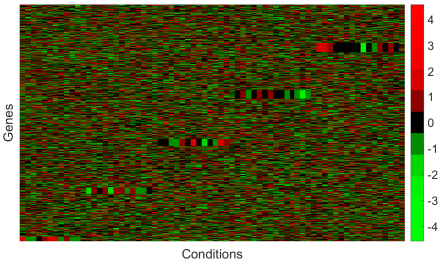


(h) Trend-preserving bicluster

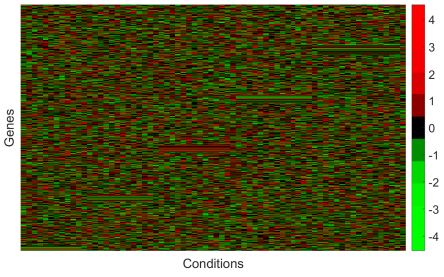
Figure 4.5: Heatmap of eight different data matrices for scenario 1.



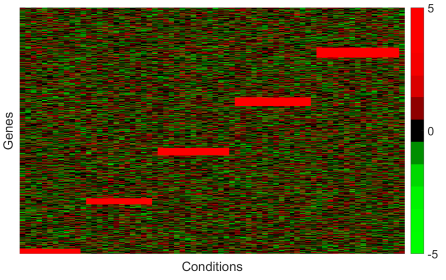
(a) Constant biclusters



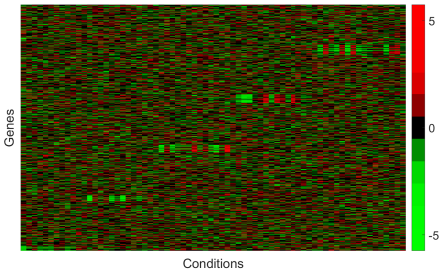
(b) Column-constant biclusters



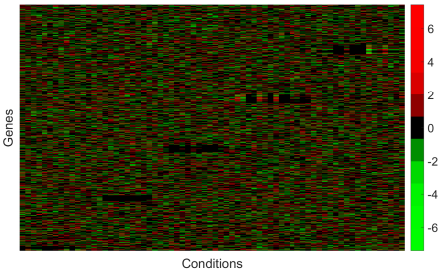
(c) Row-constant biclusters



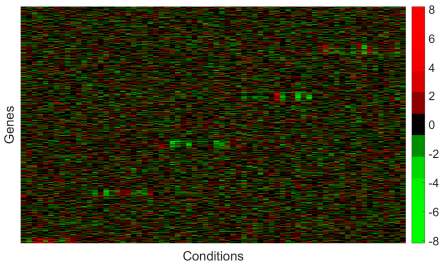
(d) Up-regulated biclusters



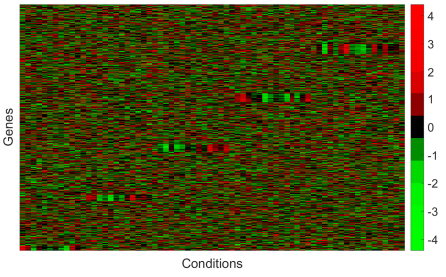
(e) Additive biclusters



(f) Multiplicative biclusters



(g) Additive-multiplicative biclusters



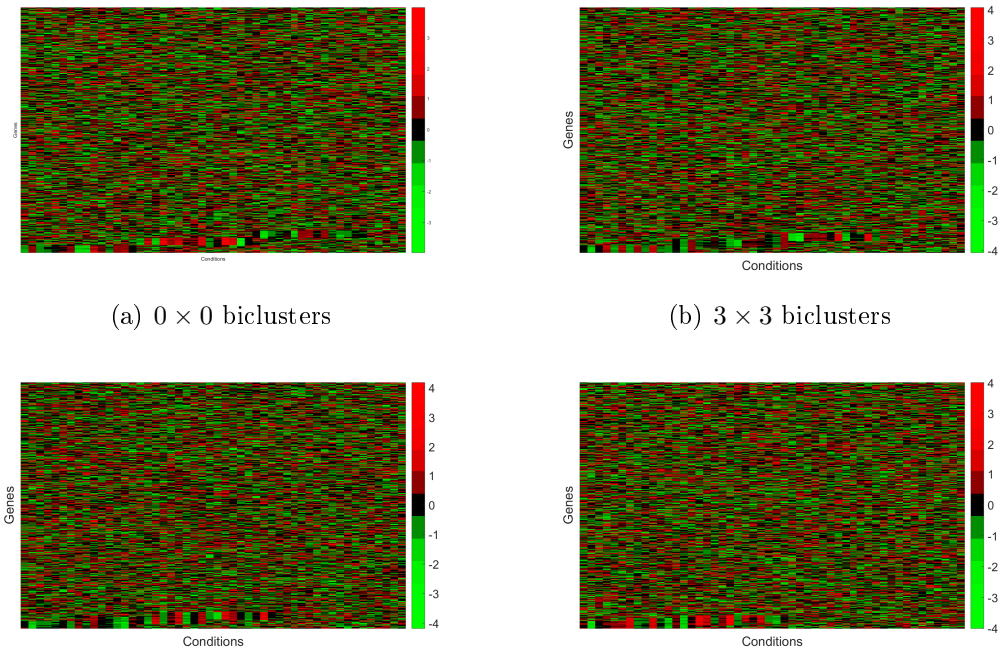
(h) Trend-preserving biclusters

Figure 4.6: Heatmap of eight different data matrices for scenario 2.

Table 4.2: A brief description of generated datasets. The first row denotes the size of the background matrix in terms of the number of rows and the number of columns. The second row says the number of implanted biclusters in the data matrix. The third and fourth rows present the bicluster size in terms of rows (R) and columns (C), respectively, which is specified by the lower (L) and upper (U) range.

Background matrix size	100 ×30	500 × 50	1000× 70
Implanted biclusters	2	4	5
Bicluster rows $[R_L, R_U]$	[10,15]	[15,30]	[20,40]
Bicluster columns $[C_L, C_U]$	[9,10]	[10,13]	[11,15]
Total biclusters area %	8	4.24	2.86

tering algorithm to discover overlapping biclusters in the third scenario. For this purpose, we take the background matrix of size 500×50 where we plant three trend-preserving biclusters of size 15×15 each because biologically trend-preserving is the most significant pattern [337]. The overlapping degree is 0×0, 3×3, 6×6, and 9×9. We repeat these 4 overlapping experiments 10 times to get 40 data matrices. The heatmap for scenario 3 of one instance is presented in Figure 4.7.



(a) 0×0 biclusters

(b) 3×3 biclusters

(c) 6×6 biclusters

(d) 9×9 biclusters

Figure 4.7: Heatmap of eight different data matrices for scenario 3.

Parameter settings for different biclustering algorithms

The success of any algorithm is highly dependent on appropriate parameter selection. Parameters are set as recommended by the authors, wherever possible. For a fair comparison, we use parameters such as the number of biclusters, the minimum number of rows and the minimum number of columns by providing the true values as per synthetic dataset creation [266]. The parameter settings for different synthetic datasets is shown in Tables 4.3, 4.4, and 4.5 for scenario 1, scenario 2, and overlapping experiments, respectively.

Table 4.3: Parameter values of different biclustering algorithms on scenario 1.

Method	Parameters	Scenario 1 (500×50)
C&C	K	1
BiBit	R_{min}	70
	C_{min}	40
BicSPAM	Coherency strength	7-9, 11, 14-17, 19, 20
	Minimum number of columns	7
	Filtering procedure	75%
UniBic	K	1

Table 4.4: Parameter values of different biclustering algorithms on scenario 2.

Method	Parameters	Scenario 2		
		200×30	500×50	1000×70
C&C	K	2	4	5
BiBit	R_{min}	10	15	20
	C_{min}	9	10	11
BicSPAM	Coherency strength	20	20	20
	Minimum number of columns	5	7	8
	Filtering procedure	75%	75%	75%
UniBic	K	2	4	5

C&C algorithm allows the values of δ and α to be 0.5 and 1.2, respectively [64]. Whereas the parameter K takes the exact number of implanted biclusters for synthetic data and $K = 100$ for real data. BiBit [284] algorithm works with binary data. We directly use the synthetic data without normalizing it. The discretization step initiates by dividing the range $[-3, 3]$ into equally-spaced 12 [284] levels and have discretized the expression value between 0 and 11, where each value corresponds to one level. For each level, a new binary matrix will be created. Here we consider only the highest level, such as 3 [266]. The new matrix is composed of 1 if the expression value is greater or equal to 1 or 0 otherwise. It also requires a minimum number of rows and a minimum number of columns

Table 4.5: Parameter values of different biclustering algorithms on overlapping data (scenario 3).

Method	Parameters	Overlap experiment			
		0×0	3×3	6×6	9×9
C&C	K	3	3	3	3
BiBit	R_{min}	-	-	-	-
	C_{min}	-	-	-	-
BicSPAM	Coherency strength	20	20	20	20
	Minimum number of columns	7	7	7	7
	Filtering procedure	75%	75%	75%	63%
UniBic	K	3	3	3	3

as input for the expected bicluster. According to the original paper, the values of R_{min} and C_{min} is selected as 2 and 2, respectively. These are used in a real dataset. But in the case of the synthetic dataset, we provide the correct number of input for expected biclusters for synthetic data.

The default parameterization of the BicSPAM [134] algorithm includes row-oriented normalization, Gaussian discretization, removal of missing values, row-oriented IndexSpan pattern, and merging procedure with 80% overlapping. For determining more meaningful biclusters we keep the same overlapping value as the OPBic algorithm. The algorithm decreases the support threshold iteratively until it discovers 50 non-similar biclusters. Moreover, we use order-preserving as coherency assumption, 80 as quality, zero-entries is being removed and no noise is present in the dataset. The reliability of the algorithm depends on a single iteration and the number of the minimum condition is determined by the square root of the number of columns. These parameter settings are the same for both synthetic and real datasets.

The default values of the parameters k, f, c, q, and r for UniBic [337] are 5% of columns (minimum 2), 1, 0.85, 0.50, and column number of background matrix. We have only adjusted the parameter K for different datasets. We keep $K = 100$ for real dataset.

Parameter settings for OPBic algorithm

OPBic algorithm takes four parameters such as minimum no. of rows R_{min}, minimum no. of columns C_{min}, maximum overlap O_{max}, and number of workers W. The serial version of the algorithm takes the same input parameters, where W is treated as the number of data partitions. To determine the number of workers W for the OPBic algorithm, we have experimented on datasets given in Table 4.6. The parameters of the OPBic algorithm for synthetic datasets are chosen

Table 4.6: A description of synthetic datasets used to determine number of workers.

Background matrix size	100×30	200×50
Implanted biclusters	2	3
Bicluster rows $[R_L, R_U]$	[10,15]	[15,25]
Bicluster columns $[C_L, C_U]$	[9,10]	[10,12]
Total biclusters area %	8	6.7

very carefully and experimentally given in Table 4.7. To select R_{min} and C_{min}, we take the additive-multiplicative biclustering model as it is considered as the most difficult case for identifying the biclusters. R_{min} and C_{min} are chosen in such a way so that it resembles the exact or similar number of biclusters as it is planted. The parameter O_{max} is considered as 0.25, as mentioned in the paper [278]. For determining W, we run the parallel and serial version of OPBic depending on the availability of the workers i.e., from 2 to 10 on trend-preserving biclustering model because the algorithm is capable to determine trend-preserving biclusters. We calculate the time for parallel (t_p) and serial (t_s) version to identify the biclusters. For both versions, we run the algorithm 5 times for each worker. After that, we compute the average speed-up $\mathbb{S}$ and efficiency $\mathbb{E}$ using the Equation 4.6.1 and 4.6.2, respectively.

$$\mathbb{S} = \frac{t_s}{t_p} \tag{4.6.1}$$

$$\mathbb{E} = \frac{S}{W} \tag{4.6.2}$$

Table 4.7: Parameter settings for synthetic datasets.

Background matrix size	No. of biclusters	R_{min}	C_{min}	O_{max}	W
100×30	2	6	6	.25	5
200×50	3	7	7	.25	10
500×50	1	16	16	.25	10
500×50	4	7	7	.25	10
1000×70	5	8	8	.25	10

Figure 4.8 depicts the speed-up and parallel efficiency curve, from which we can easily determine the number of workers. For dataset 100×30 and 200×50, maximum speed-up can be found at worker numbers 5 and 10, respectively. Since the maximum number of workers is 10 and from the Figure 4.8 it can be observed that larger dataset among these two datasets the maximum speed-up can be found at worker number 10. So, for the rest of the paper, we have considered the

number of workers as 10.

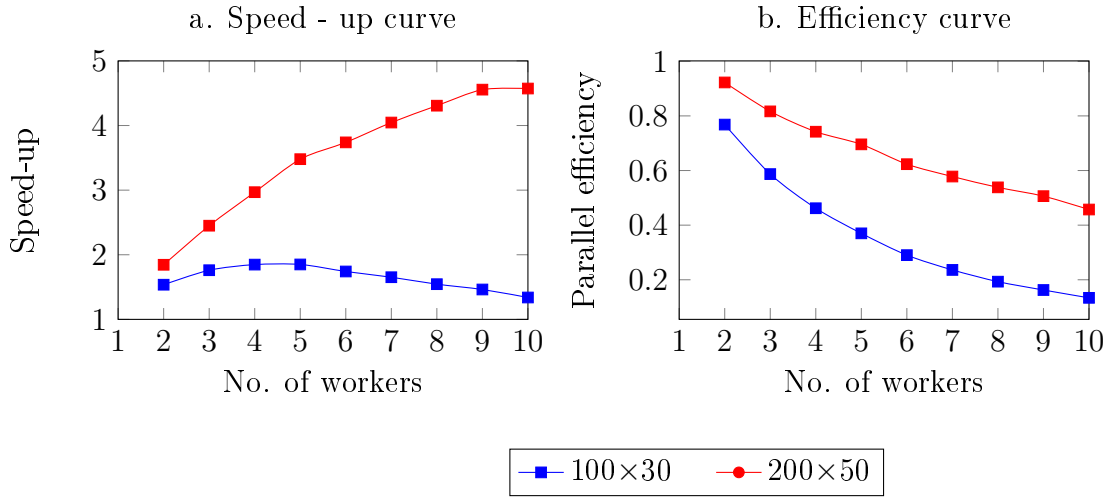


Figure 4.8: Speed-up and efficiency curve on two synthetic data.

For the overlap experiment, we set the parameters as $R_{min} = 9, C_{min} = 9, O_{max} = 0.25$, and $W = 10$ for all the overlapped synthetic datasets except for the highest overlapped degree. In this case, the hidden bicluster is overlapped by 9 rows and 9 columns, so that the proportion of mutual element is 36% size of the implanted bicluster is of [266]. Therefore, we allow a maximum overlap of 0.36 for the OPBic algorithm in this scenario.

4.6.2 Performance on synthetic datasets

The accuracy of a resulting bicluster can be assessed in multiple ways with respect to implanted biclusters. The most popular and widely used evaluation metric is *MS* (Equation 2.3.30) reported in [278] and presented in Chapter 2.

Results for scenario 1

The average performance of the first set of the synthetic dataset over all biclustering algorithms is presented in Figure 4.9. The reason behind this experiment is to avoid misleading results by considering all biclustering models. In this figure, we see that OPBic and UniBic perform consistently well in all test cases. Table 4.8 summarizes the model selection of each of the biclustering algorithms which are subsequently used in the second set of artificial data. In this table, we put a symbol 'Y' to indicate that the algorithm specified by the corresponding row is capable to identify the model given in the column. For each biclustering algorithm, the model is chosen in such a way that the average relevance and recovery score achieves a value more or equal to 0.7.

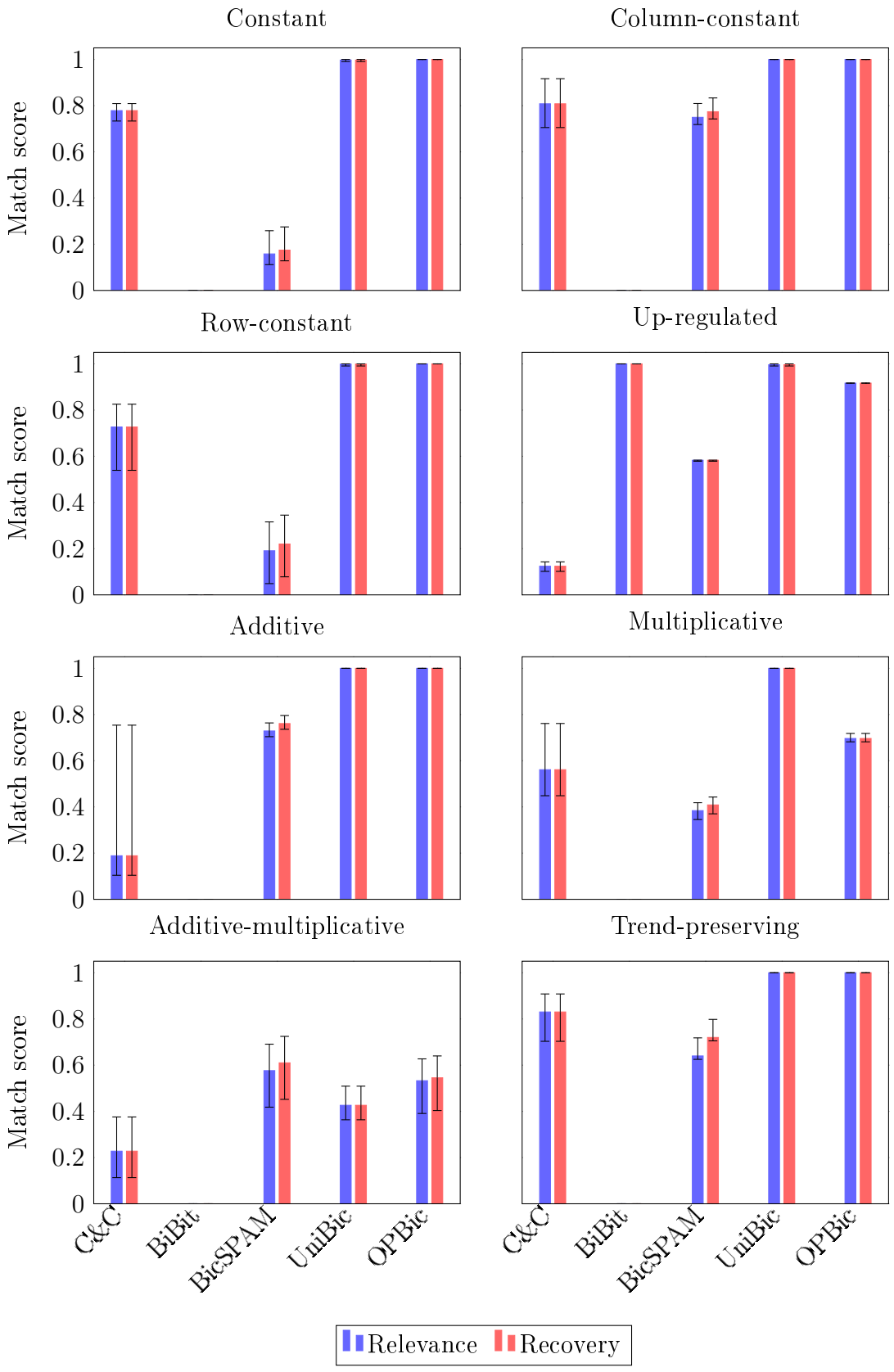


Figure 4.9: Relevance and recovery scores with error bars (range) of different biclustering algorithms on eight different biclustering models over scenario 1.

Table 4.8: Model selection of each of the biclustering algorithms.

Algorithm	Cnst	C_cnst	R_cnst	U_reg	Add	Mul	A_mul	Trnd_pres
C&C	Y	Y	Y					Y
BiBit	-	-	-	Y	-	-	-	-
BicSPAM		Y			Y			
UniBic	Y	Y	Y	Y	Y	Y		Y
OPBic	Y	Y	Y	Y	Y	Y		Y

Abbreviations: Cnst: Constant, C_cnst: Column-constant, R_cnst: Row-constant, U_reg: Up-regulated, Add: Additive, Mul: Multiplicative, A_mul: Additive-multiplicative, T_pres: Trend-preserving

Among all biclustering approaches, OPBic and UniBic are capable of identifying all biclustering models with good average relevance and recovery scores (above 0.7) except the additive-multiplicative type. It is worth mentioning that the higher recovery score means that the algorithm is quite capable of finding implanted biclusters and a higher relevance score describes that the found biclusters are highly relevant to the original biclusters. In the case of the constant biclustering model, OPBic is equivalent to the UniBic algorithm with an average relevance score of 1 for OPBic and 0.997 ($\simeq$ 1) for UniBic, and an average recovery score of 1 for OPBic and 0.997 ($\simeq$ 1) for UniBic. Similarly, the performance of the OPBic algorithm is equivalent to that of the UniBic algorithm in the case of the row-constant model with both average relevance score and average recovery score of 1 for OPBic versus 0.999 ($\simeq$ 1) for UniBic. The OPBic and UniBic algorithms overwhelmingly outperform all other competing biclustering methods for column-constant, additive and trend-preserving models by scoring 1 for both average relevancy and average recovery score. In the Up-regulated model, BiBit and UniBic algorithms successfully identify biclusters with average relevance and average recovery score of 1 and 0.996 ($\simeq$ 1), respectively. OPBic discovers up-regulated biclusters with a match score (both relevancy and recovery) of 0.92. For the additive-multiplicative model, BicSPAM performs significantly better than other biclustering techniques with an average relevance score of 0.58 compared to the second-highest average relevance score of 0.53 for OPBic.

C&C shows the best performance with a data matrix with trend-preserving biclusters and is also good for discovering constant type biclusters. So, it is expected that C&C can discover up-regulated bicluster also. However, Figure 4.9 confirms that it shows the poorest output for up-regulated biclusters among the six algorithms. The reason behind this behavior can be found in [266]. BiBit is only capable of discovering up-regulated biclusters because we use the

highest level in a discretization step, which considers only up-regulated values. BicSPAM achieves moderately good performance for the column-constant and additive biclustering model.

Taking all the results together, the overall comparisons confirm that UniBic is slightly inferior to OPBic in the case of constant, row-constant and additive-multiplicative models. On the other hand, the performance of UniBic is higher than OPBic in terms of up-regulated and multiplicative models. So, we can state that OPBic significantly outperforms almost all other biclustering algorithms and is a close competitor of the UniBic algorithm. Next, we analyze the performance of the proposed algorithm in obtaining biclusters with a varying number of implanted biclusters.

Results for scenario 2

To study the capability of the OPBic algorithm with an increasing number of biclusters, we further test the selected algorithms which successfully identifies bicluster models in the previous experiment with the second set of synthetic datasets on the eight models. For each biclustering algorithm, the model is chosen in such a way that the average relevance and recovery score is more or equal to 0.7 in scenario 1. No algorithm performs well for additive-multiplicative biclusters. Therefore, we consider only OPBic and UniBic results for the additive-multiplicative patterns though they do not perform well in scenario 1. The comparison results of relevance and recovery scores of all approaches are given in Figure 4.10. The figure suggests that the OPBic algorithm performs best in terms of relevance and recovery scores for constant, additive, additive-multiplicative, row-constant, and trend-preserving models throughout all test matrices in comparison to all competing biclustering methods. UniBic holds the second-best position for aforementioned bicluster types in terms of both relevance and recovery scores. We execute C&C for constant, column-constant, row-constant and trend-preserving types and found it to be the worst performer in this scenario. In the column-constant model, the recovery scores of the OPBic algorithm are higher than UniBic, C&C, and BicSPAM but the relevance scores of BicSPAM are higher than OPBic. A fluctuating result can be seen for the multiplicative model. In the multiplicative model, OPBic performs lower than UniBic for smaller sized matrices, equivalent performance for medium-sized matrices and is better than UniBic for large-sized matrices. As in the previous case, here also no algorithm can beat BiBit for up-regulated bicluster type with an increasing number of biclusters and test matrices. Thus, from the overall comparison, we can

conclude that OPBic outperforms all other biclustering algorithms for scenario 2.

Results for scenario 3

We test the four biclustering algorithms on synthetic datasets with overlapping biclusters for trend-preserving. The purpose of this experiment is to test the efficacy of our algorithm whether it can identify the overlapped biclusters or not. We have considered all biclustering algorithms except BiBit, as it only works for up-regulated bicluster type. The average relevance and recovery scores of each of the test matrices are shown in Figure 4.11. OPBic can find implanted biclusters with (i) no overlap and (ii) 3×3 overlap with higher relevance and recovery scores than the other three algorithms. Another observation is that OPBic has high relevance scores as the overlap degree increases in contrast to all other algorithms. But a strange behavior can be found with overlap degree 6×6, where UniBic has better recovery scores than our algorithm. The possible reason for such behavior is due to three datasets among ten datasets, where the implanted biclusters were not found accurately. As we take the average of the results of all 10 datasets, it impacts the average recovery and relevance values. The results are also dependent on the minimum number of conditions and it may identify biclusters with a lower number of conditions that are already eliminated in the preliminary investigation of our algorithm. However, considering all overlapping degrees OPBic shows better performance with respect to recovery scores than any other biclustering algorithms except UniBic for 6×6 overlap. Regarding the relevance score then OPBic outperforms all three algorithms. C&C algorithm shows the worst performance for identifying overlapping biclusters. For most of the overlapping degrees, it has been found that UniBic has better recovery scores than BicSPAM but sometimes it displays lesser relevance scores than BicSPAM. Taking all of this into account UniBic is the second best and BicSPAM is the third-best performer in this regard. So, overall we can say that OPBic is capable of discovering overlapped biclusters.

4.6.3 Results for real datasets

This section presents experiments, dealing with real data and discusses biclustering results which are obtained by the proposed method. In order to evaluate the effectiveness of the OPBic algorithm, we use three cancer microarray gene expression datasets i.e., Laiho, Singh, and GSE20437. Datasets Laiho and Singh are already mentioned in Chapter 3, Section 3.6.2.

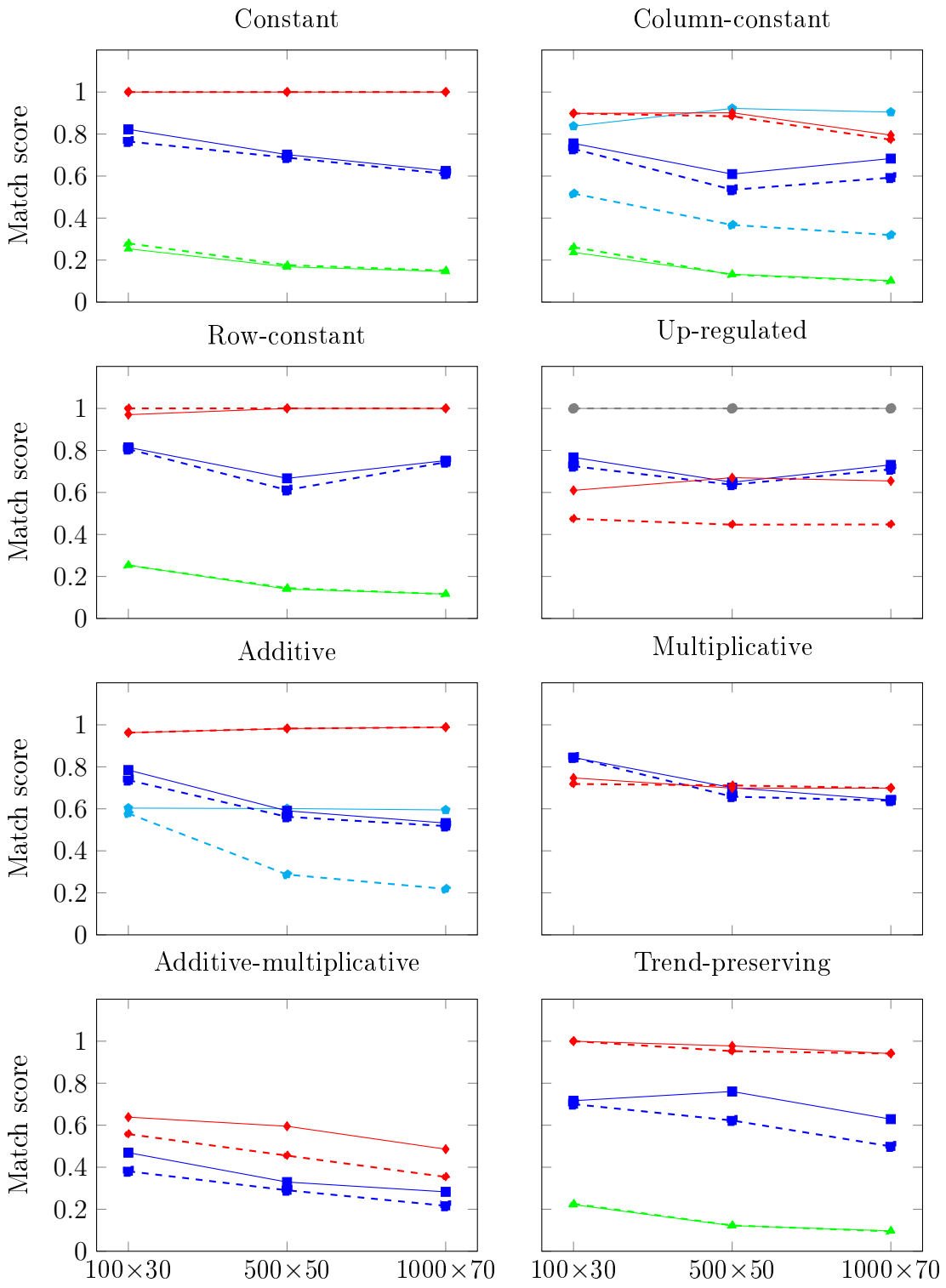


Figure 4.10: Relevance and recovery scores of different biclustering algorithms for eight different biclustering models in scenario 2.

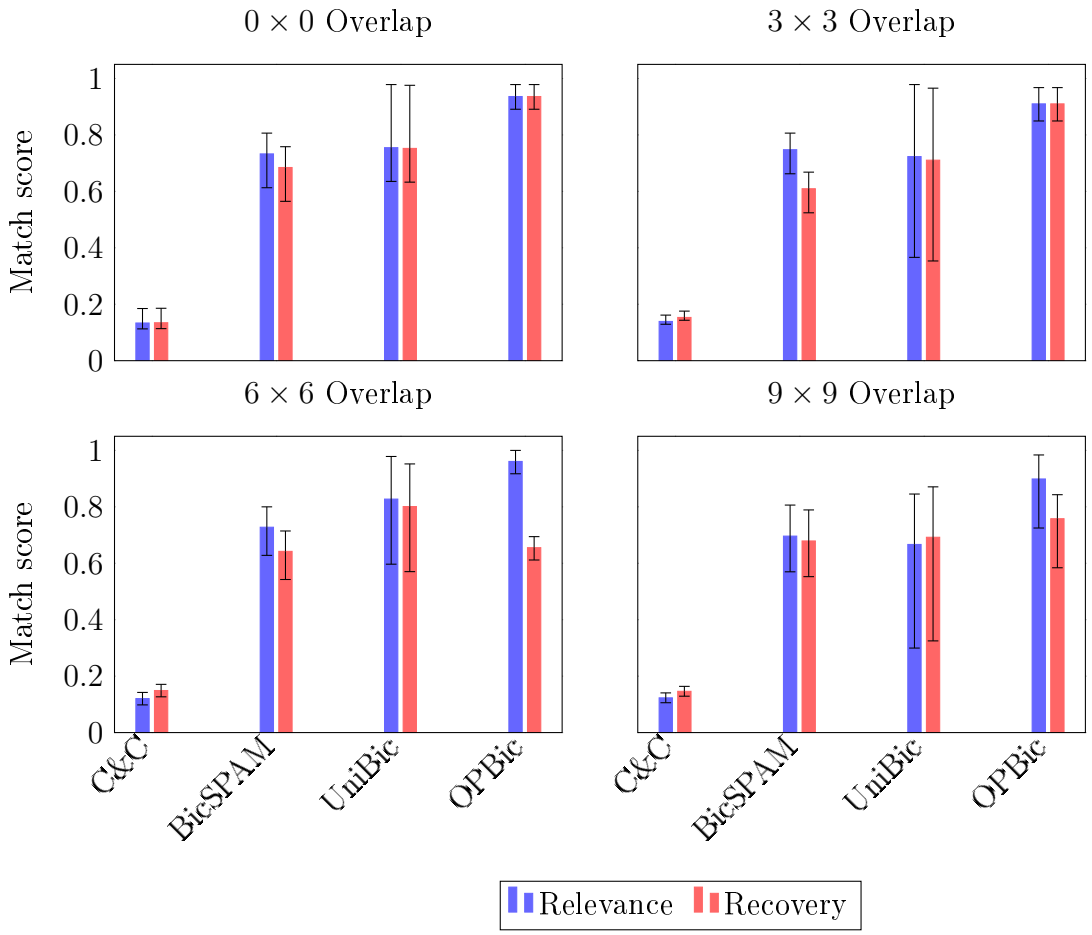


Figure 4.11: Relevance and recovery scores with error bars (range) of different biclustering algorithms for the trend-preserving model with overlapping biclusters (scenario 3).

GSE20437: GSE20437 is downloaded from National Center for Biotechnology Information (NCBI). The dataset uses Affymetrix HU133A microarrays to investigate the breast cancer gene expression data of 22283 genes over 42 samples [121]. The samples are categorized into three groups, 18 women considered to be a usual breast cancer risk and undergone mammoplasty reduction (RM), 18 women diagnosed with breast cancer undergoing surgery, 9 of them Estrogen receptor (ER+) and 9 (ER−) for breast tumor (HN) and 6 high-risk patients undergoing prophylactic mastectomy (PM). We have identified Affymetrix probes that are mapped to the same official gene symbols. Then, we attempt to keep only one Affymetrix probe with maximum standard deviation among the same official gene symbols. Thus, we reduce the dataset to 12982 genes and 42 samples. Finally, we have selected the top 5% genes from the reduced dataset based on higher standard deviation. Thus, the number of genes in the dataset is 649. We normalize this dataset with μ 0 and σ 1.

Table 4.9: GO enrichment analysis result of different biclustering algorithms on cancer microarray datasets

Algorithm	Total biclusters	Enriched biclusters					
		BP		MF		CC	
		T	%	T	%	T	%
C&C	300	40	13.33	20	6.67	18	6
BicSPAM	26	15	57.69	8	30.77	14	53.85
UniBic	177	100	56.50	69	**38.98**	92	51.98
OPBic	300	194	**64.67**	114	38	181	**60.33**

Abbreviation: T-Total.

We execute biclustering algorithms with real datasets. For OPBic algorithm, the parameter C_{min} is chosen by $\lceil 5\%$ of the total number of samples$\rceil$[337] (default 4) and R_{min} (default 4) is the same as C_{min}, according to all synthetic data experiments. We set $W = 10$, $O_{max} = 0.25$ and considered only first 100 biclusters [64] as an output. The BiBit algorithm does not work for all types of biclusters; therefore we have excluded the result from real datasets.

Enrichment analysis

To compare results of different biclustering algorithms on real data, we perform functional enrichment analysis based on GO categories with the help of FuncAssociate [35], assuming the level of significance as 5%. A bicluster is said to be enriched if the p-value of one of the GO terms is less than the level of significance. Table 4.9 shows the percentage of enriched biclusters for different biclustering algorithms for three distinct domains, viz. BP, MF, and CC. The percentage of enriched biclusters is calculated by Equation 4.6.3. A higher percentage of enrichment indicates better functional groupings. From the analysis, we can say that OPBic outperforms all other biclustering algorithms.

$$\% \text{ of enriched biclusters} = \frac{\text{Number of enriched biclusters}}{\text{Total number of biclusters}} \times 100 \qquad (4.6.3)$$

Subtype identification

To test the effectiveness of the OPBic algorithm, additionally, we perform subtype identification to establish our biclustering algorithm which gives importance to both genes as well as samples. Most of the review works available in the literature, evaluate the various biclustering algorithms solely depending on genes and does not consider the samples involved in a bicluster. It is important to compare

Table 4.10: Subtype identification with different biclustering algorithms for three cancer gene expression datasets.

Algorithm	Subtypes identified								%
	Laiho		Singh		GSE20437				
	SCRC	CCRC	Normal	Tumor	RM	ER+	ER-	PM	
C&C	0	44	17	13	0	0	0	0	24.67
BicSPAM	0	6	0	0	0	0	0	0	23.07
UniBic	0	7	0	0	1	0	0	0	4.52
OPBic	0	51	5	3	4	0	1	0	21.33

the biclustering algorithms not only in the context of genes but also samples, as bicluster comprises of subsets of genes and subsets of samples, which differentiates it from traditional clustering results. In literature, a very limited amount of work has been done in order to judge biclustering algorithm based on samples [66, 262].

We focus on the true effectiveness of biclustering algorithms for retrieving the homogeneous types of samples in a bicluster known as subtypes. Identification of subtypes quantifies how well an algorithm can differentiate the samples of the datasets. Suppose, there are d different types of samples and K is the total identified biclusters. Let us again consider a bicluster β_k and $1 \leq k \leq d$, where all the rows are a subset of the dataset and all the columns are associated with samples of type k. $1 \leq |\beta_k| \leq K$, denotes the number of biclusters for sample type k. Therefore, our target is to identify such types of biclusters among the resulting biclusters. Table 4.10 reports the percentage of subtypes identified by each of the algorithms. From second to ninth columns indicate the number of biclusters that successfully discovers individual subtype for each of the datasets. The tenth column represents the percentage of aggregated subtypes with respect to the total number of biclusters. Taking all the datasets together it has been found that OPBic achieves the third position after C&C and BicSPAM. In this experiment, UniBic does not perform well in recognizing homogeneous samples. From Table 4.10, we conclude that OPBic can identify maximum homogeneous types of samples for breast cancer (GSE20437) and colon cancer (Laiho) datasets.

4.6.4 Results of miRNA breast cancer data: a case study

Breast cancer is a frequently diagnosed malignant disease among women in developing countries and is the second most common cause of mortality. Early-stage disease detection and response to treatment play an important role in good prog-

nosis because recurrence of breast cancer is generally incurable [187, 299]. Breast cancer is well known as a heterogeneous disease, which is a collection of breast tumors associated with multiple entities with a variety of histopathological features and clinical characteristics [323]. Currently, breast tumors are molecularly classified into five distinct types: Basal-like, HER2 (Human Epidermal growth factor Receptor 2), Luminal A, Luminal B, and Normal-like [335]. The management of treatment for this critical disease is largely dependent on these subtypes, which help in diagnostic prediction [127].

A broader way to categorizing breast tumor is by presence or absence of Immuno-Histo-Chemical (IHC) markers, such as Estrogen Receptor (ER), Progesterone Receptor (PR), and HER2 Protein. The lack of all these IHC markers defines a tumor as Triple Negative Breast Cancer (TNBC) [48] and the presence of all the three receptors is known as non-TNBC [102]. Women diagnosed with TNBC are more likely to have low survival rates (up to five years), increased aggression, poor prognosis, and higher recurrence with non-TNBC cancer [48]. The proper diagnosis of TNBC is a challenging problem and is therefore gaining a lot of focus. Moreover, the diversity of pathological features creates challenges in the prognosis of this complex disease. Although significant advancements have been made in recent years towards therapeutics, the biology of breast cancer makes it complicated for treatment procedures [102]. Thus, there is always a need for improvement in the treatment and therapy given to the patients [354]. Estimation of the risk factors of cancer plays an important role for better diagnosis of breast cancer [127].

We use the breast cancer miRNA expression data from a large cohort of patients. The data is collected from Supplementary Table S4D of the work by Farazi et al. [108]. This expression profile can also be downloaded from NCBI with GSE28884 accession number. The dataset is composed of 1112 mature miRNA and 185 breast specimens including 17 non-invasive, 151 invasive breast carcinomas, 6 cell lines, and 11 normal breast tissues. The miRNA expression profile is obtained by barcoded Solexa-sequencing and based on sequence read numbers [108]. Breast cancer is characterized by the presence or absence of immunohistochemical (IHC) markers, such as Estrogen Receptor (ER), Progesterone Receptor (PR), and Human Epidermal growth factor Receptor 2 (HER2) protein. According to clinical and histopathological features the patients are classified as Ductal carcinoma in situ (DCIS), Invasive ductal (IDC), Cell-line (MCF7, MCF10A, HCC38, BT474, MDA-MB134, and ZR-751), Normal samples (healthy), and others (Mucinous A, Atypical Medullary, Metaplastic, Apocrine

Adenoid, ILC). Using the second characteristic, molecular subtypes, the patients are divided into Normal, HER2, Basal, Luminal A, Luminal B, and NA. Figure 4.12 summarizes the composition of patient's information and clinical information for the miRNA breast cancer dataset.

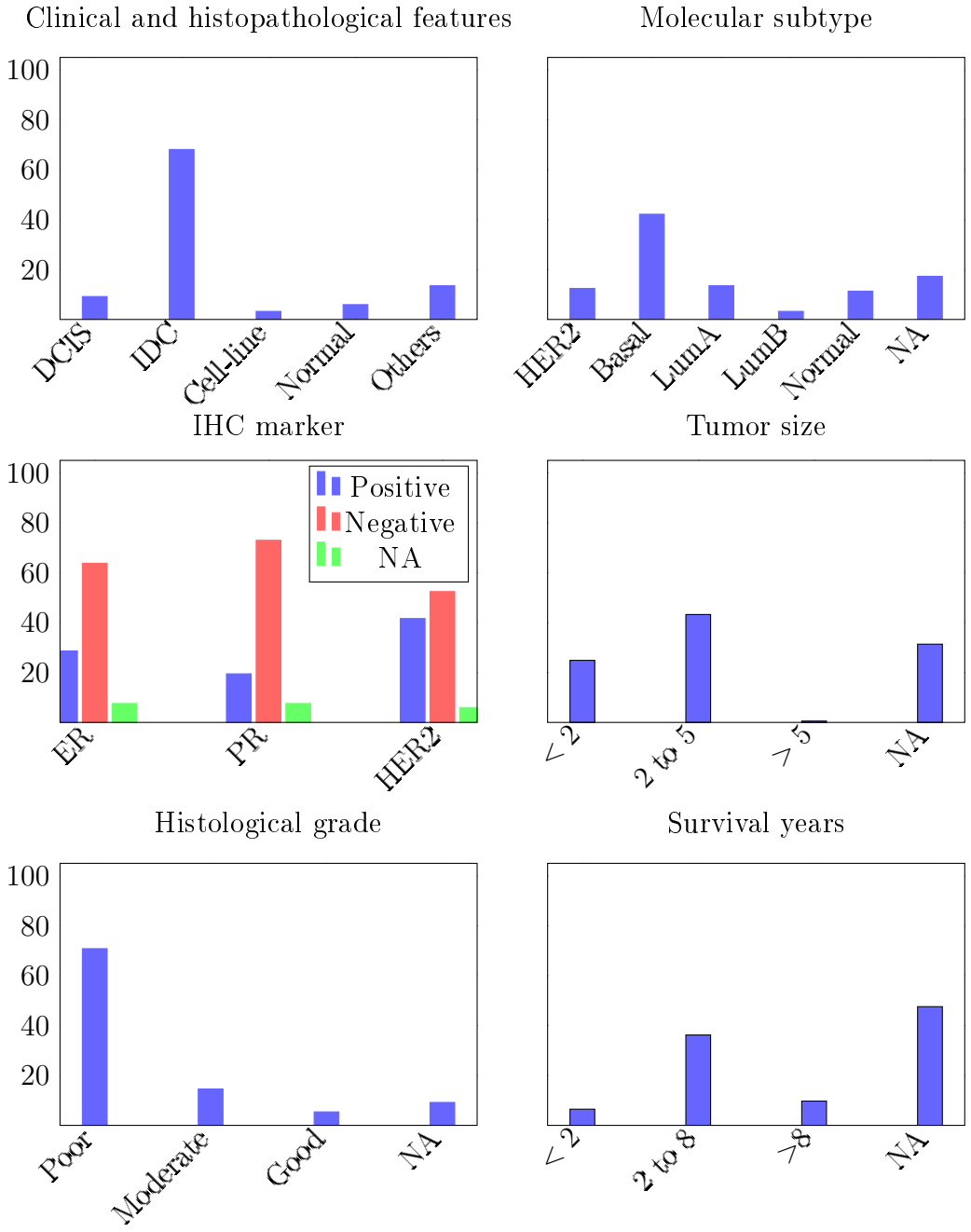


Figure 4.12: Clinical annotations for 185 patients. The bar shows the percentage of values for each clinical category. It includes clinical and histopathological features, molecular subtypes, IHC marker (presence or absence), Tumor size (cm), histological grade, and survival years.

The dataset is preprocessed by removing those miRNA samples which

have low expression values i.e., less than 20 across all samples [222]. After this step, the number of miRNAs is reduced to 534. Then, we perform z-score normalization on the reduced dataset. We run the OPBic algorithm with $R_{min} = 10, C_{min} = 10, O_{max} = 0.25$, and $W = 10$ parameter settings. We obtain 68 biclusters with an average of 12 samples in approximately each of the biclusters using the OPBic algorithm on this dataset.

Associated samples of each bicluster identifies similar clinical characteristics

To determine whether the resulting biclusters detected by OPBic are biologically and clinically meaningful or not, for each bicluster we assess whether the samples associated with it have similar clinical features or not, and whether the associated subset of miRNAs has a distinct pathway [335]. To do so, we compute the proportion of clinical-pathological variables for the associated samples for each bicluster. The proportion of clinical annotation for each bicluster is summarized in Figures 4.13 to 4.20. This result suggests that biclusters have not been generated by chance, rather they have strong clinical relations. Moreover, the result shows that clinicopathological outcomes of the biclusters are similar. Bicluster 22 is an aggressive [335] one, with 10 tumor samples. Among the tumor samples, 90% tumors are of TNBC or 100% of PR-; 80% of Basal subtypes; 70% of IDC; high (100%) frequency of poor tumor grade; 40% are of <2 cm in size; and 40% of 2 to 8 years survival years. On the other hand, the characteristics of bicluster number 41 show that the tumors of the patients with this tumor subgroup are less aggressive. It consists of 11 tumor samples where 90.91% of tumors are of HER2-; 27.27% are Basal or 27.27% of Luminal A subtypes; 45.45% of IDC; 36.36% of tumors are <2 cm in size; 9.09% have 2 to 8 and 9.09% of >8 survival years; and 27.27% are moderate and 27.27% poor tumor grade. From the analysis, we note that a somewhat low ($\leq$60) percentage of TNBC patients for each bicluster has more than 8 years survival years. Another distinguishing feature is that if patients have a higher ($\geq$70) percentage of TNBC in one subgroup, they also have more than 8 survival years if tumor size is $<$ 2 cm. The percentage of clinical composition for each sample across all identified 68 biclusters from OPBic algorithm is given in Figures 4.13 (percentage of ER), 4.14 (percentage of PR), 4.15 (percentage of HER2), 4.16 (percentage of clinical and histopathological), 4.17 (percentage of molecular subtype), 4.18 (percentage of tumor size), 4.19 (percentage of histological grade), and 4.20 (percentage of survival years).

Figure 4.13: Percentage of IHC markers (ER) for each sample across all biclusters.

Figure 4.14: Percentage of IHC markers (PR) for each sample across all biclusters.

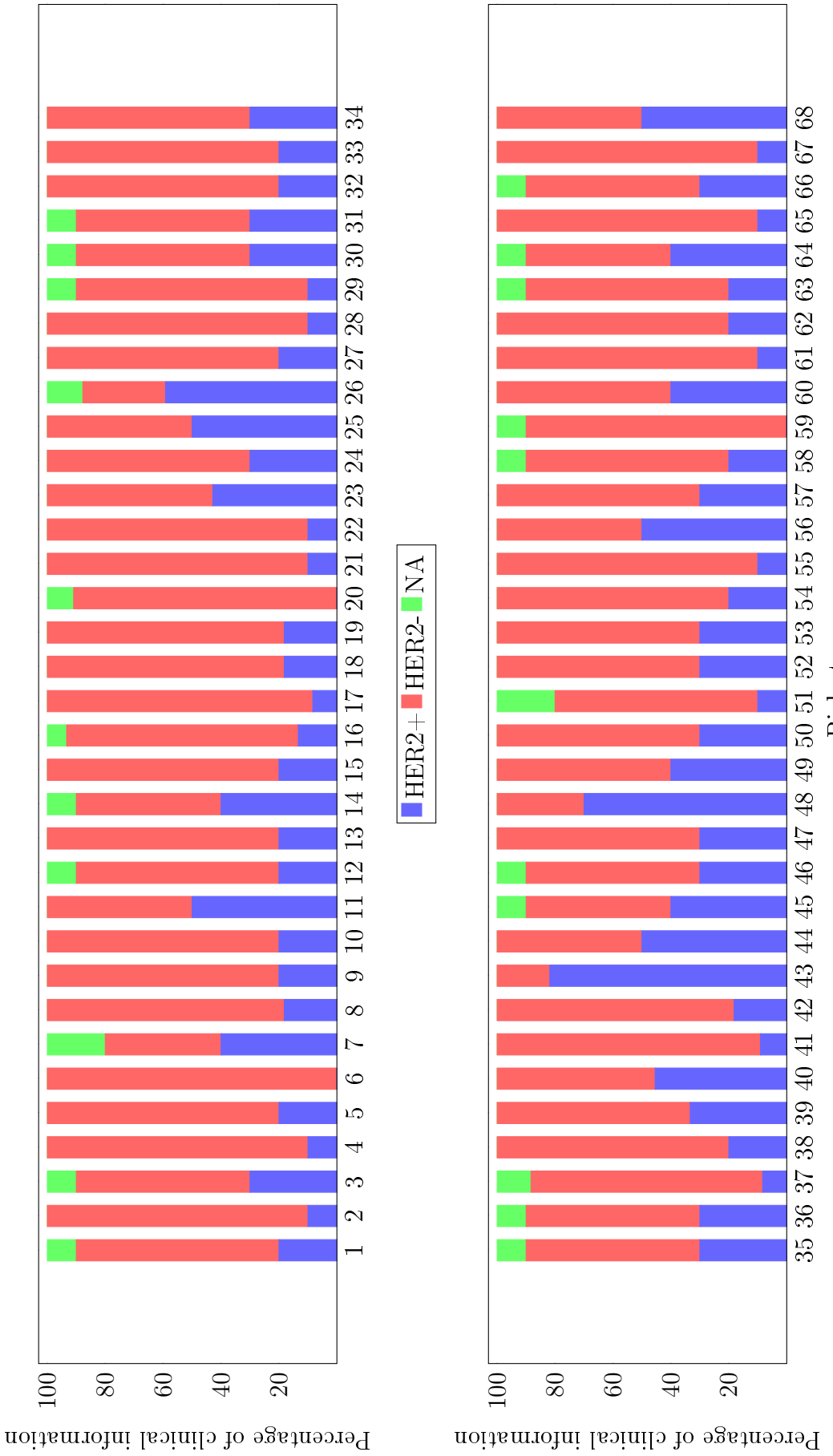


Figure 4.15: Percentage of IHC markers (HER2) for each sample across all biclusters.

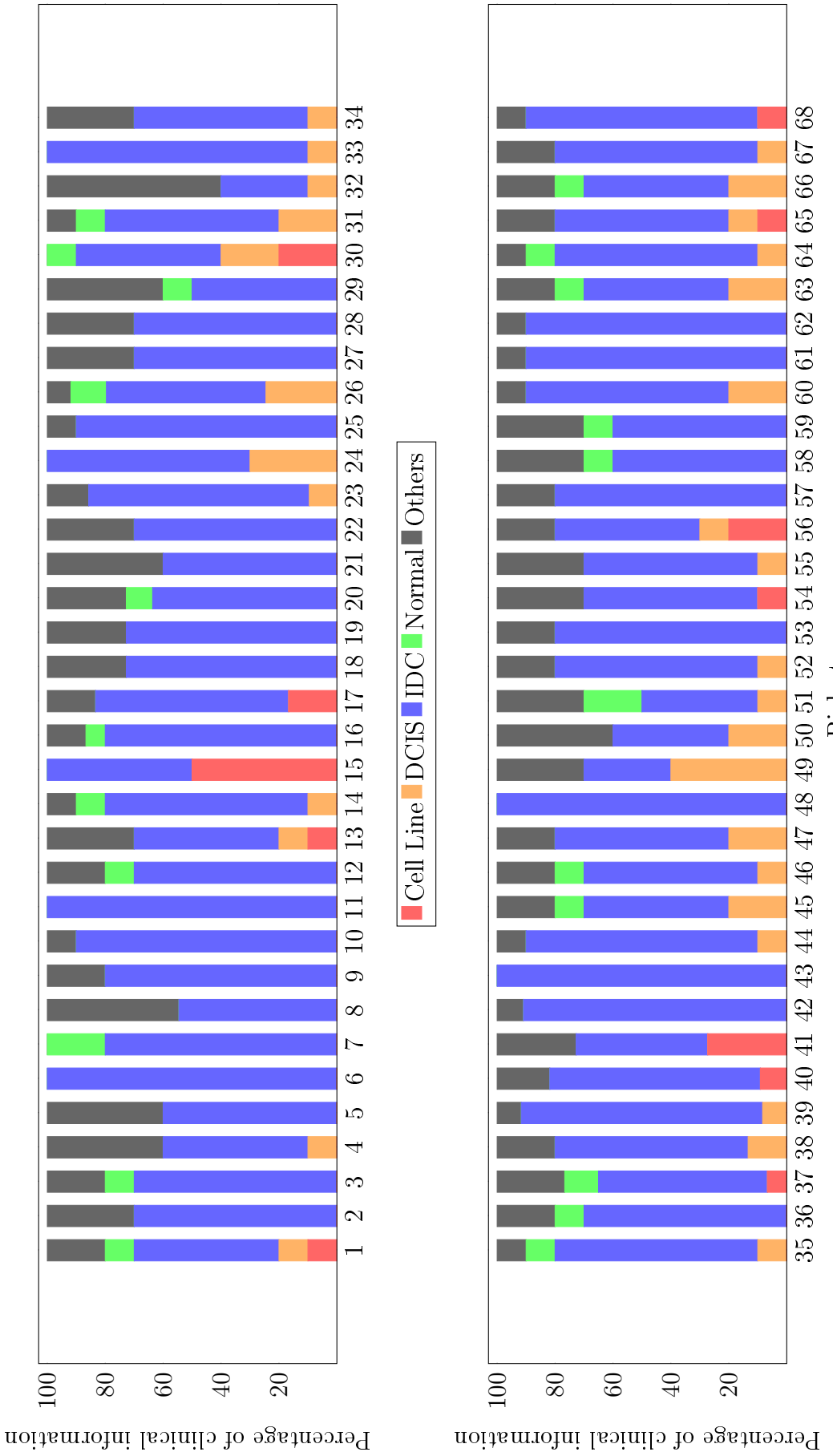


Figure 4.16: Percentage of clinical and histopathological features for each sample across all biclusters.

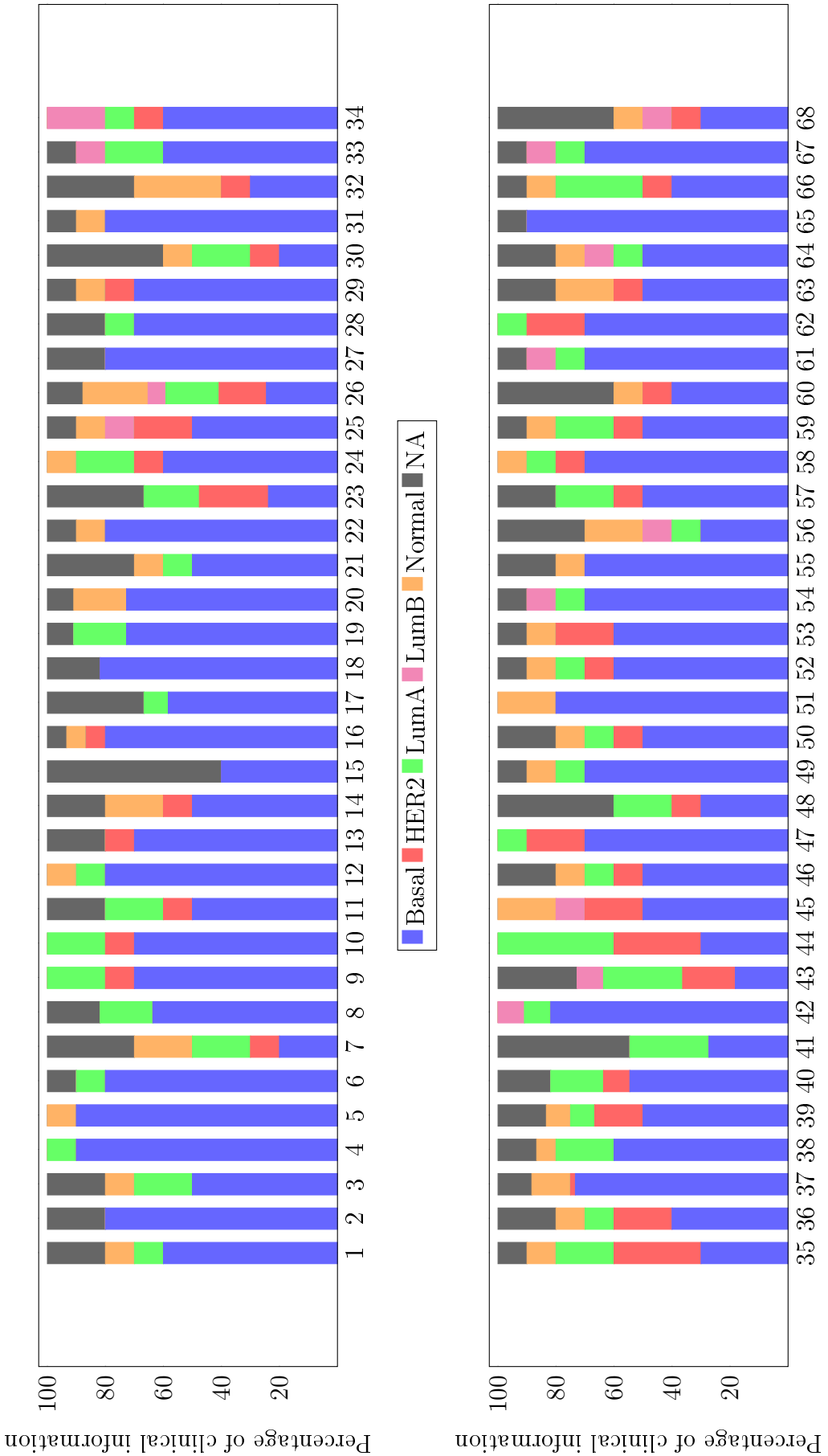


Figure 4.17: Percentage of molecular subtype for each sample across all biclusters.

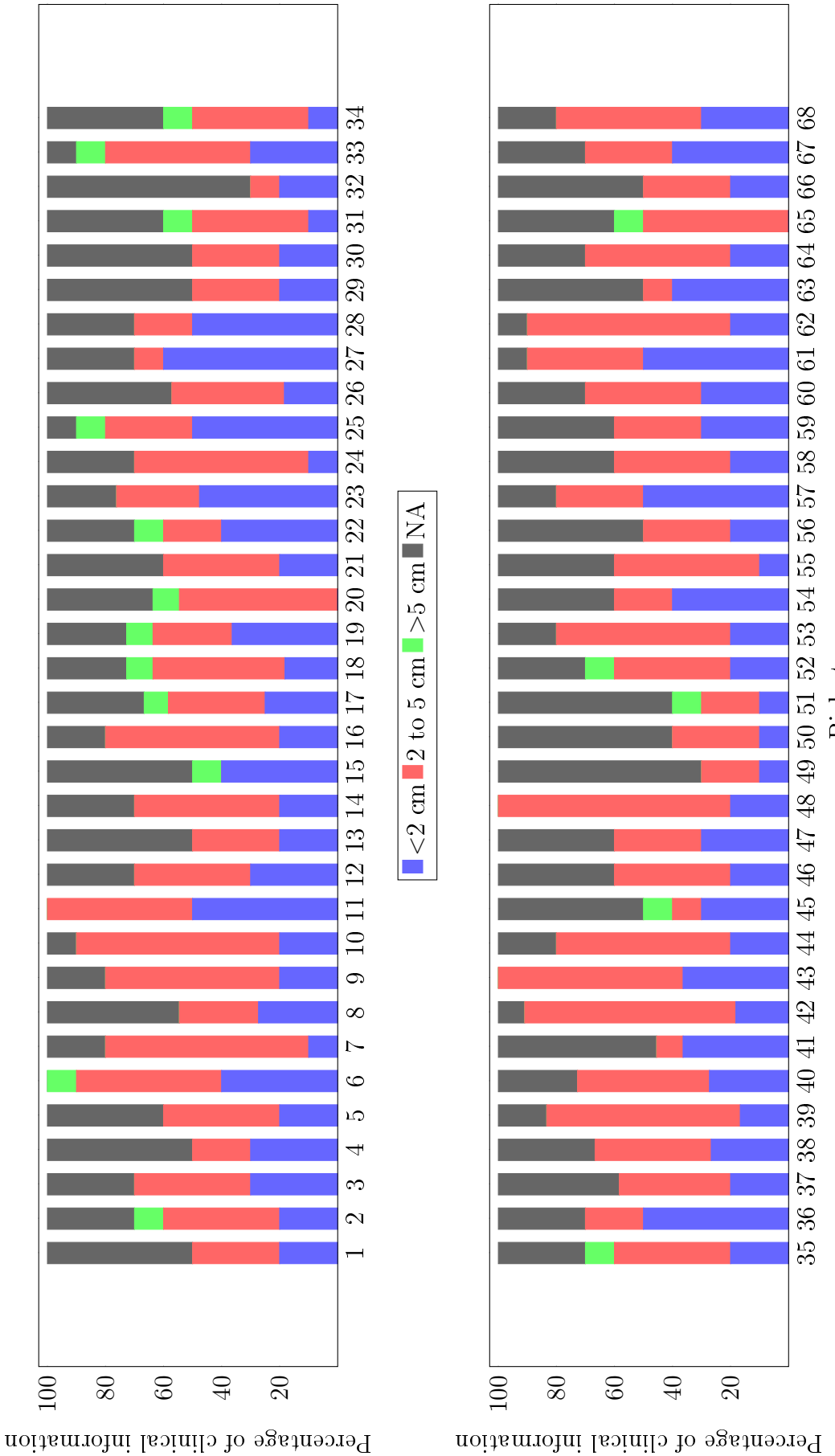


Figure 4.18: Percentage of tumor size for each sample across all biclusters.

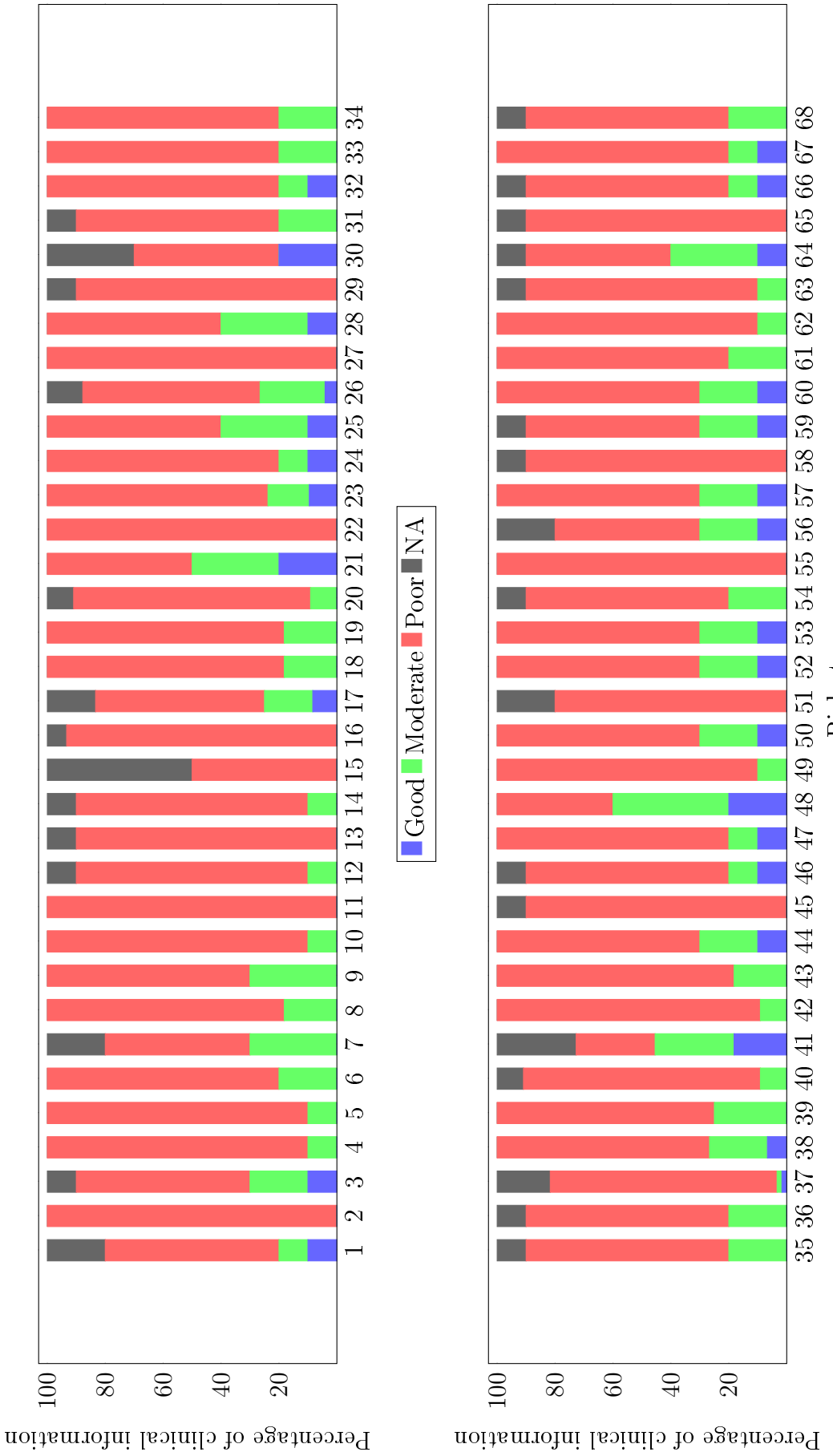


Figure 4.19: Percentage of Histological grade for each sample across all biclusters.

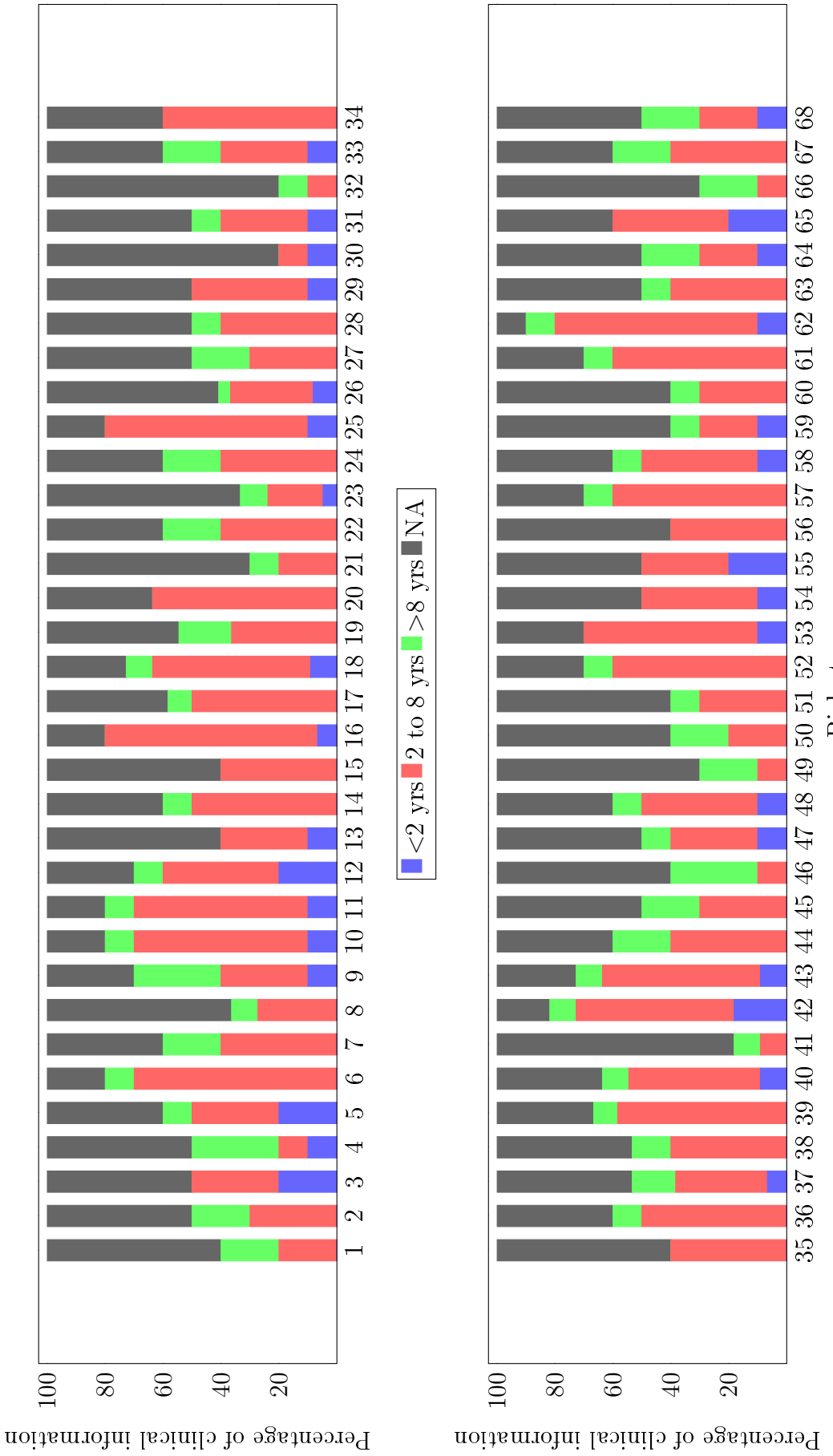


Figure 4.20: Percentage of overall survival years for each sample across all biclusters.

Table 4.11: A list of top 5 KEGG enriched pathways for the result of OPBic algorithm for miRNA dataset.

KEGG pathway	p-value	#Genes	#miRNAs
Prion diseases	5.11E-28	1	1
Fatty acid biosynthesis	1.65E-21	3	2
Glycosphingolipid biosynthesis - lacto and neolacto series	4.73E-12	5	4
Signaling pathways regulating pluripotency of stem cells	5.43E-11	60	4
Mucin type O-Glycan biosynthesis	5.43E-11	4	2
Proteoglycans in cancer	7.03E-10	98	24

Enrichment analysis

To investigate the function of each bicluster composed of miRNAs, we perform KEGG pathway and GO enrichment analysis with the help of DIANA miRPath v.3 [329], assuming the level of significance is 5%, with the help of TarBase v7.0 and specifying human as the species for each of the biclusters. This search provides the enriched KEGG pathway and GO annotation terms with less the 5% p-values. The list of miRNAs for each bicluster is submitted to the web application for computing the p-values of related KEGG pathways and GO categories.

We find that 91.18% (62 biclusters) of all identified biclusters are significantly enriched with KEGG pathways. Interestingly, we notice that 31 and 27 biclusters are related to *TGF-beta signaling pathway* and *proteoglycans in cancer*, respectively. We also observe that *prion disease* pathway, which is involved in cancer is the most (based on lowest p-value) enriched one [71] among all the biclusters. The top 5 significant pathways are reported in Table 4.11 with corresponding p-values by aggregating all the biclustering enrichment results. Table 4.12 summarizes the top 5 enriched terms depending on the lowest p-value considering all the biclustering results of the OPBic algorithm together. GO category *organelle* is the most enriched term and we observe it in 59 biclusters. Other terms such as *ion binding* and *cellular nitrogen compound metabolic process* are found in 58 and 57 biclusters, respectively.

4.7 Potential biomarkers identification

We further analyze our algorithm on the basis of biomarker identification capability. We propose another biomarker identification technique, named frequency-based biomarker identification. Like network-based biomarker identification,

Table 4.12: A list of top 5 enriched GO categories for the result of OPBic for miRNA dataset.

GO ID	p-value	#Genes	#miRNAs
organelle	1.2E-225	3441	27
ion binding	1.4E-116	2111	27
cellular nitrogen compound metabolic process	1.2E-102	1654	27
biosynthetic process	1.88E-69	1384	27
cellular protein modification process	8.19E-67	887	27

frequency-based also uses two user-defined thresholds i.e., φ and ψ. Frequency-based biomarker identification solely relies on the biclustering results. Considering all the biclusters provided by an algorithm, we can calculate the frequency of each gene or miRNA. Frequency is nothing but a number of occurrences of a gene or miRNA in all the biclusters. In other words, how many times a gene or miRNA can be present in a set of biclusters. To recognize biomarkers, we select the top (higher) φ frequently occurred genes or miRNAs from all the biclusters. If the number of selected frequently occurred genes or miRNAs are more than say, ψ, we perform statistical analysis to identify significant genes or miRNAs on the expression matrix and select top (lower) φ p-valued genes or miRNAs as biomarkers. For this purpose, we use Multiple Experiment Viewer version 4_9_0 (MEV) [290] software for Kruskal Wallis statistical non-parametric test to select the significant genes or miRNAs based on the p-value.

φ value is obtained after exhaustive experimentation. At the very beginning of our experiment, we start the frequency-based method for microarray data with $\varphi = 1$, i.e., the highest frequent genes and $\psi = 10$, (as recommended in [253]). Next, we run it for $\varphi = 2$ and $\psi = 10$. It can be observed that the number of identified gene biomarkers are more than the initial parameter setting ($\varphi = 1$, $\psi = 10$). So, we opt for the second choice. It is possible to find more biomarkers if we increase the value of φ. However, if we keep on increasing the φ value it might so happen that all of the genes present in the biclusters become frequent and hence it will lead to performing the statistical test mentioned before for all frequent genes. This same effect might occur for the result of all biclustering algorithms leading to the same gene biomarkers. Therefore, in this study, We keep $\psi = 10$ and $\varphi = 2$ as the possible parameter values for biomarker identification throughout all the experiments.

We use both frequency-based and network-based approaches in order to identify biomarkers and they have been reported in Table 4.13 and 4.14, respec-

Table 4.13: Potential biomarkers identification of different biclustering algorithms using frequency-based method.

Dataset	Algorithm	Potential biomarkers
miRNA	C&C	*hsa-miR-148b**, *hsa-let-7a-3*, *hsa-miR-126*, *hsa-miR-128-1*, *hsa-miR-32*, *hsa-miR-361-5*, *hsa-miR-376a-2-5p*
	UniBic	*hsa-miR-449b*, *hsa-miR449c-5p*
	OPBic	*hsa-miR-410*, *hsa-miR-483-5p*
Laiho	C&C	*MUC5AC*, *NTRK2*
	BicSPAM	*COL3A1*
	UniBic	*LGALS1*, *COL3A1*, *SLC22A2*
	OPBic	*MAGEA2B*
Singh	C&C	*RAN*, *HSPA5*, *NCL*
	BicSPAM	*PDLIM5*
	UniBic	*RACK1*
	OPBic	*HINT1*, *TMBIM6*
GSE20437	C&C	*SNORD36B*, *GLI3*, *PAXBP1*
	BicSPAM	*IFT57*, *LSM2*, *CCZ1B*, *LOC101929240*, *GCC2*
	UniBic	*ZBTB5*, *GLT8D1*, *FICD*
	OPBic	*MLH1*, *SAE1*

Table 4.14: Potential biomarkers identification of different biclustering algorithms using network-based method.

Dataset	Algorithm	Potential biomarkers
miRNA	C&C	*hsa-miR-335-5p, hsa-miR-26b*
	UniBic	*hsa-miR-15a, hsa-miR-181d*
	OPBic	*hsa-miR-454, hsa-miR-137*
Laiho	C&C	*PTPRM, NTRK2*
	BicSPAM	*COL1A2, DCN*
	UniBic	*FN1, JUN*
	OPBic	*COL1A2, SPARC*
Singh	C&C	*RPL12, RPS5*
	BicSPAM	*RPS23, RPS3A*
	UniBic	*RPS23, RPS16*
	OPBic	*RPS16*, *RPS5*
GSE20437	C&C	*KARS*, *GMPS*, *KPNB1*
	BicSPAM	*RPL9*, *RPS4X*
	UniBic	*IARS*, *KARS*
	OPBic	*RPS23*, *RPL9*

tively. From the resulting biclusters of the OPBic algorithm, we successfully identify top (according to the highest frequencies) two frequency-based miRNA biomarkers *hsa-miR-410* and *hsa-miR-483-5p*, which we validate based on established results. In the literature [342], *hsa-miR-410* has been found as a circulating miRNA in stage II and III breast cancer patients. The role of *hsa-miR-483-5p* has been identified as breast cancer-associated miRNA. We also find the top (depending on the higher degree) two network-based biomarkers, viz. *hsa-miR-454* and *hsa-miR-137*. The evidence that these miRNAs are involved in breast cancer is found in the literature. It has been reported that *hsa-miR-454* acts as an oncogene or tumor suppressor in most cancers. Cao et al. [54] state that *hsa-miR-454* is a potential predictor of prognosis in TNBC. Zhao et al. [366] identify that an important component of breast cancer estrogen-related receptor α (ERRα) is regulated by a tumor suppressor miRNA biomarker *hsa-miR-137*.

We compare the biomarker identification results from OPBic with other competing algorithms except for BicSPAM as we have not found any biclusters from the BicSPAM algorithm for the miRNA expression dataset. C&C identifies *hsa-miR-148b**, *hsa-let-7a-3*, *hsa-miR-126*, *hsa-miR-128-1*, *hsa-miR-32*, *hsa-miR-361-5*, *hsa-miR-376a-2-5p* and UniBic algorithm discovers *hsa-miR-449b* and *hsa-miR- 449c-5p* miRNAs as potential biomarkers. *Hsa-let-7a-3*, *hsa-miR-126*, and *hsa-miR-128-1* discovered by C&C are established as biomarkers according to Human MicroRNA Disease Database version 2.0 (HMDD v2.0) [205]. Whereas the biomarkers found by UniBic can not be established as biomarkers by HMDD. Therefore, both BicSPAM and UniBic can not identify biomarkers in these cases.

The identified gene biomarkers are validated through literature. First, we discuss frequency-based biomarkers then network-based. The identified gene biomarkers are validated through previously published literature and Cancer Genetics Web[1]. From the Table 4.13, we can clearly observe that C&C, BicSPAM, UniBic, and OPBic algorithms identify a total of 8, 7, 7, and 5 numbers of frequency-based gene biomarkers from all cancer gene expression datasets. According to Cancer Genetics Web, genes *MUC5AC*, *LGALS1*, *HINT1*, *GLI3*, *MLH1*, and *RACK1* are strongly associated with different cancer types which can be found in different PubMed publications. *NTRK2* [62] is found to be involved in cell proliferation, apoptosis and differentiation through *PI3K*, *RAS/MAPK/ERK pathways*. Genes *COL3A1* and *SLC22A2* are treated as prognostic biomarkers of colorectal cancer [34, 334]. In the literature, genes

[1] `www.cancer-genetics.org`

HSPA5, *NCL*, *PDLIM5*, and *TMBIM6* are reported to be strongly associated with prostate cancer [176, 216, 229, 294]. *PDLIM5* plays a critical role in developing malignant tumor cells, proliferation and is considered an oncogene for the progression of prostate cancer.

Network-based biomarkers *NTRK2*, *COL1A2*, *JUN*, and *SPARC* are validated through the Cancer Genetics Web. In Chapter 3, we have seen that *COL1A2*, *FN1*, *RPS5*, and *RPS16* are considered to be potential biomarkers. Li et al. [195] have shown that the gene *DCN* is a novel potential biomarker for the prognosis of colon cancer. In the study [180], it has been proved that *RPL12* is overexpressed in cancer samples. *KPNB1* is associated with gastric cancer, cervical cancer, hepatocellular cancer including breast cancer [52]. According to Human Protein Atlas[2], gene *GMPS* and *KARS* are classified as proto-oncogene and disease-related genes, respectively. *RPS4X* is a new predictive and prognostic biomarker of breast cancer as well as ovarian cancer [320]. *RPL9* may play an important role in developing malignant growth in colorectal cancer cells [24]. Therefore, it may be treated as a potential biomarker in breast cancer.

4.8 Discussion

A significant advantage of our method to others except UniBic lies in its capability of discovering several types of biclusters. Another advantage of our algorithm is that it is independent of any normalization technique. There is always a need for new biclustering algorithms which can effectively work for all types of bicluster models. In this thesis, we have proposed a biclustering algorithm called OPBic, to analyze cancer gene and miRNA expression datasets. We compare OPBic with state-of-the-art biclustering algorithms using synthetic and gene expression datasets. It is found that OPBic performs significantly well in all testing scenarios especially outperforming in scenario 2 and in discovering overlapping biclusters. GO enrichment analysis of gene expression data shows that OPBic outperforms all other biclustering algorithms. With this, OPBic is capable of identifying gene and miRNA cancer potential biomarkers. OPBic does not require the number of clusters a priori. But for our own convenience, we have considered the first 100 biclusters to reduce the number of biclusters for further analysis. Despite having significant advantages, OPBic suffers from high computational time complexity. If it is applied to gene expression data with a large number of samples and genes, the number of conditions increases at a greater amount. It actually creates dif-

[2] `https://www.proteinatlas.org/`

ficulty to manage an enormous amount of data even using parallel computing. A modification of this algorithm is highly desirable in order to get significant biclusters in lesser time. Moreover incorporating external biological knowledge (such as GO or pathways) in the analysis of gene expression data has been found to be quite effective. Towards the goal, we present a semi-supervised biclustering algorithm in the next chapter.

5

Semi-supervised Bicluster Analysis of Cancer Transcriptomics Data

Traditional biclustering algorithms use biological knowledge to evaluate the biclusters found and does not use it during the bicluster identification process. In Chapter 4, we have proposed an unsupervised biclustering algorithm to analyze gene and miRNA expression datasets. But that algorithm may not find significant biclusters throughout all the datasets. Therefore, we modify the previous idea of the OPBic algorithm by proposing a semi-supervised biclustering algorithm in this chapter. Inspired from the chapter 3, we have incorporated external biological knowledge in our algorithm to get better quality biclusters. The chapter is organized as follows. The chapter initiates by an introduction in Section 5.1 followed by a related work in Section 5.2. Section 5.3 clearly states the motivation of this particular work. Next, we elaborate our proposed work in Section 5.4. The time complexity of the proposed algorithm with other algorithms in literature is given in Section 5.5. Section 5.6, we evaluate our proposed algorithm with other existing algorithms and show a comparative study with respect to synthetic as well as cancer transcriptomics data. Section 5.7, reports the potential gene and miRNA cancer biomarkers. Finally, the chapter ends with a discussion in Section 5.8.

5.1 Introduction

Gene expression data is usually noisy. To make a fair comparison among genes across a set of samples, it is more relevant to have relative expressions rather than absolutes [348]. To reveal biological knowledge from the input data, we focus on identifying genes with similar patterns across experimental conditions. Therefore, in this work, we are also interested in a pattern-based biclustering algorithm, using pattern similarity. The key goal is to look for Order-Preserving Submatrices (OPSMs), which are considered to be the most biologically meaningful groups as mentioned in Chapter 4. OPSM is a non-contiguous submatrix where expression values of each row increase monotonically, according to the column permutations [348]. Biologically, the subset of co-regulated genes which are active under a subset of conditions are generally order-preserving or order-reversing in nature [337]. Such a submatrix captures the consensus trends of rows over columns, i.e., the tendency of patterns to get priority over the actual values.

This study extends the concept of OPPM which is a well-known problem in computer science, where a given pattern matches the relative order of substring of values for a given text instead of seeking for a fragment of text which exactly coincides with the pattern. Recalling from Chapter 4 as mentioned in Definition 4.4.4, let us, understand OPPM with a diagram as demonstrated in Figure 5.1-A, where the order of *pat* $\{c_6, c_8, c_3, c_7, c_1, c_5, c_4, c_2\}$ exactly matches the order of g_a i.e., $\{c_6, c_8, c_3, c_7, c_1, c_5, c_4, c_2\}$. Further, a variant of the OPPM problem is

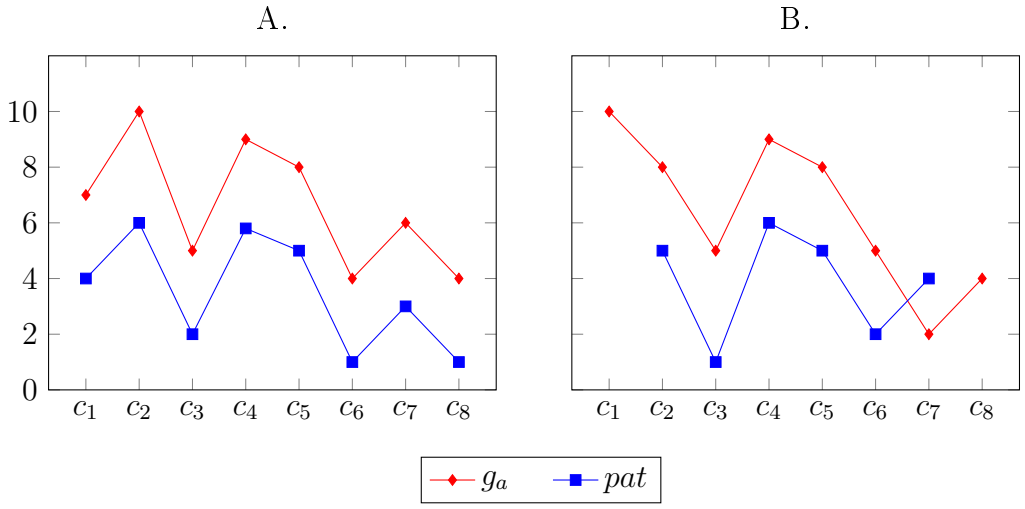


Figure 5.1: An illustration of OPPM. The x-axis denotes the conditions and the y-axis represents the expression values. A. An exact OPPM of *pat* with g_a. B. Approximate OPPM of *pat* with g_a with 1 mismatch at position 6.

considered in [115], which is used in our method. In [115], a generalized version of

OPPM, referred to as Order-Preserving Pattern Matching, with $\mathcal{K}$ mismatches is proposed, relaxing the exact pattern matching to approximate pattern matching.

Definition 5.1.1 *Two sequences (orders)* $X = (x_1, x_2, \ldots, x_n)$ *and* $pat = (y_1, y_2, \ldots, y_{pt})$*, where* $|X| \geq |pat|$ *are said to be an approximate match with* $\mathcal{K}$ *mismatches, denoted by* $(x_1, x_2, \ldots, x_n) \sim^{\mathcal{K}} (y_1, y_2, \ldots, y_{pt})$*, where* $1 \leq \mathcal{K} \leq \frac{pt}{2}$*. This means, the sequence* X *matches pat in a window size of* w *(where* $w = |pat| - \mathcal{K}$*) in the same order, but not necessarily consecutively.*

If we remove the corresponding mismatched elements from both the sequences, the resulting new sequences must be order-isomorphic, see Figure 5.1-B. In this figure, *pat* approximately matches g_a for conditions c_2 to c_6 and one mismatch at condition c_7.

With this conceptual idea, we propose a novel semi-supervised method known as the **P**athway-based **O**rder-**P**reserving **Bic**lustering (POPBic) algorithm incorporating KEGG pathway information. Our algorithm exploits the goodness of both order-preserving biclustering as well as that of pathways. Most of the proposed knowledge-driven algorithms use GO as biological knowledge. Each GO term annotates individual gene products and KEGG pathways are used to annotate the classes of gene products [365]. To the best of our knowledge, no algorithm has attempted to perform biclustering using pathway information. The major contributions of our work are summarized below.

- Effective integration of an order-preserving biclustering algorithm with domain (KEGG pathway) knowledge.
- Generation of artificial datasets to evaluate the effectiveness of any biclustering algorithm.
- Statistical and biological validation of our method using both synthetic and benchmark transcriptomics (cancer) datasets.
- Ability to discover all eight types of biclustering models and to perform consistently well for noisy datasets.
- Identification of both positively and negatively correlated co-expressed patterns.

5.2 Related work

Researchers have adopted various algorithmic strategies to solve the biclustering problem. These can be divided into two broad categories: traditional [227, 266,

277] and knowledge-driven. Firstly, we have highlighted some popular traditional biclustering algorithms in Chapter 2, Section 2.3.3. We have also summarized the algorithms which focus to identify order-preserving submatrices in Chapter 4, Section 4.2.

We now discuss some knowledge-driven biclustering algorithms. Liu et al. [214] have proposed a Smart Hierarchical Tendency Preserving clustering (SHTP-clustering) algorithm using a biclustering model, which directly incorporates GO information into the clustering process. Markus and Wiuf [49] have presented a co-clustering method based on Self-Organizing Maps (SOMs) with GO annotations to extract better biologically significant clusters. Visconti et al. [328] have modified an existing algorithm called ISA and developed Additional Information-Driven Iterative Signature Algorithm (AID-ISA) which used annotated data retrieved from ontologies to refine the search. A scatter-search based biclustering algorithm is presented in [255] that integrated biological knowledge and showed the improvement over classic biclustering algorithms. BiClustering with Constraints using PAttern Mining (BiC2PAM) is able to mine efficient clusters to guide pattern-based biclustering using background knowledge [135]. Recently, Nepomuceno [257] have extended the idea demonstrated in [256] to enhance the quality of biclusters for high dimensional gene expression datasets. A method for inferring biological function for a set of genes with previously unknown function, given a set of genes with well-known information has been explored in [189]. Beyond all these strategies, Dussaut et al. [93] deal with a slightly different problem where a biclustering method along with topological information is used to infer pathway associations. Based on the premise that genes with similar behavior are grouped together, it is possible to detect groups of unknown genes clustered with genes with known biological information (such as GO annotations).

5.3 Motivation

The main motivation behind this proposed algorithm is to steer from unsupervised to semi-supervised biclustering algorithm. Researchers have achieved success by incorporating GO functional annotations during clustering, giving rise to better quality biclusters than traditional biclustering algorithms for gene expression data. Nowadays, integration of external knowledge has become an essential part of bicluster extraction [22]. In Chapter 3, we have proposed semi-supervised clustering algorithms incorporating GO and successfully identified

significant clusters. Recently, it has been demonstrated that pathway-based approaches are more reliable and robust for analysis of gene expression data [365]. This is because gene expression data and protein-protein networks do not yield coherent gene subsets as pathways do [178]. In addition, such gene sets seem to change with individual experiments. Moreover, it has been noticed that one gene may show active involvement in more than one pathway [266]. Pathways provide a stable set of functional relationships with respect to molecular activities such as signalling, metabolic, gene regulations, and protein interactions [55]. According to Kim et al. [178], integrating pathways and gene information improve the performance of semi-supervised learning with the goal of differentiating disease phenotypes. Like the SDC algorithm, our proposed biclustering algorithm is suitable to find positively and negatively co-expression patterns that define the relationship among genes [261] as mentioned in Section 3.1. Most of the existing studies focus on mining only positively expressed patterns and ignore the negative patterns. But biological evidence shows that negatively and positively co-expressed patterns should be grouped in the same cluster [142]. For example, genes *YLR367W* and *YKR057W* of Yeast dataset negatively co-expressed with another gene *YML009C* across eight conditions. It is important to mention that, these genes should be clustered together as they are a part of the protein translation and translocation process. Comparatively less amount of work has been done in order to identify both the patterns in a bicluster [261, 286]. Lastly, it is desirable to discover all eight types of bicluster patterns. To address this issue, POPBic identifies order-preserving submatrices as mentioned in Chapter 4.

5.4 Proposed method

Our method called POPBic utilizes the information provided by KEGG [172] along with gene expression data to detect biclusters. POPBic includes two major steps as described below. It is dependent on four input parameters namely, the minimum number of conditions C_{min}, significance level α, seed selection criteria t_s, and maximum error ϵ to extract the final set of biclusters.

5.4.1 Selection of significant seed genes

Let us consider m genes $G = \{g_1, g_2, \ldots, g_m\}$ of the input matrix and h KEGG pathways $P = \{p_1, p_2, \ldots, p_h\}$. We assemble the information by creating a list of all genes with the known pathways, called the Pathway Gene List (PGL). It has been observed that a gene may or may not participate in one or more

biological pathways. Genes with lower p-value (< 0.05) are selected as significant or enriched genes, which have higher variability. Therefore, for each gene of PGL, we perform the variance analysis or ANOVA test (α is set to be 0.05 or 5%) to identify the significant genes [91]. These significant genes are then inserted in the Significant Gene List (SGL) in ascending order of their p-values. It is assumed that genes with lower variability over samples are considered less significant for bicluster identification [40]. In order to reduce the search space, top t_s genes (according to lower p-value) of the SGL is taken as Significant Seed Gene List ($SSGL$) if $|SGL| \geq \lceil t_S * m \rceil$ otherwise, SGL is taken as $SSGL$. Similar to the study in [91], we also consider top 10% as t_s values for real datasets. The formation of biclusters starts with each of the genes assembled in $SSGL$ with the assurance that a gene with higher variability gets first preference than one with lower variability.

5.4.2 Extraction of biclusters

POPBic algorithm discovers the biclusters in a greedy iterative fashion. POPBic algorithm groups genes based on subsets of conditions using pathway information to select the initiator genes for a bicluster during the biclustering process. We focus on finding similar (positively and negatively co-expressed) patterns under a subset of conditions from the expression data. The POPBic algorithm is presented in Algorithm 3. Next, we explain the proposed approach in detail.

Transforming into order matrix: The POPBic algorithm is initiated by transforming the input data matrix $ED_{m\times n}$ into two order matrices $OM_{m\times n}$ and $OM'_{m\times n}$ (Algorithm 1, line no. 2). The matrix $OM_{m\times n}$ is created by replacing the original values with the column indices of the sorted (ascending order) values of each row as described in Section 4.4.1. A similar method is used to compute $OM'_{m\times n}$ except that ties are broken by replacing the lower value with the higher column index. For better understanding, we include an illustrative example in Tables 5.1, 5.2, and 5.3.

Table 5.1 shows an example expression data matrix $ED_{4\times 6}$ with four rows and six columns and Figure 5.2-A depicts the corresponding patterns for this data matrix. It can be noticed that, if all the four genes and six conditions are considered, then we cannot find any coherent patterns among these genes. From the Figure 5.2-B, we can visualize the similar patterns of genes g_1, g_2, and g_3 (excluding g_4) under conditions c_1, c_2, c_3, c_4, and c_5 (excluding c_6) only, which forms a bicluster showing some common trends. If we have considered all the

Algorithm 3: Extraction of biclusters

Input : $ED_{m\times n}$ with a set of genes $G = \{g_1, g_2, \ldots, g_m\}$ and a set of columns $C = \{c_1, c_2, \ldots, c_n\}$, $SSGL$, C_{min}, ϵ
Output: Bic: A final list of biclusters

$Bic = \phi$
Compute $OM_{m\times n}$ and $OM'_{m\times n}$
for *all* $g_a \in SSGL$ **do**
  $\mathcal{I} = g_a$
  for *all* $g_b \in PGL$ **do**
    if $g_b \neq g_a$ **then**
      Compute $Oscore(g_a, g_b)$
    end
  end
  Determine a gene list G' which has maximum Oscore with g_a
  $Seed^{pat}_{g_a} = \phi$
  for *all* $g_w \in G'$ **do**
    Compute pat_l by LCS between pair of (g_a, g_w)
    if $|pat_l| \geq C_{min}$ **then**
      $Seed^{pat}_{g_a} = Seed^{pat}_{g_a} \cup pat_l$
    end
  end
  Sort all patterns of $Seed^{pat}_{g_a}$ in descending order, arrange $g_w \in G'$ based on longest pattern produced with g_a and cluster expansion starts with new set of genes
  for *all* $pat_l \in Seed^{pat}_{g_a}$ *and* $g_w \in G'$ **do**
    $\mathcal{I} = \mathcal{I} \cup g_w$
    $\mathcal{J} = pat_l$
    for *all* $g_z \in G \setminus \{\mathcal{I}\}$ **do**
      if $OM_{g_z\times n}$ *approximately matches to pattern* pat_l *with* ϵ *error* **then**
        $\mathcal{I} = \mathcal{I} \cup g_z$
        else if $OM'_{g_z\times n}$ *approximately matches to pattern* $Reverse(pat_l)$ *with* ϵ *error* **then**
          $\mathcal{I} = \mathcal{I} \cup g_z$
        end
      end
    end
    if $|\mathcal{I}| \geq |\mathcal{J}|$ **then**
      $\beta \leftarrow \{\mathcal{I}, \mathcal{J}\}$
      Add β to Bic
      break
    else
      $\mathcal{I} = g_a$
    end
  end
end

Table 5.1: An example of an expression data matrix. The table contains four genes and six conditions.

	c_1	c_2	c_3	c_4	c_5	c_6
g_1	3	9	6	9	5	7
g_2	-1	5	2	6	1	9
g_3	-3	-7	-4	-8	-3	-10
g_4	5	-5	-8	2	3	2

conditions, and all genes we may not find bicluster. Rather, we have identified a subset of genes and subset of conditions. Here, g_4 does not show any common trend like other three genes with respect to conditions c_1 to c_5. On the other hand, Figure 5.2-C displays expression values of a bicluster showing their trends across conditions $c_1, c_5, c_3, c_2,$ and c_4.

Table 5.2: Transformed $OM_{m\times n}$ from expression matrix.

g_1	1	5	3	6	2	4
g_2	1	5	3	2	4	6
g_3	6	4	2	3	1	5
g_4	3	2	4	6	5	1

Table 5.3: Transformed $OM'_{m\times n}$ from expression matrix.

g_1	1	5	3	6	4	2
g_2	1	5	3	2	4	6
g_3	6	4	2	3	5	1
g_4	3	2	6	4	5	1

Computation of overlap score: To minimize the computational time as well as the number of biclusters, we limit our search space. A bicluster is initiated with a seed gene g_a ($Seed_{g_a}$) from the list of significant seed genes $SSGL$. In the lines 5-9, the association between two genes, g_a and g_b (where $a, b \in \{1, 2, \ldots m\}$, g_a is seed gene, $g_b \in PGL$ and $g_a \neq g_b$) is determined using an overlap score, referred to as O_{score} [241] which is defined by Equation. 5.4.1. Here, p_{g_a} and $|.|$ represent the set of related pathways for g_a and the number of elements, respectively.

$$O_{score}(g_a, g_b) = \frac{|p_{g_a} \cap p_{g_b}|}{min(|p_{g_a}|, |p_{g_b}|)} \tag{5.4.1}$$

If g_a and g_b share the same pathways then O_{score} is 1, whereas if the genes do not share any pathways, O_{score} value will be 0. To be particular, this is the

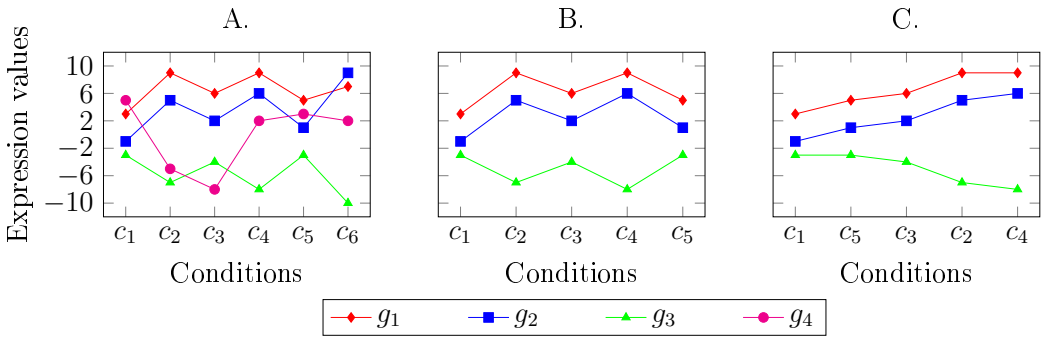


Figure 5.2: A. Original expression matrix mentioned in Table 5.1 with four rows and six columns. B. Positive and negative co-expressed patterns with the maximum allowed mismatch of 1 over five conditions. C. The expression values mentioned in B are arranged in ascending order. In all the figures, the x-axis denotes the conditions and the y-axis represents the expression values.

normalized term overlap estimating similarity which is originally proposed in the work [241]. It is worth mentioning that if any of the genes do not share any pathways then the overlap score is directly considered to be 0. Moreover, it is quite easy to understand that a high overlap score indicates a high number of common pathways shared between two genes. The concept of overlap score computation is visualized in Figure 5.3. The figure depicts the Venn diagram of associated pathways of two genes *PRPS1* and *FBP1*. The pathways associated with *PRPS1* and *FBP1* are shown by blue and green color, respectively. The common pathways shared by both the genes are shown in red color. The overlap score between these two genes is 0.67. Next, we consider those genes whose

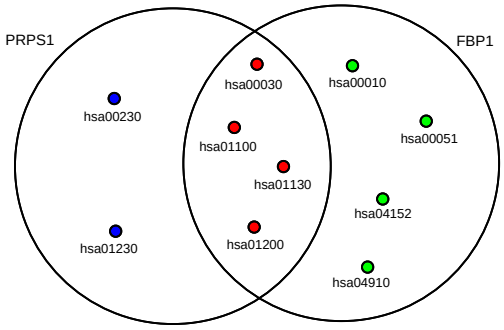


Figure 5.3: The venn diagram of associated pathways of two genes: *PRPS1* and *FBP1*.

overlap score is maximum among all gene pairs. Therefore, we get a list of genes $G' \subseteq PGL$ that share the maximum number of pathways with the seed gene g_a which signifies the maximum O_{score} in line no. 10. After this, bicluster expansions proceed as given in line (11-38) of Algorithm 1.

Identification of seed pattern: We compute the LCS between a pair of genes (g_a, g_w) where $g_a \in SSGL$, $g_w \in G'$, G' is a gene list having maximum overlap scores with g_a, $G' \subseteq PGL$ and $g_a \neq g_w$. Determining the LCS (Longest Common Subsequence) for a given pair of sequences is considered a traditional problem in sequence processing [70].

Definition 5.4.1 *Given two sequences g_a and g_w, where $g_a \in SSGL$ and $g_w \in G'$, the condition pattern pat_l is the maximum set of matching conditions between g_a and g_w which is obtined by LCS(g_a, g_w) which gives the LCS between $OM_{g_a \times n}$ and $OM_{g_w \times n}$ and $|pat_l| \geq C_{min}$.*

In order to obtain the maximal set of conditions (condition pattern) $pat = \{c_1, c_2, \ldots, c_n\}$, keeping in mind the definition of a bicluster, we apply LCS between pairs of rows from the order matrix $OM_{m \times n}$. The fundamental concept in applying the LCS algorithm is built on the observation about the presence of a common subsequence between two rows, which occur in an order-preserving submatrix. LCS can be illustrated by an example. Suppose, the order of two genes are $R_{g_1} = \{1, 5, 3, 6, 2, 4\}$ and $R_{g_2} = \{1, 5, 3, 2, 4, 6\}$. $\{1, 5, 3, 2, 4\}$ is a common subsequence of both R_{g_1} and R_{g_2} of maximum length. Hence, the LCS between these two ordered sequences is $\{1, 5, 3, 2, 4\}$.

It is important to note that positive and negative co-expressed patterns are opposite. An opposite pattern (negative or positive) can easily be calculated by reversing the given pattern $Reverse(pat)$ (positive or negative). If a given pattern is $\{4, 2, 3, 5, 1\}$, the opposite pattern is $\{1, 5, 3, 2, 4\}$.

Each condition pattern obtained for g_a is listed in the seed pattern i.e., $Seed_{g_a}^{pat} = \{pat_1, pat_2 \ldots, pat_L\}, L = |G'|$ and length of each pattern i.e., $|pat_l| \geq C_{min}$, where $l = \{1, 2, \ldots L\}$ otherwise, we exclude those patterns. We place a longer pattern before a smaller one in $Seed_{g_a}^{pat}$ i.e., $|pat_1| \geq |pat_2| \ldots \geq |pat_L|$. The reason for considering the longest patterns is to fulfill the goal of finding bicluster i.e., maximum sized biclusters. It is also worth mentioning that the entire process repeats with the next gene from $SSGL$ till all seed genes from $SSGL$ are exhausted (line no. 3-38) of Algorithm 3.

Lemma 5.4.1 *$Seed_{g_a}^{pat}$ can have a maximum of L condition patterns where $L = |G'|$.*

Proof: Let $g_a \in SSGL$ be seed gene and $Seed_{g_a}^{pat}$ contains all condition patterns obtained for g_a according to definition 5.4.1. Suppose, $Seed_{g_a}^{pat} = \{pat_1, pat_2, \ldots, pat_L\}$, i.e., L patterns are obtained with respect to g_a. Moreover,

each pat_l is obtained by LCS(g_a,g_w) for each $g_w \in G'$, and $|G|' = L$. Another important condition of definition 5.4.1 is $|pat_l| \geq C_{min}$. Therefore, it might happen that for some $l = \{1, 2, \ldots, L\}$ LCS$(g_a, g_w) < C_{min}$. Those g_w genes will not be inserted into $Seed_{g_a}^{pat}$. Thus, $Seed_{g_a}^{pat}$ has condition patterns $\{pat_1, pat_2, \ldots, pat_x\}$, where $x \leq L$. Therefore, $Seed_{g_a}^{pat}$ can have at most L condition patterns.

Mining OPPM genes: The biclustering process starts with two initiator genes (g_a, g_w), where $g_w \in G'$ and g_w has the largest LCS with respect to g_a. Bicluster expansion now proceeds by adding genes from $G \setminus \{g_a, g_w\}$ having OPPM with the pat in $Seed_{g_a}^{pat}$. To add genes, we formulate the generalized version of the biclustering problem for extracting biclusters as follows. Given an expression matrix $ED_{m \times n}$ and the bicluster $\beta_{\mathcal{I} \times \mathcal{J}}$, a submatrix of ED, can be extracted by identifying the subset of rows $\mathcal{I} \subseteq G$ under a subset of columns $\mathcal{J} \subseteq C$, where each $g_i \in \mathcal{I}$ is order-preserved with pattern $\mathcal{J}$, and each of the conditions $c_x \in \mathcal{J}$ with maximum error ϵ calculated as in Equation 5.4.2, where $\mathcal{K}$ is the maximum number of mismatches allowed. The main goal of the POPBic algorithm is to identify both the positive and negative expression patterns for a particular subset of conditions. In order to identify more than one bicluster, we need to extract biclusters for different subsets of conditions.

$$\epsilon = \frac{\mathcal{K}}{|\mathcal{J}|} \tag{5.4.2}$$

Each pattern $pat_l \in Seed_{g_a}^{pat}$, where $l = \{1, 2, .., L\}$ triggers the formation of an initial bicluster $\beta_{\mathcal{I} \times \mathcal{J}}$, a subset of genes $\mathcal{I} = \{g_a, g_w\}$, and a subset of columns $\mathcal{J} = \{pat_l\}$, where LCS of (g_a, g_w) corresponds to pat_l. The present step focuses on identifying the co-expressed patterns (positive as well as negative) on the basis of the seed pattern. We further extend our initial bicluster by adding a new gene g_z at a time from $OM_{g_z \times n}$, considering the seed pattern with the help of OPPM with maximum ϵ error, where $g_z \in G \setminus \{\mathcal{I}\}$. If g_z is not included in the first check, it can be added next using the reverse seed pattern in a similar fashion, as mentioned before from $OM'_{g_z \times n}$. We keep on adding new genes (may be positive or negative) into the bicluster until all genes are considered once (line no. 22-29). A bicluster β is treated as a final one if it satisfies the criteria $|\mathcal{I}| \geq |\mathcal{J}|$ [337] (line no. 30-36) else further search for another pat_l from $Seed_{g_a}^{pat}$ is to be continued until the list is exhausted (line no. 19-38). It has been observed that the real gene expression data have a larger number of genes than the experimental conditions. To resemble the same property, we have kept the stopping criteria as $|\mathcal{I}| \geq |\mathcal{J}|$ [337]. It is noteworthy to mention that once we get a bicluster from

a seed gene we will not search for all the pat_l because the algorithm is greedy by nature and also we try to reduce the time complexity. The entire process is repeated (line no. 3-38) until all the genes in $SSGL$ are visited once to get a list of biclusters Bic. The POPBic algorithm outputs a maximum of $|SSGL|$ number of biclusters because, for each significant seed gene, only one bicluster is identified. Optionally, the algorithm removes the biclusters with more than a user-specified threshold where we keep larger biclusters and eliminate the smaller ones.

Lemma 5.4.2 *A bicluster $\beta_{\mathcal{I}\times\mathcal{J}}$ has the maximum condition pattern with respect to g_a satisfying the condition $|\mathcal{I}| \geq |\mathcal{J}|$.*

Proof: Let, the bicluster be $\beta_{\mathcal{I}\times\mathcal{J}}$, where $\mathcal{I}$ is subset of genes and $\mathcal{J}$ denotes subset of columns. Let, the seed pattern $Seed_{g_a}^{pat}$ has maximum of L number of condition patterns according to lemma 5.4.1. Initially, it is assumed that $\mathcal{I} = \{g_a\}$ and $\mathcal{J} = \phi$. Before identifying the actual bicluster, the $Seed_{g_a}^{pat}$ is sorted, where $|pat_1| \geq |pat_2| \ldots \geq |pat_L|$. The identification of bicluster starts with largest pattern pat_1. Therefore, the subset of genes and subset of columns are $\mathcal{I} = \{g_a, g_w\}$ and $\mathcal{J} = \{pat_l\}$, respectively. Genes are added (either positively or negatively co-expressed) in $\mathcal{I}$ from the remaining genes $G \setminus \{\mathcal{I}\}$. The entire gene list G is checked once and then the condition $|\mathcal{I}| \geq |\mathcal{J}|$ is verified. Thus, a bicluster $\beta_{\mathcal{I}\times\mathcal{J}}$ is found with respect to g_a with maximum number of genes and maximum number of conditions as condition pattern itself is largest with respect to g_a as $Seed_{g_a}^{pat}$ is sorted in descending order. If the bicluster $\beta_{\mathcal{I}\times\mathcal{J}}$ is identified for pat_1 then the search has not proceed further for pat_2 and so on. Hence the proof.

5.5 Time complexity

Brute force biclustering algorithms that do not have heuristics have been found to be computationally intractable and are considered an NP-hard problem [3]. Therefore, it is highly challenging to develop an effective and efficient heuristic solution. We estimate the algorithmic efficiency of POPBic in terms of its computational time complexity. Let, $ED_{m\times n}$ be the input data matrix, where m is the number of genes and n the number of columns, $\bar{S}$ be the number of significant seed genes, $\bar{P}$ be the number of PGL, and $\bar{G}$ be the number of genes which has maximum O_{score} with a seed gene $Seed_{g_a}$. POPBic takes $O(m(n \log n))$ and $O(m(n \log n))$ time to create $OM_{m\times n}$ and $OM'_{m\times n}$ matrices, respectively. The

algorithm takes $O(\bar{S}\bar{P})$ time to compute O_{score}, $O(\bar{S}\bar{G}n^2)$ for LCS identification, $O(\bar{S}(\bar{G}\log\bar{G}))$ to sort all the patterns of $Seed_{g_a}^{pat}$ $\forall g_a \in SSGL$. Computing OPPM with $\mathcal{K}$ mismatches can be done in $O(\bar{S}\bar{G}m(n(\log\log n + \mathcal{K}\log\log\mathcal{K})))$ time as in [115] to compute the OPPM with $\mathcal{K}$ mismatches. So the overall time complexity is $O(m(n\log n) + m(n\log n) + \bar{S}\bar{P} + \bar{S}\bar{G}n^2 + \bar{S}(\bar{G}\log\ \bar{G}) + \bar{S}\bar{G}m(n(\log\log n + \mathcal{K}\log\log\mathcal{K})))) \approx O(\bar{S}\bar{G}n(m+n))$.

We compare the time complexity of POPBic algorithm, with other state-of-the-art methods. Let, l the number of models, K the number of biclusters, $\wp$ time to compute sequential pattern mining task, $\bar{r}\bar{s}$ average size of the biclusters and q separation percentage parameter. Then, the time complexity of OPSM, UniBic, FABIA, and BicSPAM algorithms are $O(mn^3l)$, $O(q^2m^2n^2)$, $O(nl^2m)$, and $\theta(min(\binom{m}{2}, n)\wp + \binom{K}{\frac{K}{2}}\bar{r}\bar{s})$, respectively.

5.6 Performance analysis

For systematic evaluation, we compare POPBic with five other state-of-the-art methods OPSM [32], QUBIC [196], FABIA [141], BicSPAM [134], and UniBic [337] on both synthetic and real datasets. The POPBic algorithm is implemented on MATLAB 2016 platform. Please refer to Appendix for GUI of the POPBic and user manual. We also use a JAVA based tool named Biclustering Analysis Toolbox (BicAT) Plus [5] for OPSM, **Bic**lustering based on PAttern Mining Software (BicPAMS) [132] for BicSPAM, C implementation [337] for UniBic algorithm, and R BiclustGUI package [77] for FABIA and QUBIC. Since POPBic uses biological pathways for real datasets, however, it is not possible to incorporate biological knowledge for synthetic datasets. For simplicity, we exclude this biological knowledge from the algorithm and use all the significant rows rather than a subset of rows for further computation. We set $\alpha = 0.05$ to determine $SSGL$. All significant rows are considered seed for synthetic data and inserted into $SSGL$. Similar to the related work in [337], we also set the minimum number of conditions to $C_{min} = \lceil 5\% * |C| \rceil$. The value of ϵ is chosen based on exhaustive experimentation. We execute POPBic with (i) $0 \leq \epsilon \leq 0.7$ and (ii) step size of 0.05 and report the best possible biclusters in terms of relevance and recovery. In addition to this, a lower ϵ value is a good choice for selection. For synthetic datasets, we keep the larger biclusters and remove the smaller ones which have an overlap of more than 0.25 as in [278]. For BicSPAM and UniBic, we follow the parameter settings mentioned in Section 4.6.1. The usual meaning of parameters used in OPSM, QUBIC, and FABIA algorithms are shown in Table 5.4.

We use parameters such as the number of biclusters, the minimum number of

Table 5.4: Meaning of each parameters for different biclustering algorithms.

Method	Implementation used	Parameters	Meaning	Year
FABIA [141]	R [170]	K	No. of biclusters	2010
		cyc	No. of iterations	
		alpha	Sparsness lodingd	
		spl	Sparsness prior loading	
		spz	Sparsness factors	
		random	Random initial loading	
		scale	Scale loading vector	
		lap	Minimum value of variational	
		nL	Maximum biclusters for row	
		lL	Maximum rows per bicluster	
		bL	Cycle starts	
		non_negative	Non negative factors and loadings	
		norm	Normalization	
		center	Data centering	
OPSM [32]	Bicat Plus	p_m	Partial model	2003
QUBIC [196]	R [170]	*report.no*	Maximum biclusters	2009
		tolerance	Tolerence	
		filter.proportion	Redundant proportion	

rows and the minimum number of columns by providing the true values as per synthetic dataset creation [266]. Whenever possible we keep the parameter settings as per original authors contribution. The parameter setting for FABIA is cyc=500, alpha=0.01, spl=0, spz=0.5, random=1, scale=0, lap=1, nL=0, lL=0, bL=0, non_negative=0, norm=1, and center=2. For, real datasets the number of biclusters of FABIA is kept as 13. For OPSM, the p_m value is set to 100 for both types of datasets. The algorithm QUBIC is set to tolerance=0.95 and filter.proportion=1 for synthetic and real datasets. Moreover the number of biclusters is 100 for real datasets.

5.6.1 Synthetic datasets generation

The first set of datasets collected is used to recover the best-suited model for competing biclustering algorithms. In this phase, we use a total of 80 ar-

tificial datasets, 10 for each of the eight models: constant, column-constant, row-constant, up-regulated, additive, multiplicative, additive-multiplicative, and trend-preserving [98, 337]. We implant one bicluster of size 70×40 and follow the definition of each of the eight bicluster types with no noise and no overlap. The generation of synthetic datasets is mentioned in Chapter 4, Section 4.6.1.

It is often found that gene expression data is perturbed by noise due to errors in the experimental setup. This leads to the need for a robust algorithm that can work on noisy data. Therefore, we further test the resistance of POPBic to noise using the second dataset collection. We prepare the artificial dataset by introducing noise in the previous additive-multiplicative, multiplicative, additive, and trend-preserving synthetic datasets without any overlap. Random noise is drawn from normal distributions with μ 0 and varying σ such as 0.05, 0.1, 0.15, 0.2, and 0.25 is added to each expression value in the matrix. So, we get 5 different noisy test matrices from one synthetic dataset. This procedure is repeated 10 times. Thus, we obtain 50 different synthetic test matrices for each of the bicluster models. Hence, we obtain a total of 200 datasets considering four different bicluster types.

The third set of datasets collected comprises overlapping biclusters. We create the datasets with three overlapped biclusters of overlapping size 0×0, 3×3, 6×6, and 9×9. The biclusters are generated by replacing the selected submatrix of size 40×15 with additive-multiplicative, additive, multiplicative, and trend-preserving biclusters in the background matrix of size 500×50. We repeat the procedure 10 times. Thus, we obtain a total of 160 datasets. A summary table of synthetic datasets are available in Table 5.5.

5.6.2 Performance on synthetic datasets

We use *MS* in terms of relevance and recovery as described in Equation 2.3.30 of Chapter 2. In this study, we experiment on three scenarios (model, noise, and overlap), giving rise to a total of 440 synthetic datasets.

Evaluation on bicluster models

The input parameter C_{min} is 3 for synthetic datasets of the POPBic algorithm. The selected values of ϵ for additive-multiplicative, additive, multiplicative, column-constant, constant, row-constant, trend-preserving, and up-regulated are 0.4-0.45, 0, 0-0.05, 0, 0, 0, 0-0.2, and 0.1-0.15, respectively. Figure 5.4 illustrates the average relevance and recovery scores among all the test matrices for each biclustering algorithm. POPBic outperforms all other competing algorithms in

Table 5.5: Summary of synthetic datasets.

Exp	#	Size	Bicluster types	Extra	Total
Model	1	70×40	Constant, column-constant, row-constant, up-regulated, additive, multiplicative, additive-multiplicative, trend-preserving	No overlap no noise	80
Noise	1	70×40	Additive, multiplicative, additive-multiplicative, trend-preserving	Noise allowed (0.05, 0.1, 0.15, 0.2 and 0.25)	200
Overlap	3	40×15	Additive, multiplicative, additive-multiplicative, trend-preserving	No noise, overlap allowed (0×0, 3×3, 6×6, 9×9)	160

Descriptions: Exp- Experiment, #- Number of implanted bicluster(s), Size: Size of bicluster(s), Total: Number of generated synthetic datasets.

recovering the true additive-multiplicative biclusters with an average recovery score of 0.75 compared to the second-highest recovery score of 0.73 for OPSM, and average relevance score of 0.42 compared to the highest relevance score of 0.6 for FABIA. POPBic and OPSM return spurious biclusters which might be the reason for the low relevance score. The result shows that POPBic and UniBic have a remarkable advantage for additive biclusters by reaching the maximum average relevance and recovery scores (1, 1) followed by OPSM with (0.59, 1). UniBic successfully identifies constant, row-constant, and up-regulated biclusters with both average relevance and the recovery scores are close to (1, 1) whereas POPBic obtains (0.99, 0.99) for constant and row-constant types, and (0.94, 0.94) for up-regulated biclusters. POPBic gives comparatively less significant recovery values for trend-preserving than UniBic and OPSM, but it still gives the second-highest average relevance score after UniBic. UniBic outperforms others for column-constant and multiplicative biclusters by scoring (1, 1) as average relevance and recovery scores, whereas POPBic returns (0.97, 0.97) and (0.98, 0.98), respectively.

BicSPAM and FABIA perform best on column-constant, additive, and trend-preserving biclustering models. The comparison in Figure 5.4 demonstrates that QUBIC, FABIA, and UniBic give the same average relevance and recovery scores because the number of biclusters given as input is based on true biclusters. OPSM shows better performance for all types of models except for the average relevance score. QUBIC shows the best performance on up-regulated biclusters and also can identify column-constant and multiplicative biclusters. POPBic

Table 5.6: The selected values of ϵ for POPBic algorithm in the presence of noise.

Noise	Types of biclusters			
	Additive-multiplicative	Additive	Multiplicative	Trend-preserving
0.05	0.4-0.45	0.25-0.35	0.4-0.5	0.2-0.4
0.10	0.45	0.3-0.45	0.4-0.45	0.25-0.45
0.15	0.4-0.45	0.35-0.45	0.45	0.35-0.45
0.20	0.35-0.45	0.4-0.45	0.25-0.45	0.4-0.45
0.25	0.25-0.45	0.4-0.45	0.45	0.3-0.45

successfully identifies constant, column-constant, row-constant, additive, multiplicative, and additive-multiplicative bicluster models and performs slightly less than UniBic for up-regulated and trend-preserving types.

Robustness of POPBic in the presence of noise

The input parameters for noisy data is kept same as before. To evaluate the robustness of POPBic algorithm we select additive-multiplicative, additive, multiplicative, and trend-preserving models for experiment. Due to unavailability of biclusters, we cannot give the relevance and recovery score for some of ϵ values. The selected values of ϵ for the noisy synthetic datasets are given in Table 5.6.

In the previous section, it has been clearly demonstrated that POPBic performs satisfactorily for all eight bicluster types. Extraction of additive-multiplicative, multiplicative, and trend-preserving models is considered to be the most challenging task [3, 277, 337]. In addition to this, we also experiment with additive bicluster. Therefore, we consider these four models for studying the influence of noise on the performance of all selected biclustering algorithms. The experimental results for noisy data are displayed in Figure 5.5 with the average relevance and recovery scores. Our algorithm exhibits robust performance in the presence of noise, indicating an average recovery score above 0.88 for the additive model and outperforms all other biclustering algorithms across all noise levels. POPBic has been performing well in comparison to its other competing algorithms for noisy multiplicative and trend-preserving models in terms of recovery score. POPBic also recovers the additive-multiplicative model from noisy data, but it slightly falls in performance for higher levels of noise compared to OPSM.

FABIA shows robust performance throughout all noise levels and does not change much. Although UniBic shows good performance for datasets without noise, its match score is affected at higher levels of noise. The performance of QUBIC goes down for noisy data. POPBic can find nearly true biclusters when

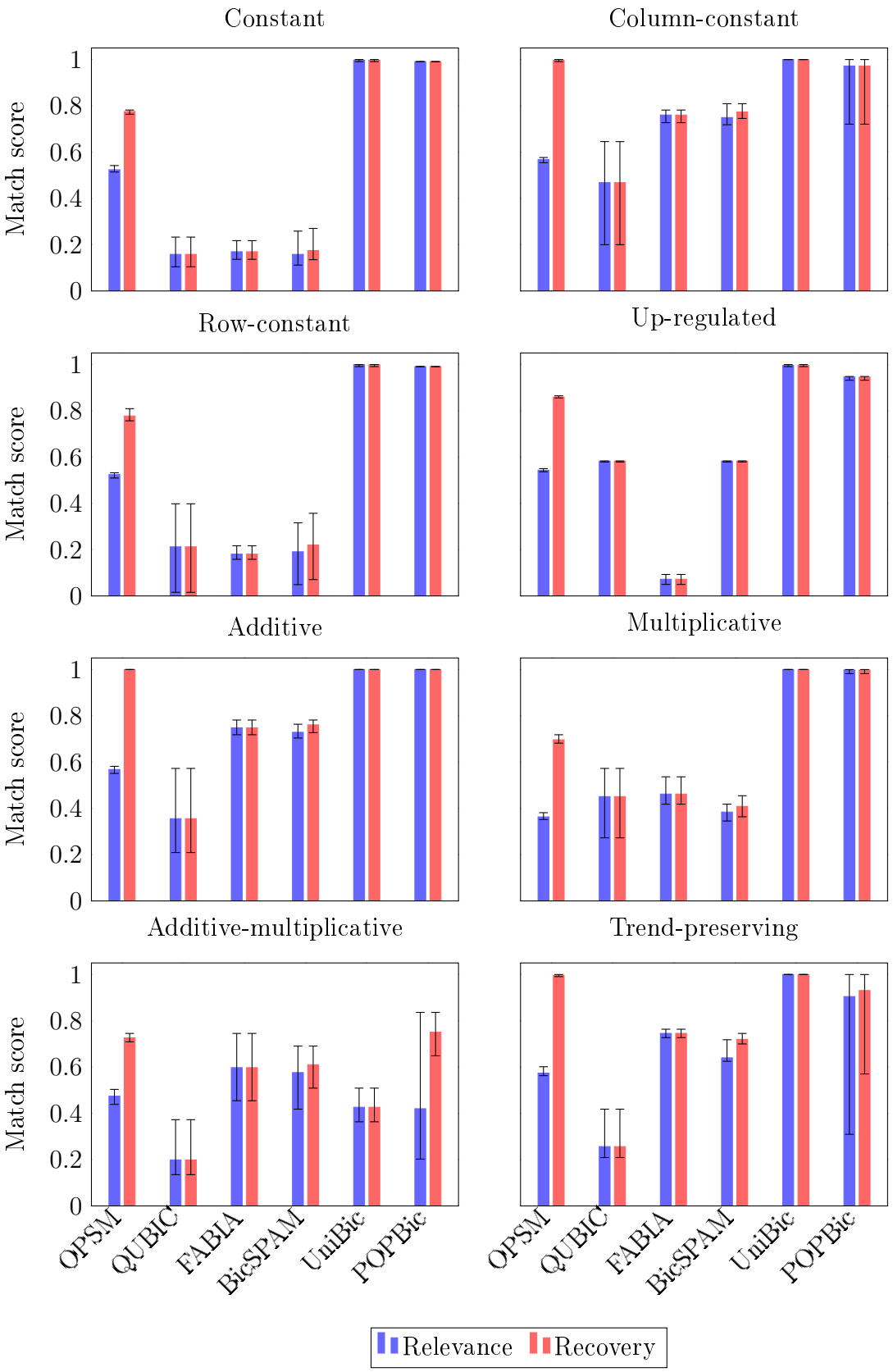


Figure 5.4: Relevance and recovery scores with error bars (range) of different biclustering algorithms on eight different biclustering models.

there is no noise. The match score gradually degrades with higher levels of noise, although it can still find most implanted biclusters. OPSM shows a more robust nature than BicSPAM. Taking all bicluster types together, it can be said that POPBic substantially outperforms with respect to recovery scores than other biclustering algorithms for noisy data.

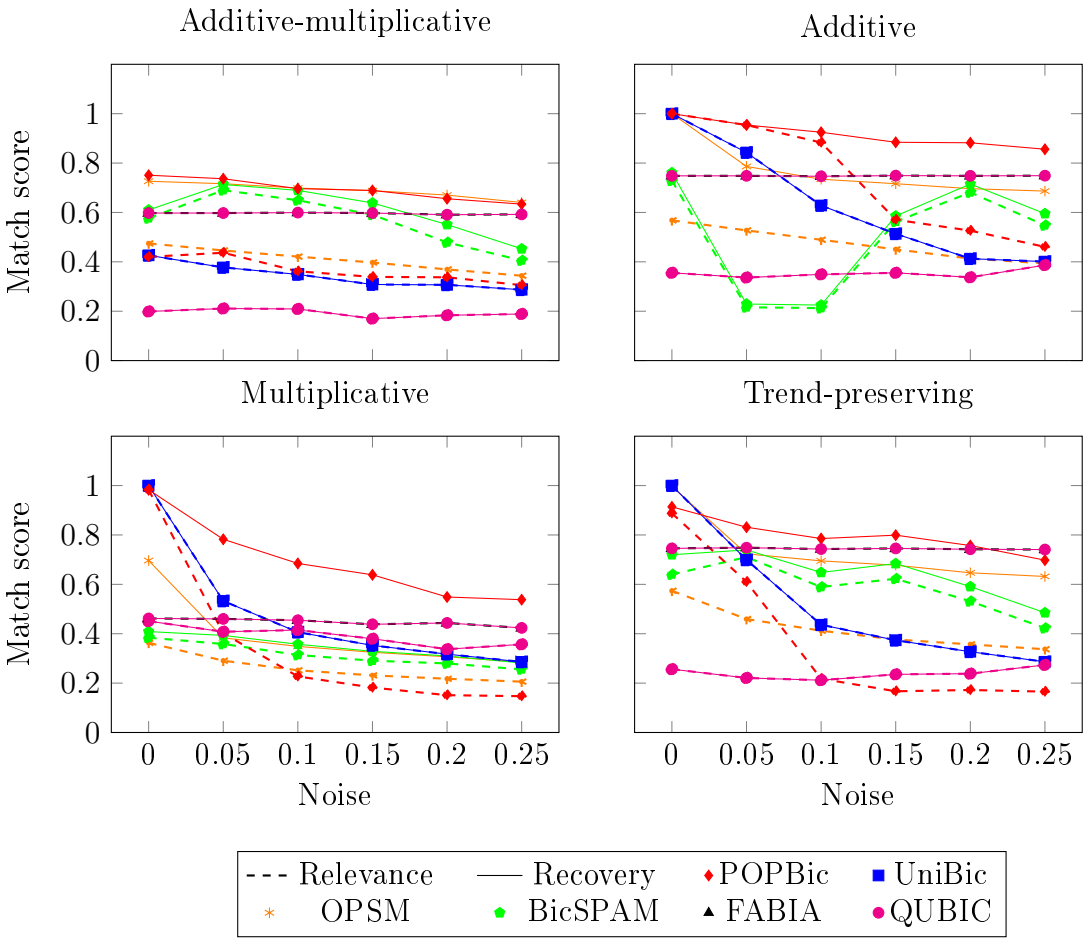


Figure 5.5: Relevance and recovery scores of different biclustering algorithms on four models over noise scenario.

Overlapping bicluster detection

We additionally compare POPBic with the five biclustering algorithms on overlapping datasets. Here, also we choose the four most challenging bicluster types for our experiment i.e., additive, multiplicative, additive-multiplicative, and trend-preserving as in the previous section. The average relevance and recovery scores of six algorithms are shown in Figure 5.6 for each type of overlapping test matrices. The input parameters for the overlapping data matrix is kept the same as the model experiment. The selected values of ϵ for overlapping data are

Table 5.7: The selected values of ϵ for POPBic algorithm in the overlapping scenario.

Overlap	Types of biclusters			
	Additive-multiplicative	Additive	Multiplicative	Trend-preserving
0×0	0.2-0.4	0-0.2	0.2-0.45	0.3-0.45
3×3	0.15-0.35	0-0.15	0.2-0.4	0.2-0.45
6×6	0.2-0.4	0-0.1	0.25-0.35	0.25-0.4
9×9	0-0.35	0-0.15	0.2-0.35	0.1-0.45

reported in Table 5.7.

Most algorithms go down when the overlapping degree is increased. The worst performer for this particular testing scenario is QUBIC; its initial score is also very low. Our algorithm is quite able to recover biclusters when the overlap is not allowed for all models. POPBic reaches the highest relevance and recovery scores (close to 1) for higher levels of degrees (up to 6×6) to detect additive biclusters. But for overlapping degree 9×9, POPBic's performance degrades more in the recovery score than OPSM and UniBic, but it still achieves the highest relevance score near 1 compared to 0.8 for UniBic. In the case of additive-multiplicative biclusters, POPBic retrieves biclusters with higher recovery scores than other algorithms, but it slightly deteriorates compared to OPSM for higher overlap degrees. For the other two models, such as trend-preserving and multiplicative, UniBic exceeds all competing methods. This analysis demonstrates that POPBic works well for additive-multiplicative and additive models for overlapping biclusters.

5.6.3 Performance on real datasets

To compare the effectiveness of the POPBic algorithm with all other algorithms, we further experiment on four cancer microarray gene expression datasets [76]. Table 5.8 summarizes the datasets used in this experiment. The KEGG [172] pathways associated with each gene are downloaded from the web-server DAVID[1] and in Table 5.8 the number of unique pathways for each of the datasets are reported.

The input parameters of POPBic for the real dataset are the same as those for synthetic datasets except for ϵ which is fixed after simulation. Based on the experimentations on artificial data, we observe that the performance of the algorithm degrades for values above 0.5. Therefore, we execute POPBic with ϵ values in the range $0.1 \leq \epsilon \leq 0.5$, with a stepwise increment of 0.05. From each

[1] `https://david.ncifcrf.gov/home.jsp`

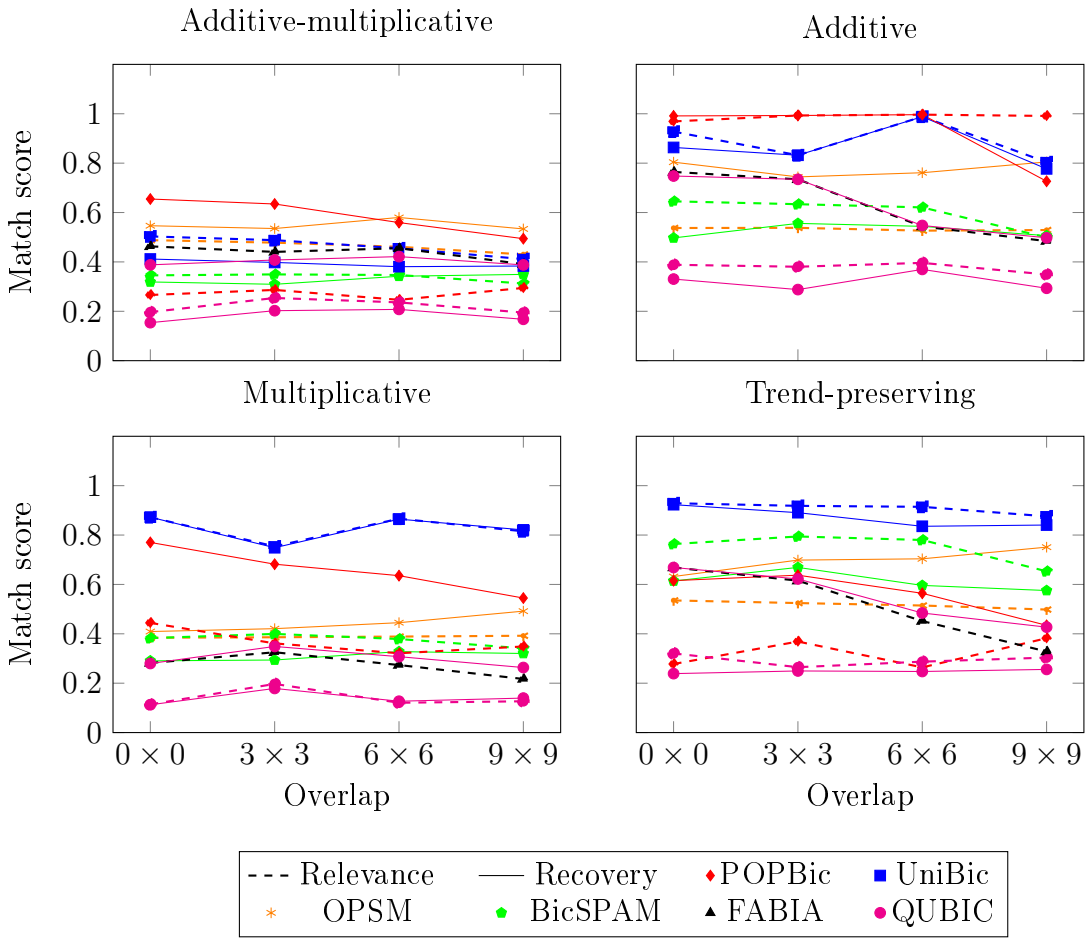


Figure 5.6: Relevance and recovery scores of different biclustering algorithms on four models over overlapped biclusters.

Table 5.8: Four cancer gene expression datasets.

Dataset	Tissue	Pathways
Armstrong-v2	Blood	288
Bhattacharjee	Lung	285
Laiho	Colon	283
Singh	Prostate	204

setting, we obtain the p-value of each bicluster and compute the percentage of enriched biclusters. Higher percentages of enrichment and lower p-values (near 0) indicate a better result. The criterion to report the best possible bicluster is to satisfy both the criteria of lower p-value and more than 90% enrichment of biclusters (according to Biological Process). The biclusters extracted by different biclustering algorithms are evaluated based on GO annotations.

Statistical analysis

The POPBic algorithm obtains a total of 326 biclusters from microarray gene expression datasets (Table 5.9). The statistical relevance of the obtained biclusters from POPBic is highly satisfactory with respect to coverage as mentioned in [21, 311]. In this chapter, we use two criteria: Gene coverage and Condition coverage to quantify the abilities of biclustering algorithms. The properties of obtained biclusters using different biclustering algorithms are reported in Table 5.9 in association with the largest sized and smallest sized biclusters. We can clearly observe that POPBic covers on average 82.18% of genes and 99.75% of conditions, taking all the datasets together. OPSM returns 84.43% of genes and 56.04% of conditions. Therefore, POPBic can generate biclusters with higher coverage of the data matrix than other biclustering algorithms. Amongst all, BicSPAM shows poor gene and condition coverages of 20.36% and 30.35%, respectively. FABIA and UniBic coverages are 99.28% and 92%, respectively of the conditions, whereas Gene coverage for both the algorithms is relatively low.

Enrichment analysis

The biological assessment of identified biclusters from real datasets is carried out using functional enrichment analysis. The key goal is to determine whether the genes of each generated bicluster are significantly enriched or not with respect to the GO annotations. Due to the unavailability of the true biclusters, we use GO enrichment analysis to evaluate the biclusters, demonstrating how well genes can match with different GO categories. To achieve this purpose, we use a web-based tool called FuncAssociate [35] for genes. The tool computes the p-values of each of the biclusters. A bicluster is considered to be enriched if p-values of all the annotation terms are less than the significance cut-off value. In Table 5.9, we report the GO terms which correspond to the lowest p-value. The result suggests that POPBic is able to determine biclusters with high biological relevance.

The performance of biclustering algorithms can be evaluated by the percentage of enriched (one term or more than one terms) biclusters to the total number of extracted biclusters as shown in Equation 4.6.3. Tables 5.10, 5.11 and 5.12 present the aggregated results considering all the four gene expression datasets for 5%, 1%, 0.5%, and 0.1% level of significance for three different domains BP, MF, and CC, respectively. POPBic seems to outperform all other biclustering algorithms for all the datasets, the values are shown in all three Tables 5.10, 5.11, and 5.12 with boldface. POPBic shows an edge over its competitors obtaining a higher percentage of enriched biclusters out of 326 biclusters

for BP as well as other domains for the different significance levels. BicSPAM outputs 21 biclusters and achieves the second-highest enrichment level for BP and CC. Even though the strongest competitor of POPBic algorithm, viz. UniBic has shown good performance for synthetic data, it could give only 62.28% (5% level of significance) enrichment rate for real datasets. OPSM and FABIA identify lower numbers of biclusters, 83 and 51, respectively with smaller percentages of enrichment rates. OPSM retrieves the lowest p-value but QUBIC has 63.5% (5% level of significance) as enrichment score with 254 enriched biclusters out of 400 biclusters. In addition, we also compare the performance of algorithms by computing the total number of unique significant annotation terms. This evaluation demonstrates that a higher number of significant terms has a better functional grouping. This is depicted in Figure 5.7. We can clearly see that on average POPBic has a better number of enriched terms than any other algorithm.

Biological study

The resultant biclusters obtained from the blood cancer dataset by POPBic algorithm, have been analyzed from the biological point of view [257]. The biological study focuses on 55 enriched biclusters for Biological Processes out of 56 biclusters. Among these biclusters, all of the 55 biclusters are associated with cancer-related genes which have been investigated by Network of Cancer Genes (NCG) [13].

For each of these enriched biclusters, we find the percentage of cancer-related genes and select bicluster 33 for further analysis since it has the highest percentage, due to its small size. Out of 80 genes (with official gene ids), 49 genes were annotated with various pathways (including cancer-related pathways such as *proteoglycans in cancer* inferred from DAVID [144], *Toll-like receptor signaling pathway*, *pathways in cancer*, and *MAPK pathway* validated from GeneAnalytics [31]). Among the genes unannotated with any pathways, we found that several of them are related to various cancers based on literature. Those genes are *DFNA5*, *MAGEA3*, *S100A4*, *CASP2*, *CCT6B*, *CHD1L*, *CLUAP1*, *DTNA*, *EMP1*, *GZMM*, *GBP1*, *HOXA4*, *HOXA5*, *HOXB2*, *IFIT3*, *MYT1L*, *NKTR*, *NFIB*, *NFIX*, *PMP2*, *PRTN3*, *RTEL1*, *RARG*, *SCN2B*, *SLC17A4*, *SNTB1*, and *TES*. Among these genes, *PRTN3* is a leukemia cancer gene inferred from GeneAnalytics, although it has not been annotated with any of the known pathways. This is important since we can now infer that even an unannotated gene belonging to a bicluster might be related to genes responsible for a particular disease sharing the highest number of pathways.

Table 5.9: Comparison of quantitative measure of obtained biclusters from real datasets.

						Maximum		Minimum				
Dataset	Algorithm	#Bics	Gc	Cc	C_{avg}	#G	#C	#G	#C	GO ID	Name	p-value
Armstrong-v2	OPSM	32	87.65	86.11	86.88	1153	3	2	52	GO:0007165	signal transduction	**1.70E-33**
	QUBIC	100	40.29	86.11	63.20	93	9	26	7	GO:0051239	regulation of multicellular organismal process	1.23E-10
	FABIA	13	24.34	98.61	61.48	76	28	27	15	GO:0044422	organelle part	2.76E-08
	BicSPAM	4	22.70	23.61	23.15	301	10	208	8	GO:0007166	cell surface receptor signaling pathway	9.56E-10
	UniBic	100	6.34	**100**	53.17	20	57	12	53	GO:0051088	PMA-inducible membrane protein ectodomain proteolysis	1.92E-06
	POPBic	56	**96.08**	**100**	98.04	696	53	39	32	GO:0048518	positive regulation of biological process	1.38E-26
Bhattacharjee	OPSM	23	**96.89**	26.11	61.50	1001	3	14	25	GO:0042221	response to chemical	**1.40E-29**
	QUBIC	100	33.51	37.44	35.47	248	9	46	15	GO:0097458	neuron part	7.26E-09
	FABIA	12	40.12	98.52	69.32	172	66	18	21	GO:0002376	immune system process	1.02E-22

Continuation of Table 5.9

Dataset	Algorithm	#Bics	Gc	Cc	C_{avg}	Maximum		Minimum		GO ID	Name	p-value
						#G	#C	#G	#C			
	BicSPAM	2	14.13	11.82	12.98	189	17	120	14	GO:0044421	extracellular region part	5.18E-13
	UniBic	100	10.56	69.46	40.01	40	116	35	99	GO:0005179	hormone activity	5.94E-13
	POPBic	69	58.91	**100**	89.76	169	125	30	23	GO:0005615	extracellular space	2.83E-18
Laiho	OPSM	13	66.44	75.68	71.06	1375	2	2	19	GO:0030198	extracellular matrix organization	9.85E-22
	QUBIC	100	35.65	91.89	63.77	71	4	26	2	GO:0031012	extracellular matrix	1.32E-21
	FABIA	13	46.68	**100**	73.34	166	18	46	10	GO:0030198	extracellular matrix organization	8.96E-25
	BicSPAM	14	11.58	70.27	40.93	48	6	39	6	GO:0030198	extracellular matrix organization	1.68E-17
	UniBic	68	**100**	**100**	100	1818	5	6	9	GO:0032501	multicellular organismal process	9.34E-20
	POPBic	183	**100**	**100**	100	1710	10	21	21	GO:0031012	extracellular matrix	**8.15E-32**
Singh	OPSM	15	86.73	36.27	61.50	209	4	2	20	GO:0006614	SRP-dependent cotranslational protein targeting to membrane	3.28E-73

Continuation of Table 5.9

Dataset	Algorithm	#Bics	Gc	Cc	C_{avg}	Maximum		Minimum		GO ID	Name	p-value
						#G	#C	#G	#C			
	QUBIC	100	92.33	78.43	85.38	158	10	13	4	GO:0006614	SRP-dependent cotranslational protein targeting to membrane	3.31E-72
	FABIA	13	51.33	100	75.66	48	34	1	33	GO:0006614	SRP-dependent cotranslational protein targeting to membrane	1.68E-28
	BicSPAM	1	33.04	15.69	24.36	112	16	112	16	GO:0019058	viral life cycle	1.33E-51
	UniBic	74	**100**	**100**	100	218	7	6	13	GO:0019058	viral life cycle	9.04E-70
	POPBic	18	73.75	99.02	86.38	75	38	26	26	GO:0006614	SRP-dependent cotranslational protein targeting to membrane	**1.33E-88**

Abbreviations: #- Number of, Bics- Biclusters, Gc- Gene coverage, Cc- Condition coverage, C_{avg}- Arithmetic mean of Gene and Condition coverages, G- Genes, C- Conditions, Gc and Cc are in %.

Table 5.10: GO enrichment analysis result of different biclustering algorithms on real datasets based on Biological Process.

Algorithm	Found	Enriched biclusters			
		5%	1%	0.5%	0.1%
OPSM	83	67.47%	66.27%	63.86%	53.01%
QUBIC	400	63.5%	43.5%	39%	34%
FABIA	51	50.98%	29.41%	25.49%	23.53
BicSPAM	21	90.48%	85.71%	76.19%	66.67%
UniBic	342	62.28%	56.43%	52.63%	43.86%
POPBic	326	**96.32%**	**89.57%**	**88.04%**	**83.44%**

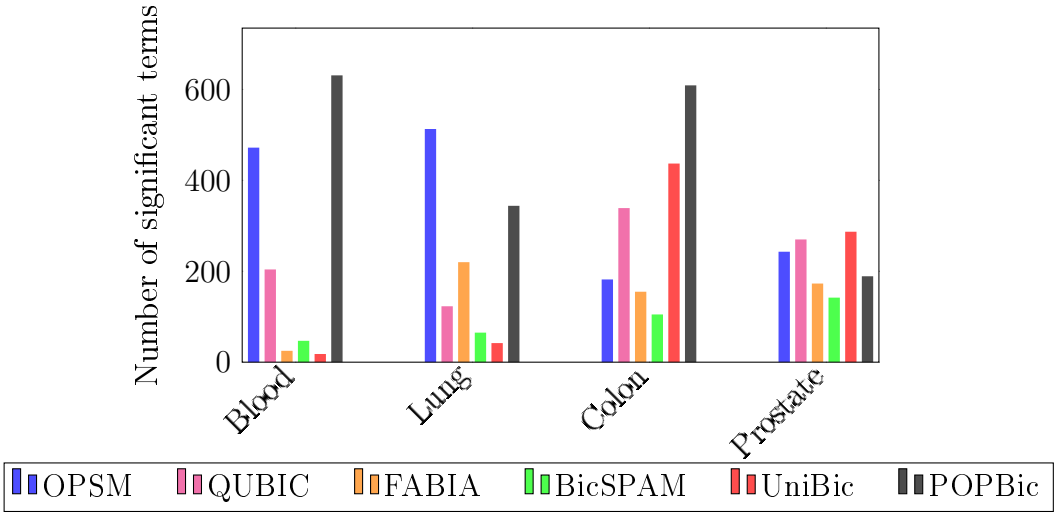


Figure 5.7: The number of enriched terms (shown in y-axis) by six different methods (shown in x-axis) for different datasets.

5.6.4 Results of miRNA breast cancer data

Now, the proposed algorithm has been applied to the miRNA dataset as described in Chapter 4, Section 4.6.4 in order to study its performance. The KEGG pathways associated with each miRNA is downloaded from the Web-server miRWalk2.0[2]. The number of pathways used for the miRNA dataset is 198. The parameters of the POPBic algorithm for this miRNA breast cancer dataset are as follows: 0.5 for ϵ and 10 for the minimum number of conditions C_{min}. POPBic algorithm obtains a total of 38 biclusters from the miRNA dataset. The obtained result is compared with other methods like OPSM, QUBIC, FABIA, BicSPAM, and UniBic. BicSPAM algorithm does not identify any biclusters for the miRNA dataset, therefore it is excluded from our further analysis.

Table 5.13 reports the percentage of enriched biclusters based on BP, MF,

[2]http://zmf.umm.uni-heidelberg.de/apps/zmf/mirwalk2/holistic.html

Table 5.11: GO enrichment analysis result of different biclustering algorithms on real datasets based on Molecular Function.

Algorithm	Found	Enriched biclusters			
		5%	1%	0.5%	0.1%
OPSM	83	51.81%	48.19%	44.58%	39.76%
QUBIC	400	43.25%	30.5%	28%	22.5%
FABIA	51	31.37%	17.65%	15.69%	15.69%
BicSPAM	21	57.14%	47.62%	47.62%	38.10%
UniBic	342	48.83%	44.74%	44.15%	40.35%
POPBic	326	**80.06%**	**74.23%**	**71.17%**	**64.11%**

Table 5.12: GO enrichment analysis result of different biclustering algorithms on real datasets based on Cellular Component.

Algorithm	Found	Enriched biclusters			
		5%	1%	0.5%	0.1%
OPSM	83	60.24%	54.22	51.81%	50.60%
QUBIC	400	39.25%	37%	34%	30.25%
FABIA	51	31.37%	21.57%	17.65%	15.69%
BicSPAM	21	76.19%	71.43%	71.43%	66.67%
UniBic	342	40.35%	28.65%	25.73%	22.81%
POPBic	326	**88.96%**	**82.82%**	**81.60%**	**78.53%**

and CC of different biclustering algorithms. In this scenario, we use DIANA mirPath v.3 [329] to compute p-value considering the level of significance as 0.05. For the miRNA dataset, QUBIC outperforms all biclustering algorithms for BP, MF, and CC. Whereas, POPBIC can only give 100% enriched biclusters for CC only. In the case of MF, it is better than OPSM and FABIA. For BP the performance of POPBic degrades than OPSM, QUBIC, and UniBic. On the other hand, we have provided the lowest p-value from the resulting biclusters of different algorithms in Table 5.14. We can observe that POPBic, UniBic, and OPSM give the lowest p-value which is 0. Hence, we can say that POPBic can identify significant biclusters.

5.7 Potential biomarkers identification

We apply network-based and frequency-based biomarker identification methods as described in Chapter 3 and 4, respectively to detect potential biomarkers. The parameter settings is similar to Sections 3.7 and 4.7. In Table 5.15, we report identified biomarkers of different cancers from the biclusters obtained by POPBic algorithm. The biomarkers are verified through different scholarly pub-

Table 5.13: Enrichment analysis result of different biclustering algorithms on miRNA dataset.

Algorithm	Found	Enriched biclusters					
		BP		MF		CC	
		T	%	T	%	T	%
OPSM	17	16	94.12	16	94.12	16	94.12
QUBIC	100	100	**100**	100	**100**	100	**100**
FABIA	13	4	30.77	4	30.77	4	30.77
UniBic	100	99	99	99	99	99	99
POPBic	38	27	71.05	36	94.74	38	**100**

Table 5.14: Comparative analysis of POPBic algorithm with other methods on miRNA datasets.

Algorithm	GO ID	Name	p-value
OPSM	GO:0043226	organelle	0
QUBIC	GO:0043226	organelle	1.51E-312
FABIA	GO:0043226	organelle	1.67E-38
UniBic	GO:0043226	organelle	0
POPBic	GO:0043226	organelle	0

Table 5.15: Potential biomarkers identification using POPBic algorithm.

Method	Dataset	Potential biomarkers
Frequency	Armstrong-v2	*WFS1*, *MFHAS1*, *PLCH1*, *MYLK*
	Bhattacharjee	*RIMBP2*, *RTN1*
	Laiho	*SULF1*, *POSTN*
	Singh	*RPS5*, *RPL12*, *RPS17*, *RPLP0*, *RPL14*, *RPS14*, *RPL19*, *PPIB*, *RPL11*
	miRNA	*hsa-miR-1185-1-5p*, *hsa-miR-1185-2-5p*, *hsa-miR-410*, *hsa-miR-654*
Network	Armstrong-v2	*FYN*, *AUTS2*
	Bhattacharjee	*CPS1*, *PCSK1*
	Laiho	*COL1A2*, *FN1*
	Singh	*RPS23*, *RPS3A*, *RPS15A*
	miRNA	*hsa-miR-335-5p*, *hsa-miR-27a*

lications, Cancer Genetics Web[3], and The Human Protein atlas[4]. As mentioned in The Human Protein atlas, frequency-based biomarker genes *WFS1*, *MFHAS1*, *MYLK*, *RPS17*, and *PPIB* are considered to be diseased genes. *RIMBP2* is highly expressed in lung squamous cell carcinoma (LUSC) and also has poor prognosis [347]. This type of cancer i.e., LUSC is closely related with smoking. *SULF1* is occurred in higher frequency commonly in three types of cancers, such as breast, colon, and central nervous system than other normal tissue [327]. It is also reposted in several studies that *SULF1* is up-regulated in lung carcinoma, pancreatic cancer, gastric cancer, and hepatocellular carcinoma [327]. In Chapter 3, we have seen that *RPS5* and *RPL12* are considered to be potential biomarkers. *RPL19* is a potential prognostic indicator of prostate cancer as well as in malignant epithelia [29]. Previously, in breast cancer the overexpression of *RPL19* is described in breast cancer [90]. *RPL11* is indirectly responsible for reducing translation to tumor suppressors (p27 and TP53) and which in turn promote cancer development [90]. POPBic algorithm identifies *hsa-miR-1185-1-5p, hsa-miR-1185-2-5p, hsa-miR-410*, and *hsa-miR-654* miRNA frequency-based biomarkers. Among these miRNAs, *hsa-miR-410* is a potential biomarker as reported in Chapter 4. *hsa-miR-654* acts as tumor suppressor in breast cancer [8]. In many other publications it has been revealed that *hsa-miR-654* works as tumor suppressor in papillary thyroid cancer [8].

Genes *COL1A2*, *FN1*, and *CPS1* which are identified by network-based method are also reported in Chapter 3. Researchers have identified *FYN* as candidate biomarkers in chronic myelogenous leukemia (CML) [361]. In several studies, *AUTS2* may be implicating in ALL, shows the higher expression in cDNA samples of patients that normal cell ALL than the normal mononuclear cells [263]. Some publications have demonstrated the altered expression of *PCSK1* on lung cancer [81]. Network-based method recognizes *hsa-miR-335-5p* and *hsa-miR-27a* miRNA as potential biomarkers. With evidence, it is proved that *hsa-miR-27a* has a vital role in tumor development, proliferation, apoptosis, polymorphisms, and tumorigenesis [204]. *hsa-miR-27a* is also served as tumor suppressor or oncogene in multiple cancer types such as breast cancer, bladder cancer, colon cancer, pancreatic cancer, and hepatocellular carcinoma. Apart from this, it is effective in cancer management, drug sensitivity, and patients prognosis. *hsa-miR-335-5p* is considered as a promising biomarker in breast cancer and it is downregulated in breast cancer cells [114]. It is also downregulated in renal cell carcinoma and considered as a therapeutic target [114].

[3] `http://www.cancerindex.org`
[4] `https://www.proteinatlas.org`

5.8 Discussion

In this study, we have proposed a pattern-based subspace clustering algorithm for high dimensional data, which incorporates external biological knowledge from KEGG pathways. The proposed POPBic algorithm (i) extracts biclusters of high statistical significance and high biological relevance, (ii) identifies potential biomarkers, (iii) performs consistently well for both synthetic and real datasets, and (iv) performs satisfactorily even in presence of noise. A notable advantage of the POPBic algorithm is that it does not affect the biclustering result even if the input data is not normalized. To evaluate the performance of the biclustering algorithm, we have used both synthetic and real datasets. The synthetic datasets generated by us can be found at `http://agnigarh.tezu.ernet.in/~rosy8/Bioinformatics_RPBic_Data.rar`. The experiment with synthetic data suggests that it can identify several biclustering models properly.

POPBic algorithm is dependent on the proper selection of ϵ value which is being chosen experimentally. As the ϵ value is increased results are obtained in a short span of time with increasing recovery score and decreasing relevance. However, higher values of ϵ might not give the order-preserved patterns. Our focus in this work is recovery score and therefore, we use ϵ value $<= 0.5$. The POPBic algorithm does not perform well in overlapping scenarios, especially for trend-preserving biclusters. The main reason for performance degradation is the seed selection which may not be able to capture the trend-preserving pattern as trend-preserving does not follow any mathematical formula. It also achieves higher performance for noisy data than any other competing biclustering algorithms in terms of recovery score. The biological relevance of generated biclusters has been verified with the help of GO. The obtained result confirms that POPBic generates biclusters of high biological significance. Cancer subtype identification is currently gaining popularity and it would be interesting to extend this work by studying the possibility of determining the homogeneity of condition subsets as a basis towards attaining this goal. As future work, we aim to develop a parallel version of POPBic on Py-CUDA to make it scalable and to provide a method that will identify the error rate automatically. Further, the work will go on for further refinement of the results of POPBic in a differential co-expression analysis framework to enable the identification of relevant disease biomarkers for some cancer type(s). It has been seen that recent research in subspace gene clustering has steered towards Triclustering which takes the third dimension into account instead of just 2 dimensions as was the case in biclustering.

6

Semi-supervised Tricluster Analysis of Cancer Gene Sample Time Data

Triclustering as an emerging research topic has achieved a lot of attention to identifying genes exhibiting similar behavior or co-expressed under a subset of samples or experimental conditions across time points from Gene Sample Time (GST) data. This type of coherent tricluster is useful to elucidate the information about different phenotypes, potential genes associated with these phenotypes and their regulations [316]. In Chapter 2, we have discussed the details of triclustering algorithms, their types and evaluation measures. We have proposed biclustering algorithms in Chapter 4 and 5, which are restricted to two dimensions whereas triclustering extends the concept of biclustering to the temporal domain using GST data. In chapter 5, it has been observed that the semi-supervised biclustering algorithm gives more significant biclusters than unsupervised methods. Motivated by the incorporation of biological knowledge in the biclustering process, we propose a novel semi-supervised triclustering algorithm, named **P**athway-based **O**rder-**P**reserving **Tric**lustering algorithm (POPTric) for 3D data. Similar to POPBic, we guide POPTric using KEGG pathway information. To the best of our knowledge, KEGG pathway information has not been incorporated in triclustering algorithm till the writing of this thesis. This chapter is organized as

follows. In Section 6.1, we start the chapter with an introduction. Then related work is reported in Section 6.2. The motivation of this particular work is explained in Section 6.3. Section 6.4 demonstrates the proposed method in detail. The time complexity of the POPTric algorithm is analyzed in Section 6.5. The evaluation of the POPTric algorithm with other existing algorithms is discussed in Section 6.6. Next, we report the potential biomarkers in Section 6.7. We end this chapter with a discussion in Section 6.8.

6.1 Introduction

Traditional clustering essentially means a grouping of genes that are co-expressed under the full set of experimental conditions or grouping of experimental conditions over the full set of genes based on some proximity measure. However, subspace clustering is found to be more successful and biologically meaningful than full-space clustering in many of the applications because full-space clustering fails to capture the local pattern of the data. To analyze, 3D GST data we further proceed for 3D subspace clustering i.e., triclustering. Triclustering algorithms aim to find the coherent clusters that are similar across genes, conditions, and time points. Genes are clustered under a subset of conditions and a subset of time points.

Since 2005, plenty of triclustering algorithms have been proposed in the context of GST data in the last sixteen years. One of the challenges of triclustering algorithms is to identify all types of patterns (please refer Section 2.4) such as additive, multiplicative, and additive-multiplicative patterns by a single algorithm. To the best of our knowledge, such type of algorithm is not yet present. Therefore, there is always a need for good triclustering algorithms which can identify different types of patterns from the dataset. To address this issue, we propose a novel triclustering algorithm, POPTric which handles this issue nicely.

Being motivated by the POPBic algorithm in Chapter 5, propose our triclustering algorithm POPTric based on the concept of order-preserving. This concept is a well-established problem in mathematics that has been tackled in the past for gene expression data [316]. The crucial task of our proposed algorithm POPTRic is to identify order-preserving submatrices which are in strictly monotonically increasing order using an OPPM algorithm. Our algorithmic approach is particularly interested in seeking a fragment of text which is order-isomorphic to the given pattern. Let the two patterns X and pat of length n be order-isomorphic, and $pat[a] \leq pat[b]$ if and only if $X[a] \leq X[b]$ for all

$a, b = \{1, 2, \ldots, n\}$ [182]. A further variant of OPPM can be generalized in OPPM, with $\mathcal{K}$ mismatches which relaxes the exact pattern matching to approximate pattern matching [115]. The details of the fundamental concepts can be found in Chapter 5, Section 5.1.

In this work, we propose a semi-supervised triclustering algorithm guided by the KEGG pathway. A biological pathway can be defined as a series of interactions among molecules leading to change certain products in a cell [93]. Pathways are the most important key factors to understand the biological functions of a group of genes and phenotypic changes of the patients [341]. Therefore, we incorporate pathway information in our method to obtain groups of genes to show functionally and biologically relevant clusters. A lot of research that use GO, KEGG, protein-protein interaction network, and genome-wide binding data have been reported in [255]. Luque-Baena et al. have proposed a KEGG-improved evolutionary strategy incorporating KEGG information in gene feature selection method showing better results than the classical one [223]. Mallavarapu et al. have proposed a pathway-based deep clustering technique that is successfully used in the identification of cancer subtypes [341]. The salient contributions in this work are

- Incorporation of KEGG information in order-preserving triclustering.
- Generation and validation of 210 synthetic datasets to establish the efficiency of POPTric.
- Statistical and biological validation of real 3D gene expression data.
- Ability to discover three types of tricluster patterns.

6.2 Related work

In this section, we survey existing triclustering algorithms. These algorithms can also be categorized according to whether they use external knowledge [194] or not [37, 165, 364]. In Chapter 2, we have reviewed several unsupervised triclustering algorithms in detail.

Now, we revisit triclustering algorithms depending on the ability to recover different types of triclusters. TriCluster [364] algorithm can mine arbitrary, overlapped triclusters having constant, multiplicative as well as additive patterns. The algorithm ignores intertemporal coherence and is highly dependent on parameters. Following the approach of TriCluster, several versions were designed

to mine clusters, such as gTRICLUSTER [165] and ParTriCluster [16]. gTRICLUSTER does not take care of tricluster patterns rather it focuses on more biologically significant clusters than TriCluster. On the other hand, ParTriCluster detects additive-multiplicative patterns. Xu et al. [346] develop a novel pattern-based triclustering algorithm to mine additive-multiplicative co-regulation patterns. Another pattern-based triclustering algorithm is proposed by Wang et al. [330] to discover time-delayed clusters (td-cluster). This model attempts to capture coherent genes under different subsets of dimensions when genes follow time-delayed relationships. Algorithms Intersected Coexpressed Subcube Miner (ICSM) [4] and OPTricluster [316] identify order-preserving submatrices. ICSM can detect perfect additive triclusters whereas OPTricluster discovers constant and coherent patterns. Bhar et al. [37] have proposed δ-TRIMAX to identify perfect shifting triclusters. Moreover, TriWClustering [79], Trigen [126], and EMOA-δ-TRIMAX [38] algorithms also identify perfect additive patterns.

A good number of algorithms have been mentioned in Section 2.4 which do not use biological knowledge at the time of clustering process, rather external knowledge is used in the evaluation of the identified clusters. Limited work has been done on semi-supervised triclustering algorithms. The algorithm proposed in [194] is a semi-supervised learning. Authors have defined regulation expression values by incorporating gene regulatory information with gene expression data. In addition to the heuristic TRI-clustering algorithm, they have also proposed the Automatic Boundary Searching (ABS) algorithm for calculating the boundary threshold automatically. The algorithm is data source specific, therefore it does not work for GST data as well as does not find additive-multiplicative patterns.

6.3 Motivation

Unlike unsupervised machine learning approaches for gene expression data, recently it has been realized that integration of various open access data in the field of clustering or classification outperforms traditional algorithms and enhance the spurious information present in the omics data [255]. From a biological point of view, biological databases such as GO and KEGG have played an important role in the context of triclustering algorithms for judging the quality of triclusters [37, 38]. Biological knowledge is used as posterior criteria to ensure the relevancy of the discovered clusters. The active involvement of semi-supervised learning approaches have led to their gaining popularity in the field of clustering [225, 244, 307, 325] and biclustering [135, 214, 255, 328]. Li and Tuck [194]

have proposed a triclustering algorithm that integrates gene expression and gene regulatory information for clustering. Integration of any biological information from the open-access databases is a challenging task and is currently one of the most prominent research directions [103].

Various complex relationships exist among genes. One such relationship is negative correlation i.e., one gene shows high expression value for a condition whereas another gene shows low expression value for that condition and vice versa [346]. Not much attention has been given to capturing negative correlations by the existing subspace clustering algorithms. It is also being claimed that positively and negatively correlated genes are grouped to perform a biological activity [88]. Both positive and negative are very much important for identifying effective phenotypes [240]. In Figure 2.7-B, all three genes show additive-multiplicative patterns and g_c shows a negative correlation with the other two genes because of the negative scaling factor. To address this problem, we propose a novel triclustering algorithm that is capable of identifying all of these patterns including negatively co-expressed genes.

6.4 Proposed method

The objective of our subspace clustering is to identify 3D clusters from GST which can cope with the noisy nature of the data. This study also aims intuitively to identify positively and negatively co-expressed genes having the minimum number of samples S_{min} and the minimum number of time points T_{min} from GST $\mathcal{D}$ matrix. Figure 6.1 illustrates the basic functionality of POPTric algorithm.

Definition 6.4.1 *The Tricluster Diffusion (TD) score of a tricluster j is the ratio of Mean Square Residue MSR_{3D} to the volume of j^{th} tricluster as given in Equation 6.4.1, where $|X^j|$, $|Y^j|$, and $|Z^j|$ are the number of genes, the number of samples, and the number of time points of a tricluster, respectively.*

$$TD^j = \frac{MSR^j_{3D}}{|X^j| * |Y^j| * |Z^j|} \tag{6.4.1}$$

A lower TD score of a resultant tricluster indicates a better quality tricluster. The size of a tricluster should be large and MSR_{3D} (Equation 2.4.5) value should be lower to have better coherence in tricluster.

Definition 6.4.2 *Bicluster Diffusion (BD) score of a bicluster j can be defined by the ratio of Mean Square Residue MSR_{2D} (Equation 2.3.9) of the bicluster to the volume of j^{th} bicluster as given in Equation 6.4.2, where $|X^j|$ and $|Y^j|$ are the number of genes and the number of samples of a bicluster, respectively.*

$$BD^j = \frac{MSR^j_{2D}}{|X^j| * |Y^j|} \tag{6.4.2}$$

The tricluster identification process is divided into four major steps.

(i) Identification of significant seed genes considering 3D matrix.

(ii) Creation of order matrix for $G \times S$ matrix for each time plane t_z, where $z = \{1, 2, \ldots, v\}$

(iii) Mining of biclusters from $G \times S$ matrix for each time plane.

(iv) Mining of triclusters by merging biclusters on the time dimension of input data.

The parameters used in this algorithm are significant cut-off α, number of clusters K, the minimum number of samples S_{min}, the minimum number of time points T_{min}, an error-tolerant threshold ϵ, an overlapping threshold θ, and a merging threshold $\bar{\delta}$ to get a list of triclusters $Tric$. The steps involved in our proposed algorithm POPTric is described below in detail. The algorithm is outlined in Algorithm 4. The algorithm is implemented in MATLAB and the source code is available at `http://agnigarh.tezu.ernet.in/~rosy8/Bioinformatics_POPTric_Code_share.rar`

6.4.1 Significant seed gene identification

In its very first step, triclustering algorithm identifies significant seed genes utilizing the KEGG pathway information. We download a functional annotation table from a web-based tool DAVID where we get the KEGG pathway information for each of the given input genes fed into the web-server [144]. Let, there be m number of genes $G = \{g_1, g_2, \ldots, g_m\}$ in a GST data and h number of unique pathways $P = \{p_1, p_2, \ldots, p_h\}$ identified from DAVID. Therefore, we find out genes (G') that are involved in at least one biological pathway for the seed gene identification step. Thereafter, we convert input 3D GST data into 2D data (GS for each time point) and assume all the samples in a particular time point as one group. Hence, we have v number of groups which is useful to compute the p-value of all the genes in G' using variance analysis or ANOVA test, where α is set to 5% or 0.05. It is assumed that less significant genes have lower variability over samples for tricluster identification [40]. Next, we apply K-means algorithm to find K numbers of gene cluster $CL = \{cl_1, cl_2, \ldots, cl_K\}$ from $G \times (S \times T)$ data taking all time points together. After that we identify genes G'' from each

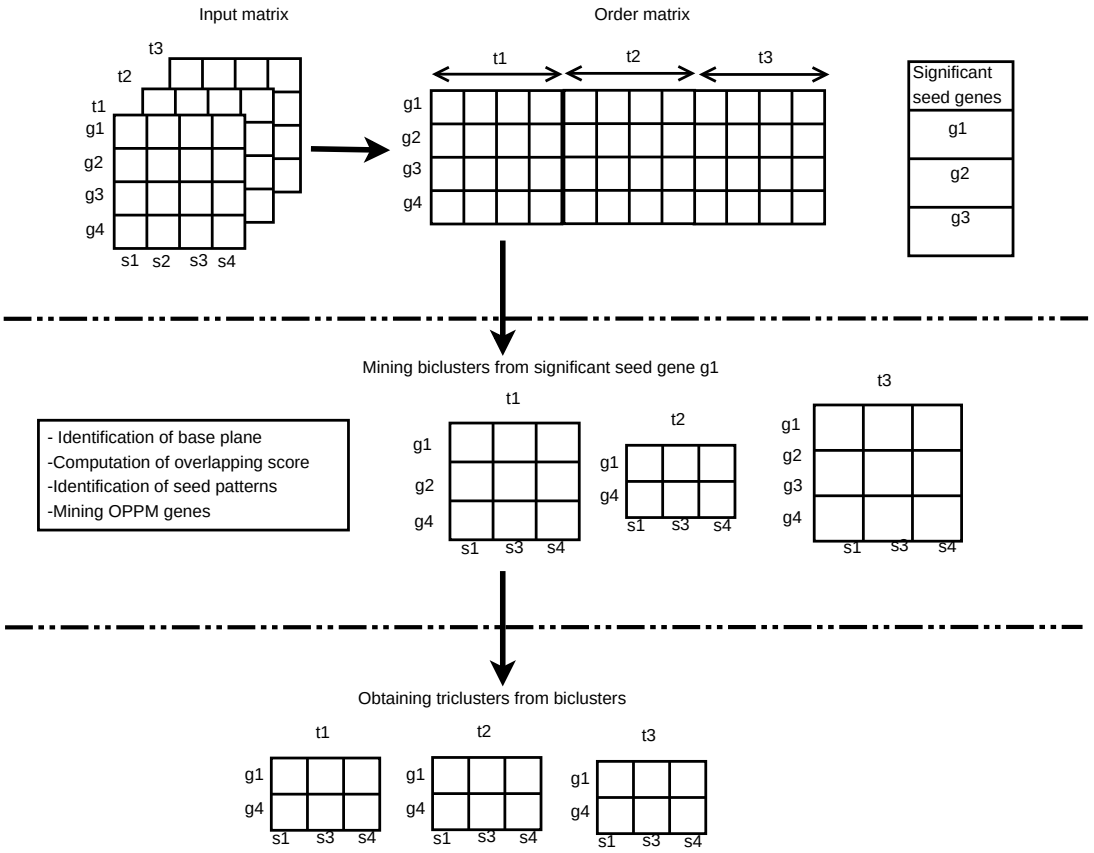


Figure 6.1: Schematic diagram of POPTric algorithm.

of the $cl_{c'}, c' = \{1, 2, \ldots K\}$, clusters with lowest p-value (< 0.05) as significant seed genes according to definition 6.4.3. Therefore, the number of seed genes can not be greater than K i.e., $|G''| \leq K$. The graphical depiction is represented in Figure 6.2.

Definition 6.4.3 *A gene g_a is said to be a significant seed gene if it satisfies the following three conditions. (i) g_a must be associated with any biological pathway. (ii) g_a must reside in a cluster, say $cl_{c'}$ and the p-value of the ANOVA test should be less or equal to significant cut-off i.e., 0.05. (iii) The p-value of gene g_a must be lowest ($<$0.05) among all other genes residing in $cl_{c'}$.*

6.4.2 Creation of order matrix

POPTric initiates by converting the gene expression 2D data i.e., GS data plane for each of the time points into two order matrices OM_{GS} and OM'_{GS}. The creation of two order matrices is the same as mentioned in Section 5.4.2 for $OM_{m \times n}$ and $OM'_{m \times n}$ matrices. In this way, we can compute order matrices for all T time points to get $OM_{G \times (S \times T)}$ and $OM'_{G \times (S \times T)}$. For a better understanding,

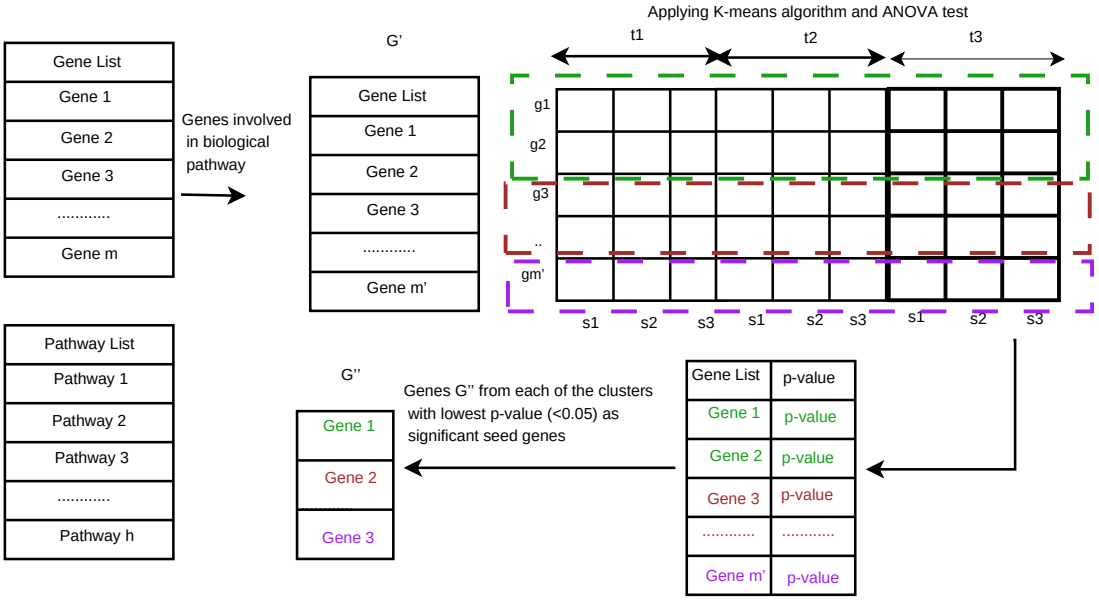


Figure 6.2: Schematic diagram of identification of significant seed genes.

we include an illustrative example in Figure 6.3.

6.4.3 Mining biclusters

POPTric is interested to seek patterns of the following types: additive, multiplicative, and additive-multiplicative across samples in a subset of time points. Additionally, it also takes into consideration both positively and negatively co-expressed genes while searching for biclusters. The algorithm utilizes the pathway information to identify the initiator gene as well as the seed patterns depending on which other genes are identified to form a bicluster. For the identification of a subset of samples, we utilize LCS. We find T number of biclusters from each time point based on each significant seed gene. Let us assume $Bic = \phi$ is an empty set for a maximum of T number of biclusters. This is shown in Algorithm 5 and explained next.

Identification of base plane: In this step, we seek for the base plane as explained next which helps to recognize bicluster from a significant seed gene i.e., $g_a \in G''$ for base time points. We assemble previously found groups of genes $I \subseteq G$ corresponding to the cluster say $cl_{c'}$ of g_a i.e., $g_a \in cl_{c'}$ and consider those genes whose p-values (the result of ANOVA test) are less than α. Thereafter, we calculate the MSR^z_{2D} (Equation 2.3.9) value for each cluster considering the

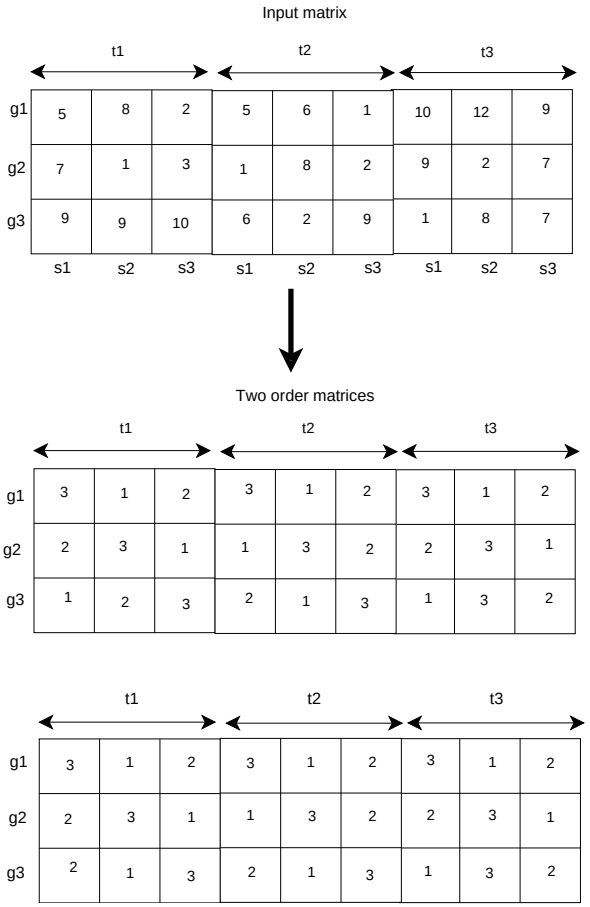


Figure 6.3: Conversion of input matrix into order matrices.

subset of genes $I \subseteq G$ and all the samples S i.e., $\beta(I, S)$ based on the IS_z, where $z = \{1, 2, \ldots, v\}$. Among all v number of MSR_{2D} values the minimum MSR_{2D} is selected as the base plane $GS_b, t_b \in T$. Next, we consider GS_b plane for further computations.

Computation of overlapping score: As triclustering algorithm is an NP-hard problem, therefore our target is to reduce the time complexity. We start the identification of biclusters β from each of the significant seed genes G''. Hence, the number of triclusters is $\leq |G''|$. Next, we compute the overlapping scores using Equation 5.4.1 between a $g_a \in G''$ and all $g_b \in I$, where $g_a \neq g_b$ in order to find a list of genes $G^M \subseteq G$ that share a high number of pathways in common with the seed gene g_a which signify the maximum overlapping score. We cannot perform this step for artificial datasets.

Identification of seed patterns: In order to identify the seed patterns, we use the LCS algorithm to get a maximal set of conditions pertaining to the definition

Algorithm 4: POPTric algorithm

Input : $\mathcal{D}_{G\times(S\times T)}$ with a set of genes $G = \{g_1, g_2, \ldots, g_m\}$, a set of samples $S = \{s_1, s_2, \ldots, s_n\}$ and a set of time points $T = \{t_1, t_2, \ldots, t_v\}$, set of pathways identified from $P = \{p_1, p_2, \ldots, p_h\}$ DAVID, K, $\alpha = 0.05$, S_{min}, T_{min}, ϵ, $\bar{\delta}$

Output: $Tric$: A final list of triclusters

$Tc = \phi$
Find out genes G' which are involved in at least one P
Compute p-value of all the genes in G' using ANOVA test
Apply K-means algorithm on $\mathcal{D}_{G'\times(S\times T)}$, where K is number of clusters $CL = \{cl_1, cl_2, \ldots, cl_K\}$
Identify significant seed genes G'' from each cluster $cl_{c'}$, where p-value $< \alpha$ and minimum among all genes present in $cl_{c'}$
for *all* $t_z \in T$ **do**
    Compute $OM_{G\times S\times t_z}$ and $OM'_{G\times S\times t_z}$
end
for *all* $g_a \in G''$ **do**
    Mining biclusters Bic ($OM_{G\times S\times t_z}$, $OM'_{G\times S\times t_z}$, where $t_z \in T$, P, S_{min}, ϵ, CL, α)
    Obtaining triclusters $\mathcal{T}$ from biclusters Bic (Bic, T_{min}, $\bar{\delta}$)
    Add $\mathcal{T}$ to Tc
end
Prune triclusters Tc to get final list of triclusters $Tric$

of bicluster having a maximum number of samples. The elementary concept of applying LCS between pair of rows is built on the observation about the presence of a common subsequence between order matrix ($OM_{G^M\times(S\times t_b)}$) rows. We obtain L number of seed patterns between g_a and for all $g_w \in G^M$. Thus, we get L number of patterns $Seed_{pt} = \{pt_1, pt_2 \ldots, pt_L\}, L = |G^M|$, where size of each pattern should be at least S_{min} and $Seed_{pt} \leq L$. $Seed_{pt}$ is in decreasing order of pattern length i.e., $|pt_1| \geq |pt_2| \ldots \geq |pt_L|$. The reason for considering the longest patterns is to find the maximum sized biclusters. It is noteworthy that we can not compute overlapping scores for the synthetic datasets. Hence, for synthetic datasets we consider all genes $g_c \in I$ to find the seed pattern $Seed_{pt} = \{pt_1, pt_2 \ldots, pt_{|I|}\}$. The main reason is to reduce the time complexity and not search for all the genes which is actually a time-consuming process. Except for the changes in the number of seed patterns, all other steps are the same for both real and artificial datasets. We use notations based on real datasets.

Mining OPPM genes: In this step, each pattern $pt_{l'} \in Seed_{pt}$, where $l' = \{1, 2, \ldots, L\}$ triggers the formation of an initial bicluster β^b for a time point t_b

Algorithm 5: Mining biclusters

Input : $OM_{G\times S\times t_z}$, $OM'_{G\times S\times t_z}$ where $t_z \in T$, S_{min}, ϵ, CL, α
Output: Bic: A list of biclusters

$Bic = \phi$
Find the cluster membership of g_a, say $cl_{c'}$
Identify the genes $I \subseteq G \in cl_{c'}$ and p-values of each gene $< \alpha$
for *all* $t_z \in T$ **do**
    Calculate MSR^z_{2D} for the cluster $cl_{c'}$ considering the subset of genes $I \subseteq G$ and all samples S in t_z
end
Find the time point t_b which gives minimum MSR^b_{2D} value
$\mathcal{X}^b = g_a$
for *all* $g_b \in I$ **do**
    Compute $O_{score}(g_a, g_b)$ and $g_b \neq g_a$
end
Determine a gene list G^M which has maximum O_{score} with g_a
$Seed^{pt} = \phi$
for *all* $g_w \in G^M$ **do**
    Compute $pt_{l'}$ by LCS between pair of (g_a, g_w) using $OM_{G^M\times(S\times t_b)}$
    if $|pt_{l'}| \geq S_{min}$ **then**
        $Seed^{pt} = Seed^{pt} \cup pt_{l'}$
    end
end
Sort all patterns of $Seed^{pt}$ in descending order, arrange $g_w \in G^M$ based on longest pattern produced with g_a and cluster expansion starts with new set of genes
for *all* $pt_{l'} \in Seed^{pt}$ *and* $g_w \in G^M$ **do**
    $\mathcal{X}^b = \mathcal{X}^b \cup g_w$
    $Y = pt_{l'}$
    for *all* $g_i \in G \setminus \{\mathcal{X}^b\}$ **do**
        if $OM_{g_i\times(S\times t_b)}$ *approximately matches to* $pt_{l'}$ *with* ϵ **then**
            $\mathcal{X}^b = \mathcal{X}^b \cup g_i$
            else if $OM'_{g_i\times(S\times t_b)}$ *approximately matches to* $R(pt_{l'})$ *with* ϵ **then**
                $\mathcal{X}^b = \mathcal{X}^b \cup g_i$
            end
        end
    end
    if $|\mathcal{X}^b| \geq |Y|$ **then**
        $\beta^b \leftarrow \{\mathcal{X}^b, Y\}$
        Add β^b to Bic
        break
    else
        $\mathcal{X}^b = g_a$
    end
end

```
for t_{z'} ∈ T \ t_b do
    X^{z'} = g_a
    for all g_i ∈ G \ {X^{z'}} do
        if OM_{g_i×(S×t_{z'})} approximately matches to Y with ε then
            X^{z'} = X^{z'} ∪ g_i
            else if OM'_{g_i×(S×t_{z'})} approximately matches to R(Y) with ε
             then
                X^{z'} = X^{z'} ∪ g_i
            end
        end
    end
    if |X^{z'}| ≥ |Y| then
        β^{z'} ← {X^{z'}, Y}
        Add β^{z'} to Bic
    end
end
```

which is initially empty, say $\beta^b = \phi$. Bicluster expansion process starts with two genes g_a and g_w i.e., $\beta^b = \{g_a, g_w\}$, where $g_a \in G'', g_w \in G^M$, and g_w has the largest LCS with respect to g_a. The more generalized version of adding genes in a bicluster can be defined as follows.

From a converted 2D matrix $G \times (S \times T)$, the bicluster $\beta^b_{\mathcal{X}^b \times Y}$ can be extracted by identifying a subset of genes $\mathcal{X}^b \subseteq G$ under a subset of conditions $Y \subseteq S$, where each of the genes $g_i \in \mathcal{X}^b$ is order-preserved having maximum error $\epsilon = \frac{\mathcal{K}}{|Y|}$ with the identified pattern $Y = pt_{l'}$, where $l' = \{1, 2, \ldots, L\}$. Here, $\mathcal{K}$ is the maximum number of allowed mismatches.

Bicluster β^b starts to add one gene at a time from $\{G \setminus \{g_a, g_w\}\}$ having a similar pattern with the $pt_{l'}$ in $Seed_{pt}$. Next, we focus on identifying positively as well as negatively co-expressed genes on the basis of seed pattern $pt_{l'}$. We further extend our β^b by checking the rest of the genes $g_i \in \{G \setminus \{g_a, g_w\}\}$ and adding a new gene using the following two rules.

(i) Add g_i at a time from $OM_{G\times(S\times t_b)}$ considering the general version of adding genes in β^b with maximum error ϵ.

(ii) If g_i is not included in the first check, it can be added next using the reverse seed pattern ($R(pt'_l)$) which indicates the opposite (negative/positive) pattern of pt_1 (positive/negative) in a similar fashion, from $OM'_{G\times(S\times t_b)}$.

It is important to note the fact that positive and negative co-expressed patterns which are actually opposite in nature can form a cluster. Negative co-expressed

patterns are explained in Section 5.4.2. We keep on adding new genes (either positive or negative) into the β^b until all genes are visited once. A bicluster is considered to be the final bicluster if it satisfies the criteria $|\mathcal{X}^b| \geq |Y|$ [337] otherwise we move to next $pt_{l'}$ in $Seed_{pt}$, where $l' = \{2, 3, \ldots, L\}$ until we find a bicluster from that seed gene g_a. It has been observed that gene expression data have a larger number of genes than the samples therefore, we keep the criteria to have a bicluster as $|\mathcal{X}^b| \geq |Y|$. To reduce the time complexity, we will not search for all $pt_{l'}$, as soon as we get a bicluster from $pt_r, r \in l'$ we skip all others $pt_{L\setminus\{1,2,\ldots,r\}}$. It may also happen that, we may not find a bicluster after searching the entire $seed_{pt}$.

Finally, we find a bicluster $Bic = \{\beta^b\}$ comprised of a subset of genes $\mathcal{X}^b \in G$ and a subset of columns $Y \in S$ for t_b time point. Now, Y is considered as the base pattern. We take the base pattern $pt_r = Y$ and repeat only the process of mining OPPM genes for all other time points $\{T \setminus t_b\}$. Let us again assume the initial bicluster $\beta^{z'} = g_a$ for $t_{z'}$, where $z' = \{T \setminus t_b\}$. We include genes from $G \setminus g_a$ by following the above-mentioned two rules until all genes are visited once. The bicluster $\beta^{z'}_{\mathcal{X}^{z'} \times Y}$ should obey the criteria of $|\mathcal{X}^{z'}| \geq |Y|$, where $\mathcal{X}^{z'} \in G$. (Again, sometimes the bicluster may not be available depending on Y.) We add $\beta^{z'}$ in $Bic = \{\beta^b, \beta^{z'}\}$. Therefore, the algorithm results in a maximum of v or $|T|$ number of biclusters $Bic = \beta^{z'}$, where $z' = \{1, 2, \ldots, v\}$ because for each time point, only one bicluster is identified.

Lemma 6.4.1 *From each seed gene, the maximum of v number of biclusters are obtained.*

Proof: Let $g_a \in G''$ be the seed gene. L number of seed patterns, $Seed_{pt} = \{pt_1, pt_2 \ldots, pt_L\}$, are obtained between g_a and all $g_w \in G^M$, where $L = |G^M|$ and G^M is the list of genes, $G^M \subseteq G$ that share the maximum number of pathways with seed gene, g_a. The size of each pattern, $pt_{l'} \in Seed_{pt}$ (where $l' = \{1, 2, \ldots, L\}$) is at least S_{min} (i.e., $|pt_{l'}| = S_{min}$). The seed patterns are sorted in descending order (i.e., $|pt_1| > |pt_2|, \ldots, |pt_L|$) to help in identifying bicluster β^b from base plane with respect to largest sized seed pattern i.e., bicluster identification starts in the order $Seed_{pt}$ is arranged. On obtaining a bicluster pt_r, $(r \in l')$ all other $pt_{l'} \in_{l'inr} Seed_{pt}$ are ignored and the search for bicluster with respect to g_a stops. Therefore, only one bicluster is obtained from a given seed pattern. This is repeated for all the time points $t_{z'}$, where $z' = \{T \setminus t_b\}$ and $z' = \{1, 2, \ldots, v\}$ for identifying other biclusters. Hence, the maximum of v number of biclusters $Bic = \{\beta^1, \beta^2, \ldots, \beta^v\}$ are obtained from each seed gene.

6.4.4 Obtaining triclusters from biclusters

After obtaining v number of biclusters $Bic = \{\beta^1, \beta^2, \ldots, \beta^v\}$ for all the time slices, we now proceed to mine a tricluster. This step takes as input Bic, $\bar{\delta}$, and T_{min}, which is an user-defined threshold. Here, biclusters have a different subset of genes and the same subset of conditions Y throughout all biclusters. Now, it's time to discover the subset of time points Z and a subset of genes X amongst all these biclusters to identify a single tricluster $\mathcal{T}$. To do this, we find the BD^{β^z}, $z = \{1, 2, \ldots, v\}$ score for each bicluster β^z present in Bic and sort the BD^{β^z} scores in ascending order such that $BD^{\beta^p} \leq BD^{\beta^q} \leq BD^{\beta^s} \ldots \leq BD^{\beta^v}$ to process the biclusters further. Let's assume bicluster $\beta^p = \{\mathcal{X}^p, Y\}$ at time t_p has the smallest BD^{β^p} score and $\beta^q = \{\mathcal{X}^q, Y\}$ at time t_q gives the second-lowest BD^{β^q} score. Therefore, the initial tricluster is made up with $\mathcal{T} = \{\phi, Y, t_p\}$ with the base pattern but no genes and only t_p time point since we have not processed any biclusters yet. We define length of genes in a bicluster LN_β as $LN_\beta = length(\{g_x, g_x \in G\})$. Let's begin to add time points in $Z = \{t_p\}$ if any one of the following criteria satisfies.

(i) For any two biclusters β^p and β^q, if $LN_{\beta^p} \geq LN_{\beta^q}$ and $\bar{\delta}_{new} = \frac{LN_{\beta^q} - LN_{\beta^p}}{LN_{\beta^q}} \leq \bar{\delta}$ we add the time point t_q to Z i.e., $Z = \{t_p, t_q\}$. This signifies that the smaller cluster has few extra elements, therefore we can add the time point.

(ii) For any two biclusters β^p and β^q, if $LN_{\beta^p} \leq LN_{\beta^q}$ and $\bar{\delta}_{new} = \frac{LN_{\beta^p} - LN_{\beta^q}}{LN_{\beta^p}} \leq \bar{\delta}$ we add the time point t_q to Z i.e., $Z = \{t_p, t_q\}$.

After addition of time points we update the merging threshold $\bar{\delta} = \bar{\delta} + \bar{\delta}_{new}$ and the intersection of $\mathcal{X}^p$ and $\mathcal{X}^q$ is added to X i.e., $X = \{\mathcal{X}^p \cap \mathcal{X}^q\}$. Subset of genes X become again $\mathcal{X}^p$. We continue to process all the biclusters from t_s time point to t_l time point with the bicluster at updated t_p time point to find the time points as mentioned before. During this process, we need to check the criteria to be a tricluster $\mathcal{T}$ to add into the list of triclusters Tc, i.e., $|Z| \geq T_{min}$. Till now we have only a single tricluster from g_a. The entire process is repeated until all the genes in G'' are visited once to get list of triclusters Tc. The pseudo-code is given in Algorithm 6.

6.4.5 Tricluster pruning

POPTric identifies a maximum of $|G''|$ number of triclusters Tc. Now, it's time to remove any smaller sized triclusters and redundant clusters. To prune the triclusters, we measure the volume of each tricluster $Vol^{\mathcal{T}} = (|X| * |Y| * |Z|)$. We

Algorithm 6: Mining triclusters

Input : $Bic = \{\beta^z\}$, where $z = \{1, 2 \ldots, v\}$, T_{min}, $\bar{\delta}$
Output: $\mathcal{T}$: Tricluster

for *all* $t_z \in T$ **do**
  Calculate BD^{β^z}
end
Sort BD^{β^z} in ascending order $BD^{\beta^p} \leq BD^{\beta^q} \leq BD^{\beta^s} \ldots \leq BD^{\beta^v}$
$\mathcal{T} = (X, Y, Z)$, $X = \phi$, $Z = \{t_p\}$
$Z = \{t_p\}$
for i=t_q *to* t_v **do**
  $\beta^q = \beta^i$
  if $LN_{\beta^p} \geq LN_{\beta^q}$ && $\bar{\delta}_{new} = \frac{LN_{\beta^q} - LN_{\beta^p}}{LN_{\beta^q}} \leq \bar{\delta}$ **then**
    $Z = \{t_p, t_q\}$
    else if $LN_{\beta^p} \leq LN_{\beta^q}$ && $\bar{\delta}_{new} = \frac{LN_{\beta^q} - LN_{\beta^p}}{LN_{\beta^q}} \leq \bar{\delta}$ **then**
      $Z = \{t_p, t_q\}$
    end
  end
  $X = \mathcal{X}^p \cup \mathcal{X}^q$
  $\bar{\delta}_{new} = \bar{\delta} + \bar{\delta}_{new}$
  $\beta^p = X$
end
if $|Z| \geq T_{min}$ **then**
  Consider $\mathcal{T}$ as a tricluster
end

first sort the triclusters according to their volume such as $Vol^{\mathcal{T}^1} \geq Vol^{\mathcal{T}^2} \geq \ldots \geq Vol^{\mathcal{T}^{|G''|}}$. At first, we take $\mathcal{T}^1$ and gradually compare with all other triclusters one by one to keep triclusters which has smaller overlapping scores than a user-specified overlap threshold θ. Therefore, no two triclusters have an overlapping score greater than θ. Suppose, two triclusters are $\mathcal{T}^i$ with a subset of genes X^i, a subset of samples Y^i, and a subset of time Z^i and $\mathcal{T}^j$ with a subset of genes X^j, a subset of samples Y^j, and a subset of time Z^j. The overlapping in between these two triclusters can be computed using Equation 6.4.3. Hence we keep the larger sized triclusters and remove all smaller ones. As an output, we receive a list of $Tric$ triclusters.

$$OV(\mathcal{T}^i, \mathcal{T}^j) = \frac{|(X^i \cup Y^i \cup Z^i) \cap (X^j \cup Y^j \cup Z^j)|}{min(|X^i \cup Y^i \cup Z^i|, |X^j \cup Y^j \cup Z^j|)} \tag{6.4.3}$$

6.5 Time complexity

In terms of time complexity, we consider creation of order matrix, mining biclusters, and obtaining triclusters from biclusters. Recall, the 3D GST data is $\mathcal{D}_{G\times S\times T}$ of size $m \times n \times v$ consisting of m number of genes, n number of samples or experimental conditions, and v number of time points. $OM_{m\times n}$ and $OM'_{m\times n}$ matrices for each time point can be calculated within $O(mv(n \log n))$ and $O(mv(n \log n))$ time, respectively. The overall time complexity for this step is $O(mv(n \log n) + mv(n \log n)) \approx O(mv(n \log n))$. Let, $\bar{S}$ be the number significant seed genes, $\bar{P}$ be the number of genes which are associated with at least one pathway, and $\bar{G}$ be the number of genes which has maximum overlapping score with a seed gene. The time required to extract biclusters for each time is $O(\bar{S}\bar{G}nv(m + n))$ (Section 5.5). The complexity of identifying base plane is negligible as compared to bicluster mining. Finally, triclusters can be identified from biclusters within $O(\bar{S}v)$ time. Therefore, the overall time complexity is $O(mv(n \log n) + \bar{S}\bar{G}nv(m + n) + \bar{S}v) \approx O(v(m(n \log n) + \bar{S}\bar{G}n(m + n) + \bar{S})$ without tricluster pruning.

The running time of TriCluster algorithm is $v \times (time(multigraph) \times time(bicluster)) + time(tricluster)$. The time required to construct multigraph is $O(mn^2v)$. In TriCluster, the bicluster and tricluster mining steps can be the most expensive if the number of clusters is huge. The optimal number of clusters highly depends on the parameters. TriCluster algorithm is efficient for microarray datasets because (i) multigraph removes noise and unnecessary information, (ii) depth of the search is very small since the number of samples and time points are smaller than the number of genes, and (iii) intermediate gene sets are kept for all triclusters which prune the search if the criteria are not met [364]. If two triclusters have actual overlap then only the pruning and merging step is applied in $O(\mathbb{C} \log \mathbb{C})$ time, where $\mathbb{C}$ is the set of clusters. The time complexity of the OPTricluster algorithm is $O(m\Gamma\Lambda)$, where Γ be the number of all possible combinations of samples and Λ is the total number of order-preserved triclusters.

6.6 Performance analysis

The effectiveness of triclustering algorithms is evaluated using 210 synthetic datasets and one real dataset. We evaluate the performance of POPTric with state-of-the-art triclustering algorithms i.e., TriCluster [364], OPTricluster [316], and EMOA-δ-TRIMAX [38]. Concerning the parameter settings, wherever possible we use the parameters recommended by the original author's contribution

and some of them are tuned in order to get better results on synthetic datasets. Before applying the OPTricluster algorithm we have to convert 3D data into $G \times T \times S$ data i.e., for each sample, we consider GT data plane. The next section describes the synthetic datasets generation process.

6.6.1 Synthetic datasets generation

To begin with, we generate three different scenarios for artificial datasets: (i) implanting one tricluster, (ii) implanting three triclusters having different sizes, and (iii) adding noise in the first set of synthetic data in a three-dimensional background matrix. At first, we create a large two-dimensional background matrix where each entry are chosen from the normal distribution $N(0, 1)$ for each time point. Then for each of the scenarios, three different types of datasets are created such as additive, multiplicative, and additive-multiplicative triclusters and repeated 10 times. Firstly, we implant one tricluster of size $50 \times 15 \times 10$ ($X \times Y \times Z$) with background matrix of size $500 \times 30 \times 20$ ($G \times S \times T$). Figure 6.4 depicts the pattern of one of the implanted additive-multiplicative tricluster. Secondly, we implant three triclusters of different size $40 \times 10 \times 5$, $70 \times 15 \times 10$, and $100 \times 20 \times 15$ without overlap and without noise into the background matrix of size $700 \times 50 \times 30$ in different positions. Therefore, we generate a total of 60 artificial datasets for model testing of both scenarios. Figure 6.4 shows one additive-multiplicative tricluster which consists of 50 genes, under 15 samples across 10 time points. In addition, we also generate a noisy dataset by adding random noise drawn from a normal distribution with μ 0 and varying σ such as 0.05, 0.1, 0.15, 0.2, and 0.25 with each cell of the previously generated dataset in the scenario i. Therefore, we have 150 datasets for noise experiments.

6.6.2 Performance on synthetic datasets

To evaluate the performance of the proposed algorithm we define *Confirmation score* (CS) similarly as mentioned in the work of Prelic et al. [278] for two sets of biclusters. Let, $\mathcal{T}^1$ and $\mathcal{T}^2$ be the two sets triclusters. The CS of $\mathcal{T}^1$ with respect to $\mathcal{T}^2$ can be defined by Equation 6.6.1 which estimates the degree of similarity between original and discovered triclusters.

$$CS(\mathcal{T}^1, \mathcal{T}^2) = \frac{1}{|\mathcal{T}^1|} \sum_{(X^1,Y^1,Z^1)\in\mathcal{T}^1} max_{(X^2,Y^2,Z^2)\in\mathcal{T}^2} JC(\mathcal{T}^1, \mathcal{T}^2) \tag{6.6.1}$$

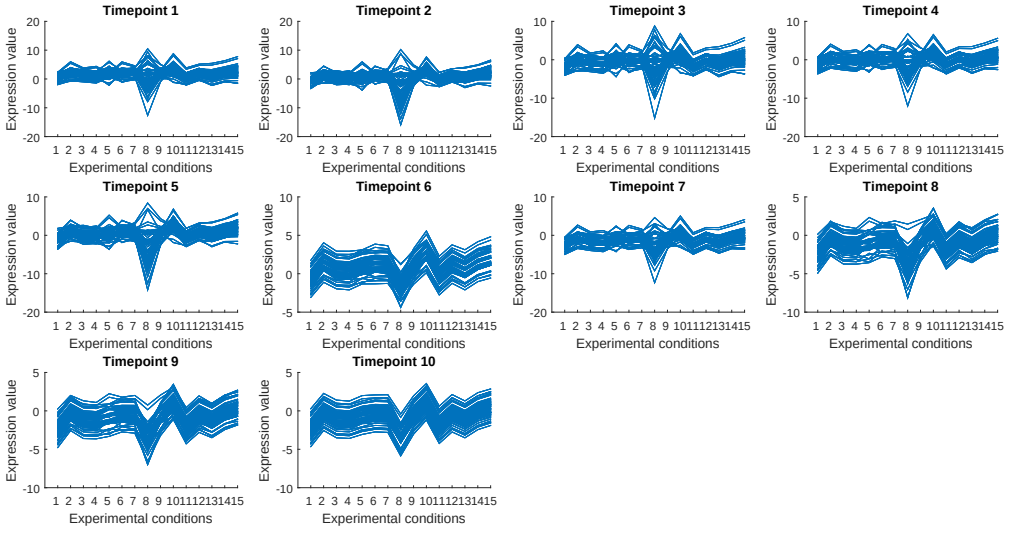


Figure 6.4: Additive-multiplicative tricluster with 50 genes, 15 experimental conditions, and 10 time points.

$$JC(\mathcal{T}^1, \mathcal{T}^2) = \frac{|(X^1 \cup Y^1 \cup Z^1) \cap (X^2 \cup Y^2 \cup Z^2)|}{|(X^1 \cup Y^1 \cup Z^1) \cup (X^2 \cup Y^2 \cup Z^2)|} \tag{6.6.2}$$

Suppose, we have two sets of triclusters $\mathcal{T}^O$ and $\mathcal{T}^D$, which represent a set of obtained triclusters by any triclustering algorithm and set of implanted triclusters, respectively. Then average recovery score $CS(\mathcal{T}^D, \mathcal{T}^O)$ corresponds to how well the triclustering algorithm is able to discover true triclusters from the input dataset. On the other hand, average relevance score $CS(\mathcal{T}^O, \mathcal{T}^D)$ quantifies to what extent retrieved triclusters represent true triclusters. Both the scores have a range from 0 to 1.

We run POPTric with the parameters $S_{min} = \lceil 5\% * |S| \rceil$ (default 2), $T_{min} = 2$ (default), $K = \sqrt{|G|}$, $\alpha = 0.05$, and $\theta = 0.25$ [278, 337]. To set the parameter $\bar{\delta}$ and ϵ we perform exhaustive experimentation by executing the algorithm from 0 to 0.5 with step size of 0.05 for ϵ and 0.1 to 0.3 with step size of 0.1 for $\bar{\delta}$ for every ϵ value. After execution, we choose the best result in terms of CS and reported the average relevance and recovery score for each of the tricluster type. OPTricluster [316] is proposed to mine triclusters for short time series data, therefore it does not yield triclusters for our synthetic datasets. For TriCluster [364] algorithm all the parameters such as minimum number of genes, minimum number of samples, and minimum number of time are kept as 2 same as our proposed algorithm. EMOA-δ-TRIMAX [38] requires four parameters i.e., λ, δ, no. of populations, and no. of generations. Bhar et al. [38] have recommended to compute λ and δ. no. of populations is kept 10, as number of triclusters in

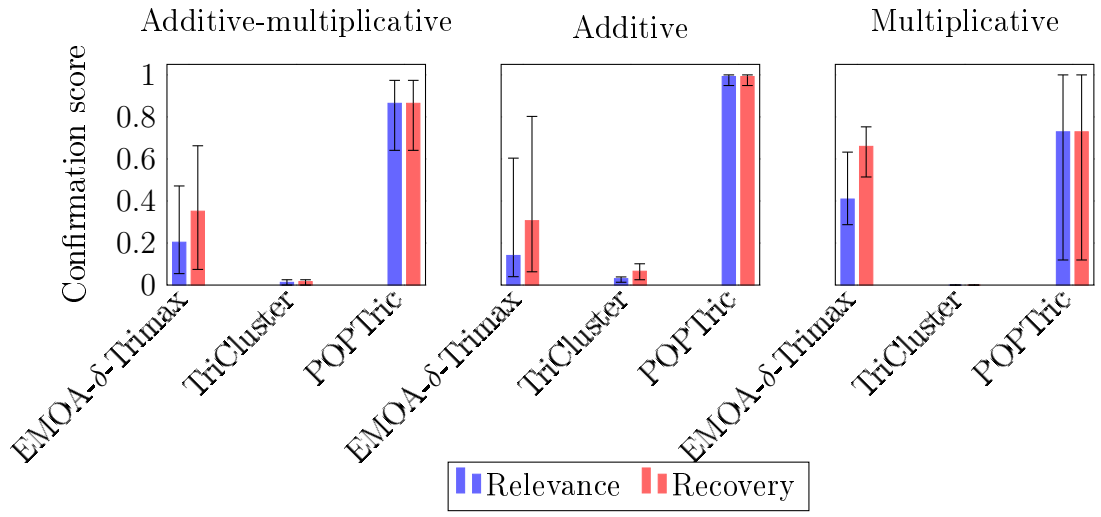


Figure 6.5: Relevance and recovery scores with error bars (range) of different triclustering algorithms on three different triclustering models for scenario i.

our synthetic dataset is less and No. of generations is 100.

Performance on model experiment: We use the first set of synthetic datasets, to know the best-suited triclustering algorithm which can recover various tricluster types. To evaluate the performance of the POPTric algorithm in the case of triclusters having different numbers of time points, we use the second set of synthetic datasets. From Figure 6.5, it can be observed that POPTric outperforms all other triclustering algorithms in terms of relevance and recovery score for scenario i. Moreover, Figure 6.6 depicts the relevance and recovery scores of different triclustering algorithms for scenario ii datasets. The figure suggests that our algorithm outperforms all other algorithms except for the multiplicative model in terms of recovery score. In the case of the multiplicative model, the recovery score is a little less than EMOA-δ-TRIMAX whereas the relevance score is higher than EMOA-δ-TRIMAX. TriCluster does not detect any triclusters for multiplicative datasets and for other datasets its relevance and recovery score is very low. EMOA-δ-TRIMAX identifies more number clusters than the original one, that is why its performance degrades for relevance score.

Robustness to noise: From the previous section, it has been clearly noticed that our algorithm performs well for all the triclustering models. In addition to the model experiment, we have also experimented to evaluate the robustness of POPTric in presence of noise. Finally, we present the average relevance and recovery scores for 10 datasets together for three models. The experimental results of noisy datasets are depicted in Figure 6.7. The result suggests proposed

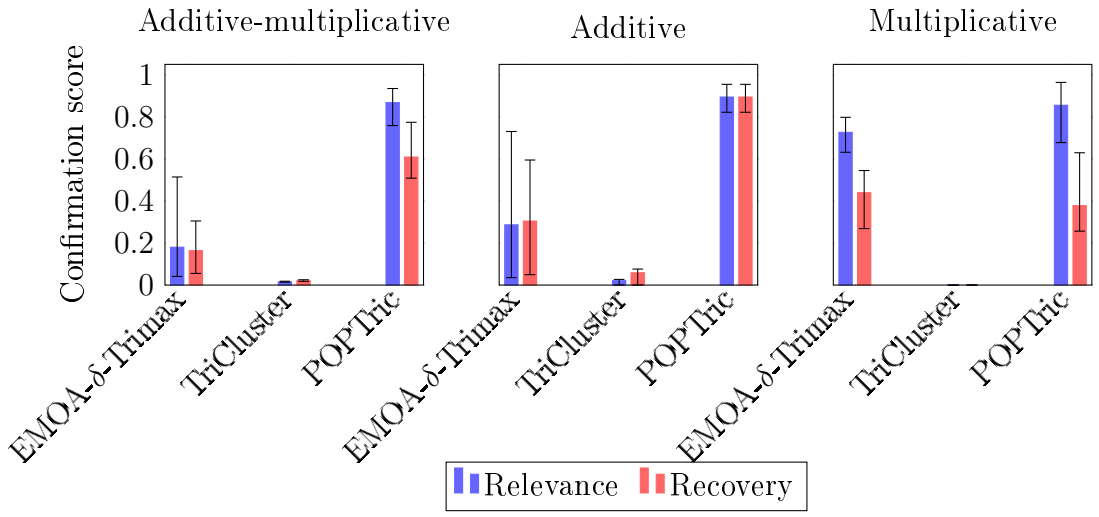


Figure 6.6: Relevance and recovery scores with error bars (range) of different triclustering algorithms on three different triclustering models for scenario ii.

algorithm outperforms all other triclustering algorithms for the additive model. EMOA-δ-TRIMAX works well for noisy multiplicative datasets and POPTric gives satisfactory results for the additive-multiplicative pattern. Our algorithm can not identify the significant seed genes for the multiplicative pattern that is why it gives a poor performance in discovering multiplicative triclusters.

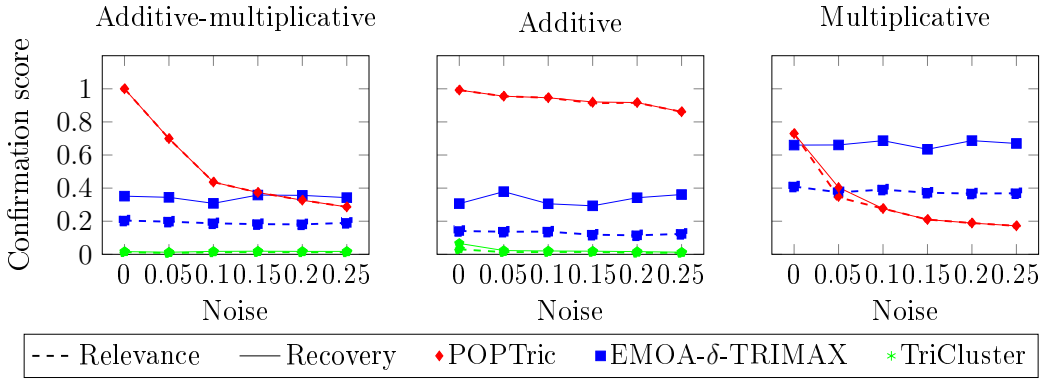


Figure 6.7: Relevance and recovery scores of different triclustering algorithms on three models over noisy data.

6.6.3 Performance on real dataset

The real microarray gene expression dataset is taken from the work mentioned in [130]. The dataset is publicly available in NCBI named GSE25011. This dataset measures the gene expression profile of 22,283 Affymetrix probe-set ids under 11 breast tumor samples of patients (MD53, MD57, MD64, MD40, MD49, MD50, MD52, MD66, MD67, MD69, MD71) in different time points. Originally there

are six different time points (0, 20, 40, 60, 120, and 180 min) and two snap frozen per sample except for MD40. Therefore, we have only considered these six time points for our experiment. Several duplicate official gene symbols are mapped into one Affymetrix probe-id. We preprocess the dataset by keeping one gene expression profile with maximum standard deviation among the same official gene symbols. After the data preprocessing step, we have ended up with 12,982 unique genes. We normalize the data for each row across all samples for a given time point to μ 0 and σ 1. We run POPTric with the parameter $\alpha = 0.05$, $K = 114$, $S_{min} = 2$, $T_{min} = 2$, $\theta = 1$, and $\epsilon = 0.1$ to 0.5 with step size 0.05 and $\bar{\delta} = 0.1$ to 0.3 with step size 0.1. We choose the parameters very carefully to report in this study. The parameters of POPTric algorithm are $\epsilon = 0.4$ and $\bar{\delta} = 0.3$. Our algorithm results in 37 triclusters for this dataset. Here, we have just removed all duplicate triclusters. The general guideline for selecting the parameters is to keep low ϵ and large $\bar{\delta}$. We compare the performance of our POPTric algorithm with other triclustering algorithms TriCluster and EMOA-δ-TRIMAX on real-life datasets using three quality indices MSR (Equation 2.4.5), Coverage (Equation 2.4.17), and TD score (Equation 6.4.1) [37, 234]. The parameter setting for real datasets can be found in Table 6.1. We have not found any triculster using TriCluster and OPTricluster algorithm for the GSE25011 dataset. Therefore, we are not able to report comparative results with these two algorithms.

Table 6.1: Parameter settings of different triclustering algorithms for real dataset

Algorithm	Parameter settings
EMOA-δ-TRIMAX	$\lambda = 1.2$, $\delta = 0.0545$, No. of population = 100, No. of generations = 100
TriCluster	Minimum no. of genes = 85, Minimum no. of samples = 4, Minimum no. of time = 3
POPTric	$S_{min} = 4$, $T_{min} = 2$, $K = 114$, $\bar{\delta} = 0.1$ to 0.3, $\epsilon = 0.1$ to 0.5, $\alpha = 0.05$, $\theta = 1$

Table 6.2 shows the comparison among all triclustering algorithms in terms of Coverage (Gene G_C, Sample S_C, Time T_C), mean MSR value, and mean TD scores among all triclusters. From the table, we can clearly observe that POPTric and EMOA-δ-TRIMAX give similar results in terms of Coverage. This comparative result shows POPTRic shows better results in terms of TD score.

Enrichment analysis: In order to establish the biological significance of genes belonging to each resulting tricluster, we perform GO enrichment analysis. For

Table 6.2: Comparisons of triclustering algorithms using different metrics.

Algorithm	G_C	S_C	T_C	Coverage	MSR	TD
EMOA-δ-TRIMAX	100	100	100	100	4.87E-01	5.59E-06
POPTric	100	100	100	100	8.64E-01	2.96E-06

Table 6.3: Comparisons of triclustering algorithms using different metrics.

Algorithm	GO ID	Name	p-value
EMOA-δ-TRIMAX	GO:0005515	protein binding	2.3975E-160
POPTric	GO:0005515	protein binding	2.9671E-164

this, we again use the web-based tool FuncAssociate [35] to calculate p-values. GO terms are said to be significant if the p-value is lower than the significant cut-off 0.05. We identify statistically enriched terms corresponding to each tricluster. Among all these statistically significant terms, we report the lowest one in Table 6.3. It can be observed the POPTric give the lowest value for p-value. Hence, POPTric provides a better significant tricluster than EMOA-δ-TRIMAX.

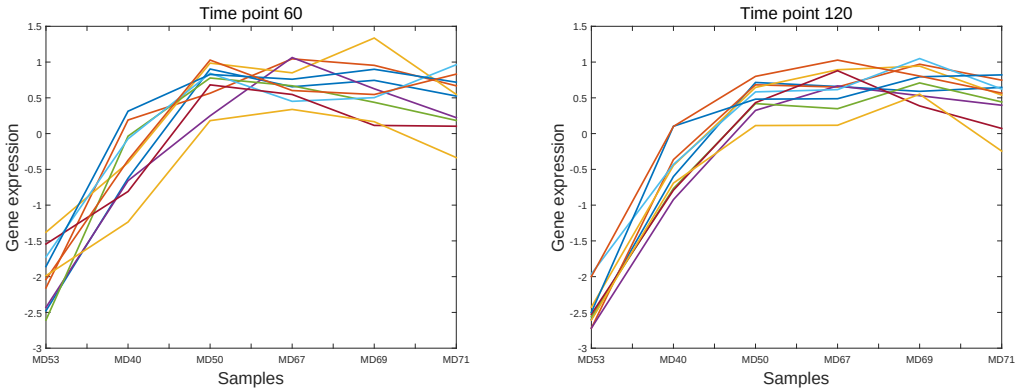
6.7 Potential biomarkers identification

Each tricluster is represented by an eigen gene [37, 186] which is calculated by using Singular Value Decomposition (SVD) of normalized (mean 0 and variance 1) data matrix of each tricluster. The expression matrix $\mathcal{D}^j_{x\times(y\times z)}$ is denoted for j^{th} tricluster, where x, y and z represent the number of genes, samples and time points. SVD of a matrix can be defined using Equation 6.7.1, where U is a $x \times (y \times z)$ matrix having orthonormal columns, D is a diagonal matrix $(y\times z)\times(y\times z)$ with singular value and V is a orthogonal matrix $(y\times z)\times(y\times z)$.

$$\mathcal{D}^j = UDV^T \tag{6.7.1}$$

Now, the eigengene of j^{th} tricluster is the first column of matrix V [37]. To identify the hub genes of each tricluster, we calculate the PCC between the eigen gene and all genes present in a tricluster. Thereafter, we take the top ten genes as hub genes having the highest PCC values. PCC values signify that hub genes are highly correlated with the eigen gene. In Figure 6.8, we have shown the gene expression profile of 10 hub genes of tricluster number 37 in two different time points in 60 minutes and 120 minutes. It can be observed from the figure that, genes in the same tricluster have a similar expression over two time points.

In Table 6.4, we report hub genes of some of the triclusters that are asso-



(a) Gene expression at time point 60 minutes (b) Gene expression at time point 120 minutes

Figure 6.8: Expression profiles of hub genes (*TTC39A*, *TRAPPC11*, *ABAT*, *APEH*, *TAF9B*, *ERCC1*, *NDUFAF3*, *SHQ1*, *GPC1*, *GNAQ*) in Tricluster number 37.

ciated with some breast cancer-related pathways. The third column of the table shows the average PCC value of 10 genes for each cluster. DAVID tool [144] is used in order to find out the pathways. In tricluster 3 we find *DBN1*, *CKAP2*, *CNN3*, *IGHV4-31*, *ITPKB*, *FLNA*, *HMGB1*, *PSME4*, *DAP3*, and *NPC2* genes as hub genes that are co-expressed over all time points. *FLNA* is associated with *MAPK signaling pathway* which plays an important role in proliferation and apoptosis progress in breast cancer [167]. *CACNA2D2* (tricluster 14 and 24) and *MAPT* (tricluster 19) are also corresponding with *MAPK signaling pathway*. *COQ4*, *MCM3AP*, *G6PC3*, *CASK*, *BICD2*, *PGR*, *COQ7*, *RPA3*, *RNF167*, and *SLX1A* are found from tricluster 5 as hub genes under 5 samples over 6 different time points. Gene *G6PC3* in this cluster belongs to *PI3K-Akt signaling pathway*. It has been observed that *PI3K-Akt signaling pathway* is active in upto 81% breast cancer patients [44]. *GNG4* (tricluster 7), *EFNA4* (tricluster 21), *COL1A1* (tricluster 33), and *COL6A2* (tricluster 33) are linked to *PI3K-Akt signaling pathway*. *Wnt signaling pathway* is responsible for developing many cancer types, one of them is breast cancer [292]. Hub genes *FZD2*, *CCND1*, *TCF7L2*, and *SFRP4* present in tricluster numbers 6, 20, 21, and 32, respectively are associated with *Wnt signaling pathway*. According to the study in [220] *CCND1* is indicated as a potential biomarker for breast cancer. The study in [276] suggests that hub gene *CHD8* is mutated in breast cancer, which is found in tricluster number 18 and 36. In tricluster 18 hub genes are co-expressed over five samples and all six time points, whereas in tricluster number 36 hub genes are co-expressed across six samples under 2 time points. *GALNT7*, *PIGB*, *H1F0*, *SCNN1A*, *GOLM1*, *RNF11*, *DIXDC1*, *RSL24D1*, *ERBB3*, and *DAG1* are found

as hub genes of tricluster number 10. The co-expressed gene *ERBB3* with this tricluster participates in *ErbB signaling pathway* which is found to be associated with breast cancer cell [304]. KEGG pathway *TGF-beta signaling pathway* is a related with gene *TFDP1* which is co-expressed with other nine hub genes *AURKA*, *PRDX6*, *FXYD5*, *PSMA7*, *BLM*, *CTSL*, *CEBPB*, *SEMA4A*, and *CKS1B* across six time points. A previous study in [308], has reported the crosstalk in between *ERα* and *TGF-beta signaling pathway* in breast cancer cells. Another breast cancer-related pathway is *p53 signaling pathway* which can be found in triclusters 14, 20, and 26 with related genes *CCNB1*, *CCND1*, and *PERP* [319]. In case of tricluster 37, hub genes *TTC39A*, *TRAPPC11*, *ABAT*, *APEH*, *TAF9B*, *ERCC1*, *NDUFAF3*, *SHQ1*, *GPC1*, and *GNAQ* are co-expressed under six samples across two time points 60 mins and 120 mins. *GNAQ* from tricluster 37 and *MMP2* of tricluster 32 are associated with *Estrogen signaling pathway*. *GNAQ* is considered to be widely altered cancer types [268]. Therefore, development of drugs to target *GNAQ* mutation and abnormalities is needed. Gene *MMP2* is highly expressed in breast cancer cell which is related to the lymph node metastasis and tumor staging [197].

Finally, hub genes in triclusters 33 and 37 are considered as potential biomarkers because these two clusters show maximum average PCC values. Strong evidences are found from some studies that genes *MFAP2* [118], *DBN1* [6], *FBLN1* [289], *COL1A1* [213], *COL6A2* [11], *ENAH* [201], *LOXL1* [360], *GNAQ* [268], *ABAT* [61], *TAF9B* [83], *ERCC1* [96], and *GPC1* [333] have significant impact on breast cancer. Out of 20 genes, 12 genes are associated with cancer development. Rest of the 8 genes *COL16A1*, *BEX3*, *PLXNA1*, *TTC39A*, *TRAPPC11*, *APEH*, *NDUFAF3*, and *SHQ1* might play an important role in breast cancer.

Table 6.4: Hub genes identified by POPTric algorithm.

Tricluster	Hub genes	PCC
3	*DBN1*, *CKAP2*, *CNN3*, *IGHV4-31*, *ITPKB*, *FLNA* (hsa04010:*MAPK signaling pathway*), *HMGB1*, *PSME4*, *DAP3*, *NPC2*	0.95
5	*COQ4*, *MCM3AP*, *G6PC3* (hsa04151:*PI3K-Akt signaling pathway*), *CASK*, *BICD2*, *PGR*, *COQ7*, *RPA3*, *RNF167*, *SLX1A*	0.95
6	*PLXNC1*, *LAS1L*, *NSUN5P1*, *SERPINH1*, *FZD2* (hsa04310:*Wnt signaling pathway*), *EFS*, *GLIPR1*, *OSBPL9*, *PPP1R15A*, *TWIST1*	0.94

Continuation of Table 6.4

Tricluster	Hub genes	PCC
7	*APOE*, *APOC1*, *CEBPA*, *MIR6756*, *SLC1A6*, *CARM1*, *LYZ*, *PHGDH*, *TNFAIP3*, *GNG4* (hsa04151:*PI3K-Akt signaling pathway*)	0.96
10	*GALNT7*, *PIGB*, *H1F0*, *SCNN1A*, *GOLM1*, *RNF11*, *DIXDC1*, *RSL24D1*, *ERBB3* (hsa04012:*ErbB signaling pathway*), *DAG1*	0.95
11	*AURKA*, *PRDX6*, *FXYD5*, *PSMA7*, *BLM*, *CTSL*, *CEBPB*, *SEMA4A*, *TFDP1* (hsa04350:*TGF-beta signaling pathway*), *CKS1B*	0.95
14	*ARL6IP1*, *BCAP31*, *TEAD4*, *MZB1*, *AMD1*, *QPRT*, *MCUR1*, *CCNB1* (hsa04115:*p53 signaling pathway*), *IL32*, *WIPI1*	0.94
15	*FXYD3*, *AGPAT1*, *AAR2*, *TRPC4AP*, *NTS*, *STAU1*, *PIGT*, *CACNA2D2* (hsa04010:*MAPK signaling pathway*), *ERGIC3*, *QDPR*	0.95
18	*UBE2C*, *MIR6756*, *ARHGEF7*, *BORA*, *B4GALT5*, *SNRPG*, *AURKA*, *PCID2*, *CHD8* (hsa04310:*Wnt signaling pathway*), *DDX39A*	0.95
19	*CA12*, *CIRBP*, *SRI*, *CERS6*, *GPD1L*, *TBC1D9*, *KLHDC2*, *MAPT* (hsa04010:*MAPK signaling pathway*), *MOAP1*, *SLC19A2*	0.95
20	*PLK2*, *CCND1*(hsa04115:*p53 signaling pathway*, hsa04151:*PI3K-Akt signaling pathway*, hsa04310:*Wnt signaling pathway*), *RNF11*, *ERBB3* (hsa04012:*ErbB signaling pathway*), *TTC39A*, *KAT6B*, *CRIP1*, *PLXNB1*, *RPL29*, *PIGB*	0.94
21	*SERPINH1*, *S100A11*, *DBN1*, *TCF7L2* (hsa04310:*Wnt signaling pathway*), *MIR664B*, *PRRC2C*, *PFDN2*, *EFNA4* (hsa04151:*PI3K-Akt signaling pathway*), *LAS1L*, *PSME4*,	0.94
24	*TCTA*, *DCXR*, *WFS1*, *TCN1*, *BCAS1*, *ZNF552*, *QDPR*, *ERGIC3*, *CPE*, *CACNA2D2* (hsa04010:*MAPK signaling pathway*)	0.93

Continuation of Table 6.4

Tricluster	Hub genes	PCC
26	*MIR7112*, *BZW2*, *IGHV4-31*, *PCID2*, *PERP* (hsa04115:*p53 signaling pathway*), *NASP*, *SNRPE*, *MMD*, *TOMM20*, *CTSC*	0.97
32	*NSA2*, *RPL27A*, *SFRP4* (hsa04310:*Wnt signaling pathway*), *RPL27*, *PPIC*, *MMP2* (hsa04915:*Estrogen signaling pathway*), *EEF1A1*, *PFDN5*, *HTRA1*, *SPARCL1*	0.96
33	*MFAP2*, *COL16A1*, *DBN1*, *FBLN1*, *COL1A1* (hsa04151:*PI3K-Akt signaling pathway*), *COL6A2* (hsa04151:*PI3K-Akt signaling pathway*), *ENAH*, *BEX3*, *PLXNA1*, *LOXL1*	0.98
36	*CHD8* (hsa04310:*Wnt signaling pathway*), *RPS7*, *MIR1178*, *HSPB6*, *ATP2B4*, *EFNB2*, *PSMA7*, *TUBB*, *MTMR14*, *STAU1*	0.96
37	*TTC39A*, *TRAPPC11*, *ABAT*, *APEH*, *TAF9B*, *ERCC1*, *NDUFAF3*, *SHQ1*, *GPC1*, *GNAQ* (hsa04915:*Estrogen signaling pathway*)	0.98

6.8 Discussion

In this chapter, we have proposed a novel semi-supervised triclustering algorithm POPTric for three dimensional Gene Sample Time data that aims to identify coherent subspaces across genes, samples, and time points. We compare the performance of our algorithm using artificial and real datasets. The assessment of artificial datasets are done by relevance and recovery scores. POPTric outperforms the existing TriCluster algorithm. For some types of tricluster EMOA-δ-TRIMAX shows better results in the case of artificial data. The results of GO enrichment analysis show that our algorithm is able to extract more biologically significant clusters than existing triclustering algorithms. In terms of Coverage, MSR score, and TD score both POPTric and EMOA-δ-TRIMAX algorithms show similar kinds of results. Additionally, we represent each of the tricluster by eigen gene and identified hub genes using the profile of eigen genes. Hub genes are associated with some breast cancer-related pathways and are also verified from the literature. From, these hub genes we further identify potential biomarkers responsible for breast cancer. Other biomarkers might be associated

with breast cancer, which is essential to be verified experimentally. This will help insights into a better breast cancer diagnosis. We now present the conclusions and future work of our thesis in the next chapter.

7

Conclusions and Future work

In this thesis we have explored the analysis of transcriptomics data with different computational methods and their impact on predicting potential biomarkers. In this regard, three major computational methods are presented full-space clustering, biclustering, and triclustering algorithms. Specifically, we have incorporated biological knowledge from GO in full-space clustering and KEGG pathway information in both biclustering and triclustering algorithms. Moreover, we compare the performance of our proposed algorithms with well-established methods for synthetic and real datasets. This chapter is structured in two sections. We provide concluding remarks regarding each one of the proposed methods in Section 7.1. Section 7.2 presents the future direction of work.

7.1 Concluding remarks

In the first study, we propose three clustering algorithms in Chapter 3. First, we present GAClust (unsupervised) and then two semi-supervised full-space clustering algorithms viz, SDC and SGAClust to analyze cancer gene expression data. The goal of our proposed algorithms is to make the algorithm parameter less which can compute parameters dynamically depending on input data. Our work is different from conventional clustering algorithms, where we concentrate

on data extraction rather than data partitioning by partial clustering. We have shown the empirical evidence in Chapter 3 that our semi-supervised approach SGAClust has been consistently performing well in different aspects (enrichment analysis, number of significant terms, and in terms of p-value) over other unsupervised algorithms. This result suggests that integration of GO in the SGAClust algorithm detects biologically more significant clusters than other algorithms. In contrast, though the SDC algorithm is semi-supervised still is not performing well because it is not suitable for high dimensional data. We believe that our work gives insights into potential biomarkers for cancer disease by different proposed algorithms.

The second study considers the biclustering algorithm which is based on the premise that a subset of genes participates in certain cellular processes active under some subsets of conditions. We have developed an order-preserving algorithm namely OPBic to analyze both gene and miRNA expression data for cancer disease in Chapter 4. The results provided by the OPBic algorithm for synthetic datasets, allow us to identify different types of bicluster patterns and overlapping clusters. In fact, very few algorithms are present in the literature which can deal with all eight types of biclusters. From the biological evaluation of biclustering results, we have shown that OPBic outperforms C&C, BicSPAM, and UniBic, particularly in enrichment analysis of gene expression data. In the case of the miRNA dataset, the algorithm is effective to discover biologically and clinically meaningful clusters. The biclusters are used to predict potential gene and miRNA biomarkers for different cancer datasets. We believe that the results obtained by OPBic result will definitely be useful in disease diagnosis.

In the third study described in Chapter 5, attention is given specifically to incorporate the biological knowledge during the search process to ensure those co-expressed genes are highly relevant biologically. Therefore, we have developed a semi-supervised biclustering algorithm using the KEGG pathway called POPBic in Chapter 5 which is an extension of the OPBic algorithm. The results obtained by the algorithm suggests that POPBic has the potential to find multiple bicluster types as well as overlapped clusters. We have shown that the algorithm can find clusters from noisy data. POPBic outperforms different biclustering algorithms under consideration for the cancer gene expression dataset. Further, we explore the resulting biclusters to get potential biomarkers for genes and miRNAs.

The fourth study discussed in Chapter 6, concentrates on triclustering algorithm which can effectively analyze 3D gene expression cancer data. Motivated by our previous work, we have developed the POPTric algorithm. The

comparative results demonstrate that our POPTric algorithm outperforms other approaches in order to identify three types of triclusters such as additive, multiplicative, and additive-multiplicative. The study presented here extracts the co-expressed genes under a subset of samples over a subset of time points. POPTric demonstrates the ability to identify triclusters from noisy data. Our triclusters have been found to be significantly enriched than triclusters of related approaches. Additionally, from the extracted triclusters we have identified potential biomarkers which highlights their usefulness in clinical diagnosis.

The bottom line of our work is that clustering has extremely rich predictive power but is a difficult problem to handle. We want to highlight some crucial points regarding the clustering of transcriptomics data that clustering is dependent on the underlying structure of the data, appropriate parameter selection, total number of clusters, and the number of genes present in each group. As a conclusion, we can say that cluster analysis of transcriptomics data greatly reduce the search space for biologists and further biological assessment is required. Also, a "Good" cluster is a pure indication of biologically meaningful groups. Unsupervised full-space clustering algorithm considers only proximity measure but semi-supervised uses proximity measure in association with semantic similarity measure from the GO regardless of clustering algorithms. In practice, we believe that combined measure that is semantic and proximity is expected to give beneficial results than only proximity measure. The result of SGAClust strongly supports our observation regarding the incorporation of GO in the combined measure.

In the realm of full-space clustering, it is being realized that there is a need for biclustering in the context of biological data and therefore we have explored two different biclustering algorithms. For this particular case, we observe that incorporation of pathway knowledge gives better quality biclusters. From the evidence, overall we can conclude that semi-supervised algorithms provide more promising results than unsupervised algorithms. Indeed, with this conclusion further, we move towards a semi-supervised triclustering algorithm to analyze three dimensional Gene Sample Time data. Some of the predicted potential biomarkers for cancer disease need to be verified through wet lab experiments before being used as designated biomarkers. Reliable biomarkers are extremely beneficial in understanding the complexity of various diseases, reduction of cost, simplifying the experimental setup, and providing a reference to the actual wet laboratory experimental results. This will be helpful in better cancer management.

7.2 Future work

Although the literature is flooded with full-space and subspace clustering algorithms, still clustering will remain a hot research area. Our study gives further insights into the future direction to carry forward research in transcriptomics data clustering. In view of the detailed study of clustering, we can infer a useful guideline and a number of open problems. It will be better to use multiple clustering algorithms such as fuzzy clustering, multiobjective, and evolutionary algorithms along with the integration of external knowledge and run with different parameter settings to get biologically relevant information. Therefore, ensemble clustering methods will be an appropriate decision to analyze such types of biological data.

In a true sense, outliers can be present in the biological data due to several reasons, such as experimental fault, noise, or instability in measurements. Therefore, it is of utmost importance to detect outliers from the dataset. Otherwise, it may cause misleading results in machine learning algorithms or degrade its performance. However, the fundamental challenge of an outlier is to determine how much different its value should be from the rest of the data in a dataset. In chapter 3, we have proposed GAClust and SGAclust, which identify singleton clusters that can be treated as outliers. Further investigation of the usefulness of the proposed algorithms for outlier detection is yet to be studied in future. Thus, outlier detection can be a promising future research direction.

There is a clear limitation of our proposed full-space clustering algorithm in its inability to extract overlapping clusters rather, they find disjoint groups. Often it has been observed that genes may participate in multiple functions and thus genes may belong to more than one functional category. Therefore, the detection of overlapping clusters is a crucial task due to the multiplicity of gene functions and can be exploited as future work.

In semi-supervised full-space clustering algorithms, two weight factors are used to compute combined similarity. This study emphasizes the proximity weight factor than the semantic similarity weight factor. The effectiveness of the performance of clustering results with varying weights is yet to be performed. Furthermore, proposed subspace clustering algorithms are not free from user-defined parameters which we have chosen experimentally. Therefore, there is always ample scope for improving biclustering or triclustering algorithms by proposing a way to estimate parameters dynamically. It will be a promising direction to design new subspace clustering algorithms which are fully automated by nature i.e., independent of parameters as well as avoid redundancy present in the clusters.

Beyond this, the running time of the proposed algorithms is not satisfactory. The volume of data is growing at a faster scale. The field of clustering will evolve and adapt to cope with increasing size. Hence, another promising avenue of research will be to develop subspace algorithms that will reduce time complexity. A parallel concept can be a possible way to address the issue.

Another possible way to extend our work is to involve regulatory information in order to get the clusters to have both co-expression and co-regulation of genes. Not only that, it is always advisable to incorporate multiple biological sources such as Reactome, BioCyc, Human Protein Reference Database (HPRD), Database of Interacting Proteins (DIP), IntAct Molecular Interaction Database to get meaningful clusters.

Cluster validation is the key tool to evaluate the algorithms and to verify the clustering results. The striking observation from the result of cluster validation as reported in Chapter 3 is that none of the algorithms is performing best throughout all internal validation measures. Thus, it is impossible to recommend one single internal measure for all algorithms to judge the quality of clusters because it is biased towards the specific structure of the cluster. Due to this biasness, it may give a higher validity rating and wrong interpretation about the clustering result. Therefore, it will be interesting to develop a new evaluation measure or to ensemble more than one measure to evaluate clusters.

The most commonly conducted cluster evaluation process is GO enrichment analysis of co-expressed gene clusters. GO enrichment analysis depends on external knowledge repository and depends on completeness and availability of benchmark databases. The enrichment result is biased by the size of the clusters. With this, we add another important point that a higher number of the identified cluster does not essentially mean obtaining highly enriched clusters. It is worth mentioning that the larger the cluster smaller will be the p-value (close to 0). Therefore, it will be effective to work with different biological databases as an external knowledge base such as pathway enrichment analysis, Transcription Factor Binding Site (TFBS) enrichment analysis.

Our study is limited to only gene expression and miRNA expression data. So far, in our study, we have identified several biomarkers that are verified through literature only. The rest of the biomarkers need to be checked computationally to see if they are associated with the disease in any way. Also, no universal method is present in the literature which can evaluate the biomarkers. Therefore, future focus can be given in this direction to enhance the assessment of biomarkers to understand any complex disease, disease diagnosis, prognosis,

or risk analysis. Additionally, it will be beneficial to identify more specific and sensitive biomarkers. We will like to extend our work in different data sources such as RNA-Seq data, single-cell RNA sequencing (scRNA-seq), multi-omics datasets to analyze data in various aspects such as survival analysis, network analysis, biomarker identification, and differential co-expression analysis. It will be also interesting to see biologists using our algorithms on biological data and evaluating the outcomes.

Finally, we can say that technological advancement will continue to evolve and the magnitude of data will increase day by day. In such a scenario, we can not forget the impact of machine learning, data mining, or statistical analysis in investigating a large volume of complex data. Therefore, the contribution of computational biology will be undeniable in the biological era.

Bibliography

[1] Abu-Jamous, B. and Kelly, S. (2018). Clust: automatic extraction of optimal co-expressed gene clusters from gene expression data. *Genome biology*, 19(1):1–11.

[2] Adryan, B. and Schuh, R. (2004). Gene-ontology-based clustering of gene expression data. *Bioinformatics*, 20(16):2851–2852.

[3] Aguilar-Ruiz, J. S. (2005). Shifting and scaling patterns from gene expression data. *Bioinformatics*, 21(20):3840–3845.

[4] Ahmed, H., Mahanta, P., Bhattacharyya, D., Kalita, J., and Ghosh, A. (2011). Intersected coexpressed subcube miner: An effective triclustering algorithm. In *2011 World Congress on Information and Communication Technologies*, pages 846–851. IEEE.

[5] Al-Akwaa, F. M., Ali, M. H., and Kadah, Y. M. (2009). Bicat_plus: An automatic comparative tool for bi/clustering of gene expression data obtained using microarrays. In *Radio Science Conference, 2009. NRSC 2009. National*, pages 1–8. IEEE.

[6] Alfarsi, L. H., El Ansari, R., Masisi, B. K., Parks, R., Mohammed, O. J., Ellis, I. O., Rakha, E. A., and Green, A. R. (2020). Integrated analysis of key differentially expressed genes identifies dbn1 as a predictive marker of response to endocrine therapy in luminal breast cancer. *Cancers*, 12(6):1549.

[7] Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750.

[8] Alqurashi, N., Hashimi, S. M., Alowaidi, F., Ivanovski, S., Farag, A., and Wei, M. Q. (2019). mir-496, mir-1185, mir-654, mir-3183 and mir-495 are

downregulated in colorectal cancer cells and have putative roles in the mtor pathway. *Oncology letters*, 18(2):1657–1668.

[9] Alves, C. E. R., Cáceres, E. N., and Song, S. W. (2008). An all-substrings common subsequence algorithm. *Discrete Applied Mathematics*, 156(7):1025–1035.

[10] Amar, D., Yekutieli, D., Maron-Katz, A., Hendler, T., and Shamir, R. (2015). A hierarchical bayesian model for flexible module discovery in three-way time-series data. *Bioinformatics*, 31(12):i17–i26.

[11] Amjad, E., Asnaashari, S., Sokouti, B., and Dastmalchi, S. (2020). Systems biology comprehensive analysis on breast cancer for identification of key gene modules and genes associated with tnm-based clinical stages. *Scientific Reports*, 10(1):1–14.

[12] An, J., Liew, A. W.-C., and Nelson, C. C. (2012). Seed-based biclustering of gene expression data. *PloS one*, 7(8):e42431.

[13] An, O., Dall'Olio, G. M., Mourikis, T. P., and Ciccarelli, F. D. (2015). Ncg 5.0: updates of a manually curated repository of cancer genes and associated properties from cancer mutational screenings. *Nucleic acids research*, 44(D1):D992–D999.

[14] Angiulli, F., Cesario, E., and Pizzuti, C. (2008). Random walk biclustering for microarray data. *Information Sciences*, 178(6):1479–1497.

[15] Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60.

[16] Araujo, R., Trielli, G., Orair, G., Meira Jr, W., Ferreira, R., and Guedes, D. (2006). Partricluster: a scalable parallel algorithm for gene expression analysis. In *2006 18th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'06)*, pages 3–10. IEEE.

[17] Arif, K., Elliott, E. K., Haupt, L. M., and Griffiths, L. R. (2020). Regulatory mechanisms of epigenetic mirna relationships in human cancer and potential as therapeutic targets. *Cancers*, 12(10):2922.

[18] Armstrong, S. A., Staunton, J. E., Silverman, L. B., Pieters, R., den Boer, M. L., Minden, M. D., Sallan, S. E., Lander, E. S., Golub, T. R., and Ko-

rsmeyer, S. J. (2002). Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature genetics*, 30(1):41–47.

[19] Ashburner, M., Ball, C., Blake, J., et al. (2006). Gene ontology: tool for the unification of biology. the gene ontology consortium database resources of the national center for biotechnology information. *Nucleic Acids Research*, 34.

[20] Ayadi, W., Elloumi, M., and Hao, J.-K. (2009). A biclustering algorithm based on a bicluster enumeration tree: application to dna microarray data. *BioData mining*, 2(1):9.

[21] Ayadi, W., Elloumi, M., and Hao, J.-K. (2012). Bicfinder: a biclustering algorithm for microarray data analysis. *Knowledge and Information Systems*, 30(2):341–358.

[22] Azuaje, F. (2013). Bioinformatics and biomarker discovery:" omic" data analysis for personalized medicine. hoboken, new jersey; 2010. 1. abreu f., sousa aa, aronova ma et al. cryo-electron tomography of the magnetotactic vibrio magnetovibrio blakemorei: insights into the biomineralization of prismatic magneto–somes. *Journ. Struct. Biol*, 181(2):162–168.

[23] Bach, D.-H., Park, H. J., and Lee, S. K. (2018). The dual role of bone morphogenetic proteins in cancer. *Molecular Therapy-Oncolytics*, 8:1–13.

[24] Baik, I. H., Jo, G.-H., Seo, D., Ko, M. J., Cho, C. H., Lee, M. G., and Lee, Y.-H. (2016). Knockdown of rpl9 expression inhibits colorectal carcinoma growth via the inactivation of id-1/nf-κb signaling axis. *International journal of oncology*, 49(5):1953–1962.

[25] Balasubramaniyan, R., Hüllermeier, E., Weskamp, N., and Kämper, J. (2005). Clustering of gene expression data using a local shape-based similarity measure. *Bioinformatics*, 21(7):1069–1077.

[26] Ball, G. H. and Hall, D. J. (1965). Isodata, a novel method of data analysis and pattern classification. Technical report, Stanford research inst Menlo Park CA.

[27] Bastide, A. and David, A. (2018). The ribosome,(slow) beating heart of cancer (stem) cell. *Oncogenesis*, 7(4):1–13.

[28] Basu, N., Ingham, S., Hodson, J., Lalloo, F., Bulman, M., Howell, A., and Evans, D. (2015). Risk of contralateral breast cancer in brca1 and brca2 muta-

tion carriers: a 30-year semi-prospective analysis. *Familial cancer*, 14(4):531–538.

[29] Bee, A., Ke, Y., Forootan, S., Lin, K., Beesley, C., Forrest, S. E., and Foster, C. S. (2006). Ribosomal protein l19 is a prognostic marker for human prostate cancer. *Clinical Cancer Research*, 12(7):2061–2065.

[30] Bellaachia, A., Portnoy, D., Chen, Y., and Elkahloun, A. G. (2002). E-cast: A data mining algorithm for gene expression data. In *BIOKDD*, pages 49–54.

[31] Ben-Ari Fuchs, S., Lieder, I., Stelzer, G., Mazor, Y., Buzhor, E., Kaplan, S., Bogoch, Y., Plaschkes, I., Shitrit, A., Rappaport, N., et al. (2016). Geneanalytics: an integrative gene set analysis tool for next generation sequencing, rnaseq and microarray data. *Omics: a journal of integrative biology*, 20(3):139–151.

[32] Ben-Dor, A., Chor, B., Karp, R., and Yakhini, Z. (2003). Discovering local structure in gene expression data: the order-preserving submatrix problem. *Journal of computational biology*, 10(3-4):373–384.

[33] Ben-Dor, A., Shamir, R., and Yakhini, Z. (1999). Clustering gene expression patterns. *Journal of computational biology*, 6(3-4):281–297.

[34] Bendova, P., Pardini, B., Susova, S., Rosendorf, J., Levy, M., Skrobanek, P., Buchler, T., Kral, J., Liska, V., Vodickova, L., et al. (2021). Genetic variations in microrna-binding sites of solute carrier transporter genes as predictors of clinical outcome in colorectal cancer. *Carcinogenesis*, 42(3):378–394.

[35] Berriz, G. F., King, O. D., Bryant, B., Sander, C., and Roth, F. P. (2003). Characterizing gene sets with funcassociate. *Bioinformatics*, 19(18):2502–2504.

[36] Bhar, A., Haubrock, M., Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., and Wingender, E. (2012). δ-trimax: extracting triclusters and analysing coregulation in time series gene expression data. In *International Workshop on Algorithms in Bioinformatics*, pages 165–177. Springer.

[37] Bhar, A., Haubrock, M., Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., and Wingender, E. (2013). Coexpression and coregulation analysis of time-series gene expression data in estrogen-induced breast cancer cell. *Algorithms for molecular biology*, 8(1):9.

[38] Bhar, A., Haubrock, M., Mukhopadhyay, A., and Wingender, E. (2015). Multiobjective triclustering of time-series transcriptome data reveals key genes of biological processes. *BMC bioinformatics*, 16(1):200.

[39] Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., et al. (2001). Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences*, 98(24):13790–13795.

[40] Bhattacharya, A. and Cui, Y. (2017). A gpu-accelerated algorithm for biclustering analysis and detection of condition-dependent coexpression network modules. *Scientific Reports*, 7(1):4162.

[41] Binns, D., Dimmer, E., Huntley, R., Barrell, D., O'donovan, C., and Apweiler, R. (2009). Quickgo: a web-based tool for gene ontology searching. *Bioinformatics*, 25(22):3045–3046.

[42] Bisgin, H., Gong, B., Wang, Y., and Tong, W. (2018). Evaluation of bioinformatics approaches for next-generation sequencing analysis of microrna s with a toxicogenomics study design. *Frontiers in genetics*, 9:22.

[43] Bohnsack, M. T., Czaplinski, K., and GÖRLICH, D. (2004). Exportin 5 is a rangtp-dependent dsrna-binding protein that mediates nuclear export of pre-mirnas. *Rna*, 10(2):185–191.

[44] Bonin, S., Pracella, D., Barbazza, R., Dotti, I., Boffo, S., and Stanta, G. (2019). Pi3k/akt signaling in breast cancer molecular subtyping and lymph node involvement. *Disease markers*, 2019.

[45] Botchkareva, N. V. (2017). The molecular revolution in cutaneous biology: noncoding rnas: new molecular players in dermatology and cutaneous biology. *Journal of Investigative Dermatology*, 137(5):e105–e111.

[46] Boullé, M. (2011). Data grid models for preparation and modeling in supervised learning. *Hands-On Pattern Recognition: Challenges in Machine Learning*, 1:99–130.

[47] Bozdağ, D., Kumar, A. S., and Catalyurek, U. V. (2010). Comparative analysis of biclustering algorithms. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 265–274. ACM.

[48] Brady-West, D. C. and McGrowder, D. A. (2011). Triple negative breast cancer: therapeutic and prognostic implications. *Asian Pac J Cancer Prev*, 12(8):2139–2143.

[49] Brameier, M. and Wiuf, C. (2007). Co-clustering and visualization of gene expression data and gene ontology terms for saccharomyces cerevisiae using self-organizing maps. *Journal of biomedical informatics*, 40(2):160–173.

[50] Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A., and Jemal, A. (2018). Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 68(6):394–424.

[51] Bryan, J. (2004). Problems in gene clustering based on gene expression data. *Journal of Multivariate Analysis*, 90(1):44–66.

[52] Çağatay, T. and Chook, Y. M. (2018). Karyopherins in cancer. *Current opinion in cell biology*, 52:30–42.

[53] Campello, R. J. G. B. and Hruschka, E. R. (2009). On comparing two sequences of numbers and its applications to clustering analysis. *Information Sciences*, 179(8):1025–1039.

[54] Cao, Z.-G., Li, J.-J., Yao, L., Huang, Y.-N., Liu, Y.-R., Hu, X., Song, C.-G., and Shao, Z.-M. (2016). High expression of microrna-454 is associated with poor prognosis in triple-negative breast cancer. *Oncotarget*, 7(40):64900.

[55] Cary, M. P., Bader, G. D., and Sander, C. (2005). Pathway information for systems biology. *FEBS letters*, 579(8):1815–1820.

[56] Cha, K., Hwang, T., Oh, K., and Yi, G. (2015). Discovering transnosological molecular basis of human brain diseases using biclustering analysis of integrated gene expression data. *BMC Med. Inf. & Decision Making*, 15(S-1):S7.

[57] Cha, K., Oh, K., Hwang, T., and Yi, G. (2014). Identification of coexpressed gene modules across multiple brain diseases by a biclustering analysis on integrated gene expression data. In *Proceedings of the ACM 8th International Workshop on Data and Text Mining in Bioinformatics, DTMBIO@CIKM 2014, Shanghai, China, November 7, 2014*, page 17.

[58] Chan, W.-C., Ho, M.-R., Li, S.-C., Tsai, K.-W., Lai, C.-H., Hsu, C.-N., and Lin, W.-c. (2012). Metamirclust: discovery of mirna cluster patterns using a data-mining approach. *Genomics*, 100(3):141–148.

[59] Chen, A. H., Tsau, Y.-W., and Lin, C.-H. (2010). Novel methods to identify biologically relevant genes for leukemia and prostate cancer from gene expression profiles. *BMC genomics*, 11(1):274.

[60] Chen, X. (2009). Curve-based clustering of time course gene expression data using self-organizing maps. *J. Bioinformatics and Computational Biology*, 7(4):645–661.

[61] Chen, X., Cao, Q., Liao, R., Wu, X., Xun, S., Huang, J., and Dong, C. (2019). Loss of abat-mediated gabaergic system promotes basal-like breast cancer progression by activating ca2+-nfat1 axis. *Theranostics*, 9(1):34.

[62] Chen, Z., Huang, Z., Luo, Y., Zou, Q., Bai, L., Tang, G., Wang, X., Cao, G., Huang, M., Xiang, J., et al. (2021). Genome-wide analysis identifies critical dna methylations within ntrks genes in colorectal cancer. *Journal of translational medicine*, 19(1):1–13.

[63] Cheng, K.-O., Law, N.-F., Siu, W.-C., and Liew, A. W. (2008). Identification of coherent patterns in gene expression data using an efficient biclustering algorithm and parallel coordinate visualization. *BMC bioinformatics*, 9(1):210.

[64] Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. In *Ismb*, volume 8, pages 93–103.

[65] Cheung, L., Cheung, D. W., Kao, B., Yip, K. Y., and Ng, M. K. (2006). On mining micro-array data by order-preserving submatrix. *International Journal of Bioinformatics Research and Applications*, 3(1):42–64.

[66] Chia, B. K. H. and Karuturi, R. K. M. (2010). Differential co-expression framework to quantify goodness of biclusters and compare biclustering algorithms. *Algorithms for molecular biology*, 5(1):23.

[67] Chui, C. K., Kao, B., Yip, K. Y., and Lee, S. D. (2008). Mining order-preserving submatrices from data with repeated measurements. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 133–142. IEEE.

[68] Coccaro, N., Tota, G., Zagaria, A., Anelli, L., Specchia, G., and Albano, F. (2017). Setbp1 dysregulation in congenital disorders and myeloid neoplasms. *Oncotarget*, 8(31):51920.

[69] Corizzo, R., Pio, G., Ceci, M., and Malerba, D. (2019). Dencast: Distributed density-based clustering for multi-target regression. *Journal of Big Data*, 6(1):43.

[70] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). Dynamic programming. *Introduction to Algorithms,*, pages 323–370.

[71] Costa, D. C., de Oliveira, G. A., Cino, E. A., Soares, I. N., Rangel, L. P., and Silva, J. L. (2016). Aggregation and prion-like properties of misfolded tumor suppressors: is cancer a prion disease? *Cold Spring Harbor perspectives in biology*, 8(10):a023614.

[72] Couto, F. M., Silva, M. J., and Coutinho, P. M. (2005). Semantic similarity over the gene ontology: family correlation and selecting disjunctive ancestors. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 343–344.

[73] Croce, C. M. (2009). Causes and consequences of microrna dysregulation in cancer. *Nature reviews genetics*, 10(10):704.

[74] Das, R., Bhattacharyya, D., and Kalita, J. (2010). Clustering gene expression data using an effective dissimilarity measure. *International Journal of Computational BioScience (Special Issue)*, 1(1):55–68.

[75] Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227.

[76] de Souto, M. C., Costa, I. G., de Araujo, D. S., Ludermir, T. B., and Schliep, A. (2008). Clustering cancer gene expression data: a comparative study. *BMC bioinformatics*, 9(1):497.

[77] De Troyer, E. (2014). Software development for biclustering methods: The biclust gui r package. Master's thesis, tUL.

[78] De Troyer, E. (2016). Bibitr r package: Ar wrapper for bibit.

[79] Dede, D. and Oğul, H. (2013). A three-way clustering approach to cross-species gene regulation analysis. In *2013 IEEE INISTA*, pages 1–5. IEEE.

[80] Dede, D. and Oğul, H. (2014). Triclust: A tool for cross-species analysis of gene regulation. *Molecular informatics*, 33(5):382–387.

[81] Demidyuk, I. V., Shubin, A. V., Gasanov, E. V., Kurinov, A. M., Demkin, V. V., Vinogradova, T. V., Zinovyeva, M. V., Sass, A. V., Zborovskaya, I. B., and Kostrov, S. V. (2013). Alterations in gene expression of proprotein convertases in human lung cancer have a limited number of scenarios. *PLoS One*, 8(2):e55752.

[82] Deodhar, M., Gupta, G., Ghosh, J., Cho, H., and Dhillon, I. (2009). A scalable framework for discovering coherent co-clusters in noisy data. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 241–248. ACM.

[83] Deryusheva, I., Tsyganov, M., Garbukov, E., Ibragimova, M., Kzhyshkovska, J. G., Slonimskaya, E., Cherdyntseva, N., and Litviakov, N. (2017). Genome-wide association study of loss of heterozygosity and metastasis-free survival in breast cancer patients. *Experimental oncology*.

[84] Desgraupes, B. (2013). Clustering indices. *University of Paris Ouest-Lab Modal'X*, 1:34.

[85] D'haeseleer, P. (2005). How does gene expression clustering work? *Nature biotechnology*, 23(12):1499–1501.

[86] D'haeseleer, P., Liang, S., and Somogyi, R. (2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726.

[87] Dharan, S. and Nair, A. S. (2009). Biclustering of gene expression data using reactive greedy randomized adaptive search procedure. *BMC bioinformatics*, 10(Suppl 1):S27.

[88] Dhillon, I. S., Marcotte, E. M., and Roshan, U. (2003). Diametrical clustering for identifying anti-correlated gene clusters. *Bioinformatics*, 19(13):1612–1619.

[89] Di Gesú, V., Giancarlo, R., Bosco, G. L., Raimondi, A., and Scaturro, D. (2005). Genclust: A genetic algorithm for clustering gene expression data. *BMC bioinformatics*, 6(1):289.

[90] Dolezal, J. M., Dash, A. P., and Prochownik, E. V. (2018). Diagnostic and prognostic implications of ribosomal protein transcript expression patterns in human cancers. *BMC cancer*, 18(1):1–14.

[91] Doungpan, N., Engchuan, W., Chan, J. H., and Meechai, A. (2016). Gsnfs: Gene subnetwork biomarker identification of lung cancer expression data. *BMC medical genomics*, 9(3):70.

[92] Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104.

[93] Dussaut, J. S., Gallo, C. A., Cecchini, R. L., Carballido, J. A., and Ponzoni, I. (2016). Crosstalk pathway inference using topological information and biclustering of gene expression data. *Biosystems*, 150:1–12.

[94] Edla, D. R., Jana, P. K., and Member, I. S. (2012). A prototype-based modified dbscan for gene clustering. *Procedia Technology*, 6:485–492.

[95] Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868.

[96] El Baiomy, M. A. and El Kashef, W. F. (2017). Ercc1 expression in metastatic triple negative breast cancer patients treated with platinum-based chemotherapy. *Asian Pacific Journal of Cancer Prevention*, 18(2):507–513.

[97] Erbes, T., Hirschfeld, M., Rücker, G., Jaeger, M., Boas, J., Iborra, S., Mayer, S., Gitsch, G., and Stickeler, E. (2015). Feasibility of urinary microrna detection in breast cancer patients and its potential as an innovative non-invasive biomarker. *BMC cancer*, 15(1):193.

[98] Eren, K., Deveci, M., Küçüktunç, O., and Çatalyürek, Ü. V. (2012). A comparative analysis of biclustering algorithms for gene expression data. *Briefings in bioinformatics*, 14(3):279–292.

[99] Eren, K., Deveci, M., Küçüktunç, O., and Çatalyürek, Ü. V. (2013). A comparative analysis of biclustering algorithms for gene expression data. *Briefings in bioinformatics*, 14(3):279–292.

[100] Erola, P., Björkegren, J. L., and Michoel, T. (2020). Model-based clustering of multi-tissue gene expression data. *Bioinformatics*, 36(6):1807–1813.

[101] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

[102] Eswaran, J., Cyanam, D., Mudvari, P., Reddy, S. D. N., Pakala, S. B., Nair, S. S., Florea, L., Fuqua, S. A., Godbole, S., and Kumar, R. (2012). Transcriptomic landscape of breast cancers through mrna sequencing. *Scientific reports*, 2:264.

[103] Fan, Y., Long, B., and Pennington, S. R. (2011). Bioinformatics and biomarker discovery:"omic" data analysis for personalized medicine francisco azuaje wiley-blackwell, 2010, p. 248 isbn: 978-0-470-74460-4. *Proteomics*, 11(22):4439–4439.

[104] Fane, M., Harris, L., Smith, A. G., and Piper, M. (2017). Nuclear factor one transcription factors as epigenetic regulators in cancer. *International journal of cancer*, 140(12):2634–2641.

[105] Fang, Q., Ng, W., and Feng, J. (2010). Discovering significant relaxed order-preserving submatrices. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 433–442. ACM.

[106] Fang, Q., Ng, W., Feng, J., and Li, Y. (2012). Mining bucket order-preserving submatrices in gene expression data. *IEEE transactions on knowledge and data engineering*, 24(12):2218–2231.

[107] Fang, Q., Ng, W., Feng, J., and Li, Y. (2014). Mining order-preserving submatrices from probabilistic matrices. *ACM Transactions on Database Systems (TODS)*, 39(1):6.

[108] Farazi, T. A., Horlings, H. M., ten Hoeve, J., Mihailovic, A., Halfwerk, H., Morozov, P., Brown, M., Hafner, M., Reyal, F., van Kouwenhove, M., et al. (2011). Microrna sequence and expression analysis in breast tumors by deep sequencing. *Cancer research*, pages canres–0608.

[109] Fiannaca, A., La Rosa, M., La Paglia, L., Rizzo, R., and Urso, A. (2015). Analysis of mirna expression profiles in breast cancer using biclustering. *BMC bioinformatics*, 16(4):S7.

[110] Flores, J. L., Inza, I., Larrañaga, P., and Calvo, B. (2013). A new measure for gene expression biclustering based on non-parametric correlation. *Computer methods and programs in biomedicine*, 112(3):367–397.

[111] Forti, A. and Foresti, G. L. (2006). Growing hierarchical tree som: An unsupervised neural network with dynamic topology. *Neural networks*, 19(10):1568–1580.

[112] Freitas, A., Ayadi, W., Elloumi, M., Luıs, J., and Oliveira, J.-K. H. (2012). Survey on biclustering of gene expression data. *Biological Knowl. Disc. Handbook*, pages 591–608.

[113] Gao, B. J., Griffith, O. L., Ester, M., Xiong, H., Zhao, Q., and Jones, S. J. (2012). On the deep order-preserving submatrix problem: A best effort approach. *IEEE transactions on knowledge and data engineering*, 24(2):309–325.

[114] Gao, Y., Wang, Y., Wang, X., Zhao, C., Wang, F., Du, J., Zhang, H., Shi, H., Feng, Y., Li, D., et al. (2021). mir-335-5p suppresses gastric cancer progression by targeting mapk10. *Cancer Cell International*, 21(1):1–12.

[115] Gawrychowski, P. and Uznański, P. (2016). Order-preserving pattern matching with k mismatches. *Theoretical Computer Science*, 638:136–144.

[116] Getz, G., Gal, H., Kela, I., Notterman, D. A., and Domany, E. (2003). Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data. *Bioinformatics*, 19(9):1079–1089.

[117] Gnatyshak, D. V. (2014). Greedy modifications of oac-triclustering algorithm. *Procedia Computer Science*, 31:1116–1123.

[118] Gong, X., Dong, T., Niu, M., Liang, X., Sun, S., Zhang, Y., Li, Y., and Li, D. (2020). Lncrna lcpat1 upregulation promotes breast cancer progression via enhancing mfap2 transcription. *Molecular Therapy-Nucleic Acids*, 21:804–813.

[119] Goodman, L. A. and Kruskal, W. H. (1954). Measures of association for cross classifications*. *Journal of the American Statistical Association*, 49(268):732–764.

[120] Goossens, N., Nakagawa, S., Sun, X., and Hoshida, Y. (2015). Cancer biomarker discovery and validation. *Translational cancer research*, 4(3):256.

[121] Graham, K., de Las Morenas, A., Tripathi, A., King, C., Kavanah, M., Mendez, J., Stone, M., Slama, J., Miller, M., Antoine, G., et al. (2010). Gene expression in histologically normal epithelium from breast cancer patients and from cancer-free prophylactic mastectomy patients shares a similar profile. *British journal of cancer*, 102(8):1284–1293.

[122] Guigourès, R., Boullé, M., and Rossi, F. (2018). Discovering patterns in time-varying graphs: a triclustering approach. *Advances in Data Analysis and Classification*, 12(3):509–536.

[123] Guo, L., Yang, S., Zhao, Y., Zhang, H., Wu, Q., and Chen, F. (2014). Global analysis of mirna gene clusters and gene families reveals dynamic and coordinated expression. *BioMed research international*, 2014.

[124] Gusenleitner, D., Howe, E. A., Bentink, S., Quackenbush, J., and Culhane, A. C. (2012). ibbig: iterative binary bi-clustering of gene sets. *Bioinformatics*, 28(19):2484–2492.

[125] Gutiérrez-Avilés, D. and Rubio-Escudero, C. (2015). Msl: a measure to evaluate three-dimensional patterns in gene expression data. *Evolutionary Bioinformatics*, 11:EBO–S25822.

[126] Gutiérrez-Avilés, D., Rubio-Escudero, C., Martínez-Álvarez, F., and Riquelme, J. C. (2014). Trigen: A genetic algorithm to mine triclusters in temporal gene expression data. *Neurocomputing*, 132:42–53.

[127] Hamam, R., Hamam, D., Alsaleh, K. A., Kassem, M., Zaher, W., Alfayez, M., Aldahmash, A., and Alajez, N. M. (2017). Circulating micrornas in breast cancer: novel diagnostic and prognostic biomarkers. *Cell death & disease*, 8(9):e3045.

[128] Hang, S., You, Z., and Chun, L. Y. (2009). Incorporating biological knowledge into density-based clustering analysis of gene expression data. In *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on*, volume 5, pages 52–56. IEEE.

[129] Hartigan, J. A. (1972). Direct clustering of a data matrix. *Journal of the american statistical association*, 67(337):123–129.

[130] Hatzis, C., Sun, H., Yao, H., Hubbard, R. E., Meric-Bernstam, F., Babiera, G. V., Wu, Y., Pusztai, L., and Symmans, W. F. (2011). Effects of tissue

handling on rna integrity and microarray measurements from resected breast cancers. *Journal of the National Cancer Institute*, 103(24):1871–1883.

[131] Hautaniemi, S., Yli-Harja, O., Astola, J., Kauraniemi, P., Kallioniemi, A., Wolf, M., Ruiz, J., Mousses, S., and Kallioniemi, O. (2003). Analysis and visualization of gene expression microarray data in human cancer using self-organizing maps. *Machine Learning*, 52(1-2):45–66.

[132] Henriques, R., Ferreira, F. L., and Madeira, S. C. (2017). Bicpams: software for biological data analysis with pattern-based biclustering. *BMC bioinformatics*, 18(1):82.

[133] Henriques, R. and Madeira, S. C. (2014a). Bicpam: Pattern-based biclustering for biomedical data analysis. *Algorithms for Molecular Biology*, 9(1):27.

[134] Henriques, R. and Madeira, S. C. (2014b). Bicspam: flexible biclustering using sequential patterns. *BMC bioinformatics*, 15(1):130.

[135] Henriques, R. and Madeira, S. C. (2016). Bic2pam: constraint-guided biclustering for biological data analysis with domain knowledge. *Algorithms for Molecular Biology*, 11(1):23.

[136] Henriques, R. and Madeira, S. C. (2018). Triclustering algorithms for three-dimensional data analysis: a comprehensive survey. *ACM Computing Surveys (CSUR)*, 51(5):1–43.

[137] Henry, N. L. and Hayes, D. F. (2012). Cancer biomarkers. *Molecular oncology*, 6(2):140–146.

[138] Herrero, J., Valencia, A., and Dopazo, J. (2001). A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics*, 17(2):126–136.

[139] Heyer, L. J., Kruglyak, S., and Yooseph, S. (1999). Exploring expression data: identification and analysis of coexpressed genes. *Genome research*, 9(11):1106–1115.

[140] Higuera, C., Gardiner, K. J., and Cios, K. J. (2015). Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PloS one*, 10(6).

[141] Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., Kasim, A., Khamiakova, T., Van Sanden, S., Lin, D., Talloen, W., et al. (2010).

Fabia: factor analysis for bicluster acquisition. *Bioinformatics*, 26(12):1520–1527.

[142] Houari, A., Ayadi, W., and Yahia, S. B. (2017). Mining negative correlation biclusters from gene expression data using generic association rules. *Procedia computer science*, 112:278–287.

[143] Huang, D. and Pan, W. (2006). Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. *Bioinformatics*, 22(10):1259–1268.

[144] Huang, D. W., Sherman, B. T., Tan, Q., Collins, J. R., Alvord, W. G., Roayaei, J., Stephens, R., Baseler, M. W., Lane, H. C., and Lempicki, R. A. (2007). The david gene functional classification tool: a novel biological module-centric algorithm to functionally analyze large gene lists. *Genome biology*, 8(9):R183.

[145] Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.

[146] Hubert, L. and Schultz, J. (1976). Quadratic assignment as a general data analysis strategy. *British journal of mathematical and statistical psychology*, 29(2):190–241.

[147] Hussain, S. F. and Ramazan, M. (2016). Biclustering of human cancer microarray data using co-similarity based co-clustering. *Expert Systems with Applications*, 55:520–531.

[148] Ignatov, D. I., Gnatyshak, D. V., Kuznetsov, S. O., and Mirkin, B. G. (2015). Triadic formal concept analysis and triclustering: searching for optimal patterns. *Machine Learning*, 101(1):271–302.

[149] Ignatov, D. I. and Kuznetsov, S. O. (2009). Frequent itemset mining for clustering near duplicate web documents. In *International Conference on Conceptual Structures*, pages 185–200. Springer.

[150] Ignatov, D. I., Kuznetsov, S. O., Magizov, R. A., and Zhukov, L. E. (2011). From triconcepts to triclusters. In *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, pages 257–264. Springer.

[151] Ignatov, D. I., Kuznetsov, S. O., Poelmans, J., and Zhukov, L. E. (2013). Can triconcepts become triclusters? *International Journal of General Systems*, 42(6):572–593.

[152] Ihmels, J., Bergmann, S., and Barkai, N. (2004). Defining transcription modules using large-scale gene expression data. *Bioinformatics*, 20(13):1993–2003.

[153] Imperiale, T. F., Ransohoff, D. F., Itzkowitz, S. H., Levin, T. R., Lavin, P., Lidgard, G. P., Ahlquist, D. A., and Berger, B. M. (2014). Multitarget stool dna testing for colorectal-cancer screening. *New England Journal of Medicine*, 370(14):1287–1297.

[154] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.

[155] Jaschke, R., Hotho, A., Schmitz, C., Ganter, B., and Stumme, G. (2006). Trias–an algorithm for mining iceberg tri-lattices. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 907–911. IEEE.

[156] Jaskowiak, P. A., Campello, R. J., and Costa, I. G. (2012). Evaluating correlation coefficients for clustering gene expression profiles of cancer. In *Brazilian Symposium on Bioinformatics*, pages 120–131. Springer.

[157] Jaskowiak, P. A., Campello, R. J., and Costa, I. G. (2014). On the selection of appropriate distances for gene expression data clustering. *BMC bioinformatics*, 15(Suppl 2):S2.

[158] Jaskowiak, P. A., Campello, R. J., and Costa Filho, I. G. (2013). Proximity measures for clustering gene expression microarray data: a validation methodology and a comparative analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 10(4):845–857.

[159] Ji, L. and Tan, K.-L. (2004). Mining gene expression data for positive and negative co-regulated gene clusters. *Bioinformatics*, 20(16):2711–2718.

[160] Ji, L., Tan, K.-L., and Tung, A. K. (2006). Mining frequent closed cubes in 3d datasets. In *Proceedings of the 32nd international conference on very large data bases*, pages 811–822.

[161] Jiang, D., Pei, J., Ramanathan, M., Tang, C., and Zhang, A. (2004a). Mining coherent gene clusters from gene-sample-time microarray data. In

Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 430–439.

[162] Jiang, D., Pei, J., and Zhang, A. (2003). Dhc: a density-based hierarchical clustering method for time series gene expression data. In *Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium on*, pages 393–400. IEEE.

[163] Jiang, D., Pei, J., and Zhang, A. (2004b). Gpx: interactive mining of gene expression data. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 1249–1252. VLDB Endowment.

[164] Jiang, D., Tang, C., and Zhang, A. (2004c). Cluster analysis for gene expression data: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 16(11):1370–1386.

[165] Jiang, H., Zhou, S., Guan, J., and Zheng, Y. (2006). gtricluster: a more general and effective 3d clustering algorithm for gene-sample-time microarray data. In *International Workshop on Data Mining for Biomedical Applications*, pages 48–59. Springer.

[166] Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.

[167] Jiang, L., Wang, Y., Liu, G., Liu, H., Zhu, F., Ji, H., and Li, B. (2018). C-phycocyanin exerts anti-cancer effects via the mapk signaling pathway in mda-mb-231 cells. *Cancer Cell International*, 18(1):1–14.

[168] Joe, S. and Nam, H. (2016). Prognostic factor analysis for breast cancer using gene expression profiles. *BMC Medical Informatics and Decision Making*, 16(1):56.

[169] Jung, I., Jo, K., Kang, H., Ahn, H., Yu, Y., and Kim, S. (2017). Timesvector: a vectorized clustering approach to the analysis of time series transcriptome data from multiple phenotypes. *Bioinformatics*, 33(23):3827–3835.

[170] Kaiser, S., Santamaria, R., Khamiakova, T., Sill, M., Theron, R., Quintales, L., Leisch, F., and De Troyer, E. (2015). biclust: Bicluster algorithms, 2015. *R package version*, 1(0).

[171] Kakati, T., Ahmed, H. A., Bhattacharyya, D. K., and Kalita, J. K. (2016). A fast gene expression analysis using parallel biclustering and distributed triclustering approach. In *Proceedings of the Second International Conference on*

Information and Communication Technology for Competitive Strategies, pages 1–6.

[172] Kanehisa, M. and Goto, S. (2000). Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30.

[173] Kendall, M. G. (1948). Rank correlation methods.

[174] Kerr, G., Ruskin, H. J., Crane, M., and Doolan, P. (2008). Techniques for clustering gene expression data. *Computers in biology and medicine*, 38(3):283–293.

[175] Kim, H., Watkinson, J., and Anastassiou, D. (2011). Biomarker discovery using statistically significant gene sets. *Journal of Computational Biology*, 18(10):1329–1338.

[176] Kim, H.-K., Bhattarai, K. R., Junjappa, R. P., Ahn, J. H., Pagire, S. H., Yoo, H. J., Han, J., Lee, D., Kim, K.-W., Kim, H.-R., et al. (2020). Tmbim6/bi-1 contributes to cancer progression through assembly with mtorc2 and akt activation. *Nature communications*, 11(1):1–16.

[177] Kim, J., Eades, P., Fleischer, R., Hong, S.-H., Iliopoulos, C. S., Park, K., Puglisi, S. J., and Tokuyama, T. (2014). Order-preserving matching. *Theoretical Computer Science*, 525:68–79.

[178] Kim, S., Kon, M., and DeLisi, C. (2012). Pathway-based classification of cancer subtypes. *Biology direct*, 7(1):21.

[179] Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480.

[180] Kreunin, P., Yoo, C., Urquidi, V., Lubman, D. M., and Goodison, S. (2007). Differential expression of ribosomal proteins in a human metastasis model identified by coupling 2-d liquid chromatography and mass spectrometry. *Cancer Genomics-Proteomics*, 4(5):329–339.

[181] Krolak-Schwerdt, S., Orlik, P., and Ganter, B. (1994). Tripat: a model for analyzing three-mode binary data. In *Information Systems and Data Analysis*, pages 298–307. Springer.

[182] Kubica, M., Kulczyński, T., Radoszewski, J., Rytter, W., and Waleń, T. (2013). A linear time algorithm for consecutive permutation pattern matching. *Information Processing Letters*, 113(12):430–433.

[183] Kulshrestha, A., Suman, S., and Ranjan, R. (2016). Network analysis reveals potential markers for pediatric adrenocortical carcinoma. *OncoTargets and therapy*, 9:4569.

[184] Laiho, P., Kokko, A., Vanharanta, S., Salovaara, R., Sammalkorpi, H., Järvinen, H., Mecklin, J., Karttunen, T., Tuppurainen, K., Davalos, V., et al. (2007). Serrated carcinomas form a subclass of colorectal cancer with distinct molecular basis. *Oncogene*, 26(2):312–320.

[185] Lam, Y., Tsang, P. W., and Leung, C. (2013). Pso-based k-means clustering with enhanced cluster matching for gene expression data. *Neural Computing and Applications*, 22(7-8):1349–1355.

[186] Langfelder, P. and Horvath, S. (2007). Eigengene networks for studying the relationships between co-expression modules. *BMC systems biology*, 1(1):1–17.

[187] Larrea, E., Sole, C., Manterola, L., Goicoechea, I., Armesto, M., Arestin, M., Caffarel, M. M., Araujo, A. M., Araiz, M., Fernandez-Mercado, M., et al. (2016). New concepts in cancer biomarkers: circulating mirnas in liquid biopsies. *International journal of molecular sciences*, 17(5):627.

[188] Lazzeroni, L., Owen, A., et al. (2002). Plaid models for gene expression data. *Statistica sinica*, 12(1):61–86.

[189] Leale, G., Bayá, A. E., Milone, D. H., Granitto, P. M., and Stegmayer, G. (2018). Inferring unknown biological function by integration of go annotations and gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(1):168–180.

[190] Ledermann, J., Harter, P., Gourley, C., Friedlander, M., Vergote, I., Rustin, G., Scott, C., Meier, W., Shapira-Frommer, R., Safra, T., et al. (2012). Olaparib maintenance therapy in platinum-sensitive relapsed ovarian cancer. *New England Journal of Medicine*, 366(15):1382–1392.

[191] Lee, R. C., Feinbaum, R. L., and Ambros, V. (1993). The c. elegans heterochronic gene lin-4 encodes small rnas with antisense complementarity to lin-14. *cell*, 75(5):843–854.

[192] Lee, W.-P. and Lin, C.-H. (2016). Combining expression data and knowledge ontology for gene clustering and network reconstruction. *Cognitive Computation*, 8(2):217–227.

[193] Lehmann, F. and Wille, R. (1995). A triadic approach to formal concept analysis. In *International Conference on Conceptual Structures*, pages 32–43. Springer.

[194] Li, A. and Tuck, D. (2009). An effective tri-clustering algorithm combining expression data with gene regulation information. *Gene regulation and systems biology*, 3:GRSB–S1150.

[195] Li, G., Li, M., Liang, X., Xiao, Z., Zhang, P., Shao, M., Peng, F., Chen, Y., Li, Y., and Chen, Z. (2017a). Identifying dcn and hspd1 as potential biomarkers in colon cancer using 2d-lc-ms/ms combined with itraq technology. *Journal of Cancer*, 8(3):479.

[196] Li, G., Ma, Q., Tang, H., Paterson, A. H., and Xu, Y. (2009). Qubic: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic acids research*, 37(15):e101–e101.

[197] Li, H., Qiu, Z., Li, F., and Wang, C. (2017b). The relationship between mmp-2 and mmp-9 expression levels with breast cancer incidence and prognosis. *Oncology letters*, 14(5):5865–5870.

[198] Li, J., Ma, S., Lin, T., Li, Y., Yang, S., Zhang, W., Zhang, R., and Wang, Y. (2019a). Comprehensive analysis of therapy-related messenger rnas and long noncoding rnas as novel biomarkers for advanced colorectal cancer. *Frontiers in genetics*, 10:803.

[199] Li, L., Guo, Y., Wu, W., Shi, Y., Cheng, J., and Tao, S. (2012). A comparison and evaluation of five biclustering algorithms by quantifying goodness of biclusters for gene expression data. *BioData mining*, 5(1):1–10.

[200] Li, L., Sun, F., Chen, X., and Zhang, M. (2018a). Isl1 is upregulated in breast cancer and promotes cell proliferation, invasion, and angiogenesis. *OncoTargets and therapy*, 11:781.

[201] Li, Q., Su, Y.-L., Zeng, M., and Shen, W.-X. (2018b). Enabled homolog shown to be a potential biomarker and prognostic indicator for breast cancer by bioinformatics analysis. *Clinical and Investigative Medicine*, 41(4):E186–E195.

[202] Li, W., Huang, K., Guo, H., Cui, G., and Zhao, S. (2014). Inhibition of non-small-cell lung cancer cell proliferation by pbx1. *Chinese Journal of Cancer Research*, 26(5):573.

[203] Li, W.-h., Zhang, H., Guo, Q., Wu, X.-d., Xu, Z.-s., Dang, C.-x., Xia, P., and Song, Y.-c. (2015). Detection of snca and fbn1 methylation in the stool as a biomarker for colorectal cancer. *Disease markers*, 2015.

[204] Li, X., Xu, M., Ding, L., and Tang, J. (2019b). Mir-27a: a novel biomarker and potential therapeutic target in tumors. *Journal of Cancer*, 10(12):2836.

[205] Li, Y., Qiu, C., Tu, J., Geng, B., Yang, J., Jiang, T., and Cui, Q. (2013a). Hmdd v2. 0: a database for experimentally supported human microrna and disease associations. *Nucleic acids research*, 42(D1):D1070–D1074.

[206] Li, Z., Herold, T., He, C., Valk, P. J., Chen, P., Jurinovic, V., Mansmann, U., Radmacher, M. D., Maharry, K. S., Sun, M., et al. (2013b). Identification of a 24-gene prognostic signature that improves the european leukemianet risk classification of acute myeloid leukemia: an international collaborative study. *Journal of Clinical Oncology*, 31(9):1172–1181.

[207] Lin, D. (1998). An information-theoretic definition of similarity. In Shavlik, J. W., editor, *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998*, pages 296–304. Morgan Kaufmann.

[208] Lin, W., Feng, M., Li, X., Zhong, P., Guo, A., Chen, G., Xu, Q., and Ye, Y. (2017). Transcriptome profiling of cancer and normal tissues from cervical squamous cancer patients by deep sequencing. *Molecular medicine reports*, 16(2):2075–2088.

[209] Liu, B., Xin, Y., Cheung, R. C., and Yan, H. (2014). Gpu-based biclustering for microarray data analysis in neurocomputing. *Neurocomputing*, 134:239–246.

[210] Liu, J., Jennings, S. F., Tong, W., and Hong, H. (2011). Next generation sequencing for profiling expression of mirnas: technical progress and applications in drug development. *Journal of biomedical science and engineering*, 4(10):666.

[211] Liu, J., Jing, L., and Tu, X. (2016). Weighted gene co-expression network analysis identifies specific modules and hub genes related to coronary artery disease. *BMC cardiovascular disorders*, 16(1):54.

[212] Liu, J., Li, Z., Hu, X., and Chen, Y. (2008). Multi-objective evolutionary algorithm for mining 3d clusters in gene-sample-time microarray data. In *2008 IEEE International Conference on Granular Computing*, pages 442–447. IEEE.

[213] Liu, J., Shen, J.-X., Wu, H.-T., Li, X.-L., Wen, X.-F., Du, C.-W., and Zhang, G.-J. (2018). Collagen 1a1 (col1a1) promotes metastasis of breast cancer and is a potential therapeutic target. *Discovery medicine*, 25(139):211–223.

[214] Liu, J., Wang, W., and Yang, J. (2004). Gene ontology friendly biclustering of expression profiles. In *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*, pages 436–447. IEEE.

[215] Liu, K.-Q., Liu, Z.-P., Hao, J.-K., Chen, L., and Zhao, X.-M. (2012). Identifying dysregulated pathways in cancers from pathway interaction networks. *BMC bioinformatics*, 13(1):1.

[216] Liu, X., Chen, L., Huang, H., Lv, J.-M., Chen, M., Qu, F.-J., Pan, X.-W., Li, L., Yin, L., Cui, X.-G., et al. (2017). High expression of pdlim5 facilitates cell tumorigenesis and migration by maintaining ampk activation in prostate cancer. *Oncotarget*, 8(58):98117.

[217] Liu, X. and Wang, L. (2007). Computing the maximum similarity bi-clusters of gene expression data. *Bioinformatics*, 23(1):50–56.

[218] Lord, P. W., Stevens, R. D., Brass, A., and Goble, C. A. (2003). Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275–1283.

[219] Lu, Y., Lu, S., Fotouhi, F., Deng, Y., and Brown, S. J. (2004). Incremental genetic k-means algorithm and its application in gene expression data analysis. *BMC Bioinformatics*, 5:172.

[220] Lundberg, A., Lindström, L. S., Li, J., Harrell, J. C., Darai-Ramqvist, E., Sifakis, E. G., Foukakis, T., Perou, C. M., Czene, K., Bergh, J., et al. (2019). The long-term prognostic and predictive capacity of cyclin d1 gene amplification in 2305 breast tumours. *Breast Cancer Research*, 21(1):34.

[221] Luo, F., Khan, L., Bastani, F., Yen, I.-L., and Zhou, J. (2004). A dynamically growing self-organizing tree (dgsot) for hierarchical clustering gene expression profiles. *Bioinformatics*, 20(16):2605–2617.

[222] Luo, Z., Zhao, Y., and Azencott, R. (2014). Impact of mirna sequence on mirna expression and correlation between mirna expression and cell cycle regulation in breast cancer cells. *PloS one*, 9(4):e95205.

[223] Luque-Baena, R., Urda, D., Claros, M. G., Franco, L., and Jerez, J. M. (2014). Robust gene signatures from microarray data using genetic algorithms enriched with biological pathway keywords. *Journal of biomedical informatics*, 49:32–44.

[224] Ma, P. C. and Chan, K. C. (2009). A novel approach for discovering overlapping clusters in gene expression data. *IEEE Transactions on Biomedical Engineering*, 56(7):1803–1809.

[225] Macintyre, G., Bailey, J., Gustafsson, D., Haviv, I., and Kowalczyk, A. (2010). Using gene ontology annotations in exploratory microarray clustering to understand cancer etiology. *Pattern Recognition Letters*, 31(14):2138–2146.

[226] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

[227] Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(1):24–45.

[228] Mahanta, P., Ahmed, H., Bhattacharyya, D., and Kalita, J. K. (2011). Triclustering in gene expression data analysis: a selected survey. In *2011 2nd National Conference on Emerging Trends and Applications in Computer Science*, pages 1–6. IEEE.

[229] Mangangcha, I. R., Malik, M. Z., Küçük, Ö., Ali, S., and Singh, R. B. (2019). Identification of key regulators in prostate cancer from gene expression datasets of patients. *Scientific reports*, 9(1):1–16.

[230] Mankad, S. and Michailidis, G. (2014). Biclustering three-dimensional data arrays with plaid models. *Journal of Computational and Graphical Statistics*, 23(4):943–965.

[231] Martinez-Ledesma, E., Verhaak, R. G., and Treviño, V. (2015). Identification of a multi-cancer gene expression biomarker for cancer clinical outcomes using a network-based algorithm. *Scientific reports*, 5:11966.

[232] Matsuyama, H. and Suzuki, H. I. (2020). Systems and synthetic microrna biology: from biogenesis to disease pathogenesis. *International journal of molecular sciences*, 21(1):132.

[233] Mattes, J., Yang, M., and Foster, P. S. (2007). Regulation of microrna by antagomirs: a new class of pharmacological antagonists for the specific regulation of gene function? *American journal of respiratory cell and molecular biology*, 36(1):8–12.

[234] Maulik, U., Mukhopadhyay, A., Bandyopadhyay, S., Zhang, M. Q., and Zhang, X. (2008). Multiobjective fuzzy biclustering in microarray data: method and a new performance measure. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 1536–1543. IEEE.

[235] Metzker, M. L. (2010). Sequencing technologies—the next generation. *Nature reviews genetics*, 11(1):31–46.

[236] Michlewski, G. and Cáceres, J. F. (2019). Post-transcriptional control of mirna biogenesis. *Rna*, 25(1):1–16.

[237] Mirkin, B. G. and Kramarenko, A. V. (2011). Approximate bicluster and tricluster boxes in the analysis of binary data. In *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, pages 248–256. Springer.

[238] Mirzaie, M., Barani, A., Nematbakkhsh, N., and Beigi, M. (2015a). Overdbc: A new density-based clustering method with the ability of detecting overlapped clusters from gene expression data. *Intelligent Data Analysis*, 19(6):1311–1321.

[239] Mirzaie, M., Barani, A., Nematbakkhsh, N., and Mohammad-Beigi, M. (2015b). Bayesian-overdbc: A bayesian density-based approach for modeling overlapping clusters. *Mathematical Problems in Engineering*, 2015.

[240] Mishra, S. and Vipsita, S. (2017). Triclustering of gene expression microarray data using evolutionary approach. In *2017 14th IEEE India Council International Conference (INDICON)*, pages 1–6. IEEE.

[241] Mistry, M. and Pavlidis, P. (2008). Gene ontology term overlap as a measure of gene functional similarity. *BMC bioinformatics*, 9(1):327.

[242] Mitra, S. and Banka, H. (2006). Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition*, 39(12):2464–2477.

[243] Mitra, S., Das, R., Banka, H., and Mukhopadhyay, S. (2009). Gene interaction–an evolutionary biclustering approach. *Information Fusion*, 10(3):242–249.

[244] Mitra, S. and Ghosh, S. (2012). Feature selection and clustering of gene expression profiles using biological knowledge. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1590–1599.

[245] Mohammed, A., Biegert, G., Adamec, J., and Helikar, T. (2017). Identification of potential tissue-specific cancer biomarkers and development of cancer versus normal genomic classifiers. *Oncotarget*, 8(49):85692.

[246] Möller-Levet, C. S., Klawonn, F., Cho, K.-H., Yin, H., and Wolkenhauer, O. (2005). Clustering of unevenly sampled gene expression time-series data. *Fuzzy sets and Systems*, 152(1):49–66.

[247] Montojo, J., Zuberi, K., Rodriguez, H., Kazi, F., Wright, G., Donaldson, S. L., Morris, Q., and Bader, G. D. (2010). Genemania cytoscape plugin: fast gene function predictions on the desktop. *Bioinformatics*, 26(22):2927–2928.

[248] Morgan, J. N. and Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American statistical association*, 58(302):415–434.

[249] Mubeen, S., Hoyt, C. T., Gemünd, A., Hofmann-Apitius, M., Fröhlich, H., and Domingo-Fernández, D. (2019). The impact of pathway database choice on statistical enrichment analysis and predictive modeling. *Frontiers in genetics*, 10:1203.

[250] Mukhopadhyay, A., Maulik, U., and Bandyopadhyay, S. (2009). A novel coherence measure for discovering scaling biclusters from gene expression data. *Journal of bioinformatics and computational biology*, 7(05):853–868.

[251] Mukhopadhyay, A., Maulik, U., and Bandyopadhyay, S. (2010). On biclustering of gene expression data. *Current Bioinformatics*, 5(3):204–216.

[252] Murali, T. and Kasif, S. (2003). Extracting conserved gene expression motifs from gene expression data. In *Pacific Symposium on Biocomputing*, volume 8, pages 77–88. World Scientific.

[253] Nagele, E., Han, M., DeMarshall, C., Belinka, B., and Nagele, R. (2011). Diagnosis of alzheimer's disease based on disease-specific autoantibody profiles in human sera. *PloS one*, 6(8):e23112.

[254] Nepomuceno, J. A., Troncoso, A., and Aguilar-Ruiz, J. S. (2011). Biclustering of gene expression data by correlation-based scatter search. *BioData mining*, 4(1):3.

[255] Nepomuceno, J. A., Troncoso, A., Nepomuceno-Chamorro, I. A., and Aguilar-Ruiz, J. S. (2015). Integrating biological knowledge based on functional annotations for biclustering of gene expression data. *Computer methods and programs in biomedicine*, 119(3):163–180.

[256] Nepomuceno, J. A., Troncoso, A., Nepomuceno-Chamorro, I. A., and Aguilar-Ruiz, J. S. (2016). Biclustering of gene expression data based on simui semantic similarity measure. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 685–693. Springer.

[257] Nepomuceno, J. A., Troncoso, A., Nepomuceno-Chamorro, I. A., and Aguilar-Ruiz, J. S. (2018). Pairwise gene go-based measures for biclustering of high-dimensional expression data. *BioData mining*, 11(1):4.

[258] Nikkilä, J., Törönen, P., Kaski, S., Venna, J., Castrén, E., and Wong, G. (2002). Analysis and visualization of gene expression data using self-organizing maps. *Neural networks*, 15(8):953–966.

[259] Obernosterer, G., Leuschner, P. J., Alenius, M., and Martinez, J. (2006). Post-transcriptional regulation of microrna expression. *Rna*, 12(7):1161–1167.

[260] O'Brien, J., Hayder, H., Zayed, Y., and Peng, C. (2018). Overview of microrna biogenesis, mechanisms of actions, and circulation. *Frontiers in endocrinology*, 9:402.

[261] Odibat, O., Reddy, C. K., and Giroux, C. N. (2010). Differential biclustering for gene expression analysis. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 275–284. ACM.

[262] Oghabian, A., Kilpinen, S., Hautaniemi, S., and Czeizler, E. (2014). Biclustering methods: biological relevance and application in gene expression analysis. *PloS one*, 9(3):e90801.

[263] Oksenberg, N. and Ahituv, N. (2013). The role of auts2 in neurodevelopment and human evolution. *Trends in Genetics*, 29(10):600–608.

[264] Ovaska, K., Laakso, M., and Hautaniemi, S. (2008). Fast gene ontology based clustering for microarray experiments. *BioData Mining*, 1.

[265] Oyelade, J., Isewon, I., Oladipupo, F., Aromolaran, O., Uwoghiren, E., Ameh, F., Achas, M., and Adebiyi, E. (2016). Clustering algorithms: their application to gene expression data. *Bioinformatics and Biology insights*, 10:BBI–S38316.

[266] Padilha, V. A. and Campello, R. J. (2017). A systematic comparative evaluation of biclustering techniques. *BMC bioinformatics*, 18(1):55.

[267] Pan, Y.-l., Jun, Q., Zhou, L., Zhang, T.-t., and Qiang, L. (2018). Ribosomal protein 16 overexpresses in prostate cancer and promotes tumor progression. *Journal of Shanghai Jiaotong University (Medical Science)*, 38(4):394–399.

[268] Parish, A. J., Nguyen, V., Goodman, A. M., Murugesan, K., Frampton, G. M., and Kurzrock, R. (2018). Gnas, gnaq, and gna11 alterations in patients with diverse cancers. *Cancer*, 124(20):4080–4089.

[269] Pesquita, C. (2017). Semantic similarity in the gene ontology. In *The gene ontology handbook*, pages 161–173. Humana Press, New York, NY.

[270] Pesquita, C., Faria, D., Falcao, A. O., Lord, P., and Couto, F. M. (2009). Semantic similarity in biomedical ontologies. *PLoS comput biol*, 5(7):e1000443.

[271] Pierouli, K., Mitsis, T., Papakonstantinou, E., and Vlachakis, D. (2019). Introductory chapter: Gene profiling in cancer in the era of metagenomics and precision medicine. In *Gene Expression Profiling in Cancer*. IntechOpen.

[272] Pinoli, P., Chicco, D., and Masseroli, M. (2015). Computational algorithms to predict gene ontology annotations. *BMC bioinformatics*, 16(Suppl 6):S4.

[273] Pio, G., Ceci, M., D'Elia, D., Loglisci, C., and Malerba, D. (2013). A novel biclustering algorithm for the discovery of meaningful biological correlations between micrornas and their target genes. *BMC bioinformatics*, 14(S7):S8.

[274] Pio, G., Serafino, F., Malerba, D., and Ceci, M. (2018). Multi-type clustering and classification from heterogeneous networks. *Information Sciences*, 425:107–126.

[275] Pirim, H., Ekşioğlu, B., Perkins, A. D., and Yüceer, Ç. (2012). Clustering of high throughput gene expression data. *Computers & operations research*, 39(12):3046–3061.

[276] Pongor, L., Kormos, M., Hatzis, C., Pusztai, L., Szabó, A., and Győrffy, B. (2015). A genome-wide approach to link genotype to clinical outcome by utilizing next generation sequencing and gene chip data of 6,697 breast cancer patients. *Genome medicine*, 7(1):104.

[277] Pontes, B., Giráldez, R., and Aguilar-Ruiz, J. S. (2015). Biclustering on expression data: A review. *Journal of biomedical informatics*, 57:163–180.

[278] Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., and Zitzler, E. (2006). A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129.

[279] Pritchard, C. C., Cheng, H. H., and Tewari, M. (2012). Microrna profiling: approaches and considerations. *Nature Reviews Genetics*, 13(5):358–369.

[280] Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.-H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J. P., et al. (2001). Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences*, 98(26):15149–15154.

[281] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850.

[282] Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.

[283] Rodriguez, A. and Laio, A. (2014). Clustering by fast search and find of density peaks. *science*, 344(6191):1492–1496.

[284] Rodriguez-Baena, D. S., Perez-Pulido, A. J., and Aguilar-Ruiz, J. S. (2011). A biclustering algorithm for extracting bit-patterns from binary datasets. *Bioinformatics*, 27(19):2738–2745.

[285] Rose, K. (1998). Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE*, 86(11):2210–2239.

[286] Roy, S., Bhattacharyya, D. K., and Kalita, J. K. (2013). Cobi: pattern based co-regulated biclustering of gene expression data. *Pattern Recognition Letters*, 34(14):1669–1678.

[287] Saber, H. B. and ELLOUMI, M. (2015). Dna microarray data analysis: A new survey on biclustering. *International Journal for Computational Biology (IJCB)*, 4(1):21–37.

[288] Sachnev, V., Saraswathi, S., Niaz, R., Kloczkowski, A., and Suresh, S. (2015). Multi-class bcga-elm based classifier that identifies biomarkers associated with hallmarks of cancer. *BMC bioinformatics*, 16(1):166.

[289] Sadlonova, A., Bowe, D. B., Novak, Z., Mukherjee, S., Duncan, V. E., Page, G. P., and Frost, A. R. (2009). Identification of molecular distinctions between normal breast-associated fibroblasts and breast cancer-associated fibroblasts. *Cancer microenvironment*, 2(1):9.

[290] Saeed, A., Sharov, V., White, J., Li, J., Liang, W., Bhagabati, N., Braisted, J., Klapa, M., Currier, T., Thiagarajan, M., et al. (2003). Tm4: a free, open-source system for microarray data management and analysis. *Biotechniques*, 34(2):374.

[291] Samee, N. M. A., Solouma, N. H., and Kadah, Y. M. (2012). Detection of biomarkers for hepatocellular carcinoma using a hybrid univariate gene selection methods. *Theoretical Biology and Medical Modelling*, 9(1):34.

[292] Schlange, T., Matsuda, Y., Lienhard, S., Huber, A., and Hynes, N. E. (2007). Autocrine wnt signaling contributes to breast cancer cell proliferation via the canonical wnt pathway and egfr transactivation. *Breast cancer research*, 9(5):1–15.

[293] Sharan, R. and Shamir, R. (2000). Click: a clustering algorithm with applications to gene expression analysis. In *Proc Int Conf Intell Syst Mol Biol*, volume 8, page 16.

[294] Sheetz, T., Mills, J., Tessari, A., Pawlikowski, M., Braddom, A. E., Posid, T., Zynger, D. L., James, C., Embrione, V., Parbhoo, K., et al. (2020). Ncl inhibition exerts antineoplastic effects against prostate cancer cells by modulating oncogenic microrna. *Cancers*, 12(7):1861.

[295] Sheng, W., Tucker, A., and Liu, X. (2010). A niching genetic k-means algorithm and its applications to gene expression data. *Soft Comput.*, 14(1):9–19.

[296] Shi, Y., Cai, Z., Lin, G., and Schuurmans, D. (2009). Linear coherent bi-cluster discovery via line detection and sample majority voting. In *International Conference on Combinatorial Optimization and Applications*, pages 73–84. Springer.

[297] Shi, Y., Hasan, M., Cai, Z., Lin, G., and Schuurmans, D. (2010). Linear coherent bi-cluster discovery via beam detection and sample set clustering. In *International Conference on Combinatorial Optimization and Applications*, pages 85–103. Springer.

[298] Shi, Y., Liao, X., Zhang, X., Lin, G., and Schuurmans, D. (2012). Sparse learning based linear coherent bi-clustering. In *International Workshop on Algorithms in Bioinformatics*, pages 346–364. Springer.

[299] Shimomura, A., Shiino, S., Kawauchi, J., Takizawa, S., Sakamoto, H., Matsuzaki, J., Ono, M., Takeshita, F., Niida, S., Shimizu, C., et al. (2016). Novel combination of serum microrna for detecting breast cancer in the early stage. *Cancer science*, 107(3):326–334.

[300] Shrifan, N. H., Akbar, M. F., and Isa, N. A. M. (2021). An adaptive outlier removal aided k-means clustering algorithm. *Journal of King Saud University-Computer and Information Sciences*.

[301] Si, W., Shen, J., Zheng, H., and Fan, W. (2019). The role and mechanisms of action of micrornas in cancer drug resistance. *Clinical epigenetics*, 11(1):25.

[302] Sill, M., Kaiser, S., Benner, A., and Kopp-Schneider, A. (2011). Robust biclustering by sparse singular value decomposition incorporating stability selection. *Bioinformatics*, 27(15):2089–2097.

[303] Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A. A., D'Amico, A. V., Richie, J. P., et al. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer cell*, 1(2):203–209.

[304] Sonne-Hansen, K., Norrie, I. C., Emdal, K. B., Benjaminsen, R. V., Frogne, T., Christiansen, I. J., Kirkegaard, T., and Lykkesfeldt, A. E. (2010). Breast cancer cells can switch between estrogen receptor α and erbb signaling and

combined treatment against both signaling pathways postpones development of resistance. *Breast cancer research and treatment*, 121(3):601–613.

[305] Spearman, C. (1904). The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101.

[306] Speer, N., Spieth, C., and Zell, A. (2004). A memetic clustering algorithm for the functional partition of genes based on the gene ontology. In *Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2004, La Jolla, CA, USA, October 7-8, 2004*, pages 252–259.

[307] Srivastava, S., Zhang, L., Jin, R., and Chan, C. (2008). A novel method incorporating gene ontology information for unsupervised clustering and feature selection. *PloS one*, 3(12).

[308] Stope, M. B., Popp, S. L., Knabbe, C., and Buck, M. B. (2010). Estrogen receptor α attenuates transforming growth factor-β signaling in breast cancer cells independent from agonistic and antagonistic ligands. *Breast cancer research and treatment*, 120(2):357–367.

[309] Stratford, J. K., Bentrem, D. J., Anderson, J. M., Fan, C., Volmar, K. A., Marron, J., Routh, E. D., Caskey, L. S., Samuel, J. C., Der, C. J., et al. (2010). A six-gene signature predicts survival of patients with localized pancreatic ductal adenocarcinoma. *PLoS Med*, 7(7):e1000307.

[310] Sugiyama, A. and Kotani, M. (2002). Analysis of gene expression data by using self-organizing maps and k-means clustering. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, volume 2, pages 1342–1345. IEEE.

[311] Sutheeworapong, S., Ota, M., Ohta, H., and Kinoshita, K. (2012). A novel biclustering approach with iterative optimization to analyze gene expression data. *Advances and applications in bioinformatics and chemistry: AABC*, 5:23.

[312] Szeto, C. Y.-Y., Lin, C. H., Choi, S. C., Yip, T. T., Ngan, R. K.-C., Tsao, G. S.-W., and Lung, M. L. (2014). Integrated mrna and microrna transcriptome sequencing characterizes sequence variants and mrna–microrna regulatory network in nasopharyngeal carcinoma model systems. *FEBS open bio*, 4:128–140.

[313] Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., and Golub, T. R. (1999). Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences*, 96(6):2907–2912.

[314] Tanay, A., Sharan, R., and Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(suppl 1):S136–S144.

[315] Tanay, A., Sharan, R., and Shamir, R. (2005). Biclustering algorithms: A survey. *Handbook of computational molecular biology*, 9(1-20):122–124.

[316] Tchagang, A. B., Phan, S., Famili, F., Shearer, H., Fobert, P., Huang, Y., Zou, J., Huang, D., Cutler, A., Liu, Z., et al. (2012). Mining biological information from 3d short time-series gene expression data: the optricluster algorithm. *BMC bioinformatics*, 13(1):54.

[317] Tellaroli, P., Bazzi, M., Donato, M., Brazzale, A. R., and Drăghici, S. (2016). Cross-clustering: a partial clustering algorithm with automatic estimation of the number of clusters. *PloS one*, 11(3):e0152333.

[318] Teng, L. and Chan, L. (2008). Discovering biclusters by iteratively sorting with weighted correlation coefficient in gene expression data. *Journal of Signal Processing Systems*, 50(3):267–280.

[319] Troester, M. A., Herschkowitz, J. I., Oh, D. S., He, X., Hoadley, K. A., Barbier, C. S., and Perou, C. M. (2006). Gene expression patterns associated with p53 status in breast cancer. *BMC cancer*, 6(1):276.

[320] Tsofack, S. P., Meunier, L., Mes-Masson, A.-M., and Lebel, M. (2014). Rps4x, a new prognostic and predictive biomarker of ovarian and breast cancer.

[321] Turner, H., Bailey, T., and Krzanowski, W. (2005). Improved biclustering of microarray data demonstrated through systematic performance tests. *Computational statistics & data analysis*, 48(2):235–254.

[322] Tusher, V. G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences*, 98(9):5116–5121.

[323] Vargo-Gogola, T. and Rosen, J. M. (2007). Modelling breast cancer: one size does not fit all. *Nature Reviews Cancer*, 7(9):659.

[324] Vendramin, L., Campello, R. J., and Hruschka, E. R. (2010). Relative clustering validity criteria: A comparative overview. *Statistical analysis and data mining: the ASA data science journal*, 3(4):209–235.

[325] Verbanck, M., Lê, S., and Pagès, J. (2013a). A new unsupervised gene clustering algorithm based on the integration of biological knowledge into expression data. *BMC bioinformatics*, 14(1):1–11.

[326] Verbanck, M., Lê, S., and Pagès, J. (2013b). A new unsupervised gene clustering algorithm based on the integration of biological knowledge into expression data. *BMC Bioinformatics*, 14:42.

[327] Vicente, C. M., da Silva, D. A., Sartorio, P. V., Silva, T. D., Saad, S. S., Nader, H. B., Forones, N. M., and Toma, L. (2018). Heparan sulfate proteoglycans in human colorectal cancer. *Analytical Cellular Pathology*, 2018.

[328] Visconti, A., Cordero, F., and Pensa, R. G. (2014). Leveraging additional knowledge to support coherent bicluster discovery in gene expression data. *Intelligent Data Analysis*, 18(5):837–855.

[329] Vlachos, I. S., Zagganas, K., Paraskevopoulou, M. D., Georgakilas, G., Karagkouni, D., Vergoulis, T., Dalamagas, T., and Hatzigeorgiou, A. G. (2015). Diana-mirpath v3. 0: deciphering microrna function with experimental support. *Nucleic acids research*, 43(W1):W460–W466.

[330] Wang, G., Yin, L., Zhao, Y., and Mao, K. (2009a). Efficiently mining time-delayed gene expression patterns. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(2):400–411.

[331] Wang, H., Zheng, H., and Azuaje, F. (2007a). Poisson-based self-organizing feature maps and hierarchical clustering for serial analysis of gene expression data. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 4(2):163–175.

[332] Wang, J. Z., Du, Z., Payattakool, R., Yu, P. S., and Chen, C.-F. (2007b). A new method to measure the semantic similarity of go terms. *Bioinformatics*, 23(10):1274–1281.

[333] Wang, S., Qiu, Y., and Bai, B. (2019). The expression, regulation, and biomarker potential of glypican-1 in cancer. *Frontiers in oncology*, 9.

[334] Wang, X.-Q., Tang, Z.-X., Yu, D., Cui, S.-J., Jiang, Y.-H., Zhang, Q., Wang, J., Yang, P.-Y., and Liu, F. (2016a). Epithelial but not stromal ex-

pression of collagen alpha-1 (iii) is a diagnostic and prognostic indicator of colorectal carcinoma. *Oncotarget*, 7(8):8823.

[335] Wang, Y. K., Crampin, E. J., et al. (2013). Biclustering reveals breast cancer tumour subgroups with common clinical features and improves prediction of disease recurrence. *BMC genomics*, 14(1):102.

[336] Wang, Z., Gerstein, M., and Snyder, M. (2009b). Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63.

[337] Wang, Z., Li, G., Robinson, R. W., and Huang, X. (2016b). Unibic: Sequential row-based biclustering algorithm for analysis of gene expression data. *Scientific reports*, 6:23466.

[338] Watson, J. D., Baker, T. A., Bell, S. P., Gann, A., Levine, M., and Losick, R. (2004). The structures of dna and rna. *Molecular biology of the gene Volume chapter*, 6.

[339] Wu, F. (2008). Genetic weighted k-means algorithm for clustering large-scale gene expression data. *BMC Bioinformatics*, 9(S-6).

[340] Wu, G., Zhao, Z., Yan, Y., Zhou, Y., Wei, J., Chen, X., Lin, W., Ou, C., Li, J., Wang, X., et al. (2020). Cps1 expression and its prognostic significance in lung adenocarcinoma. *Annals of translational medicine*, 8(6).

[341] Wu, M.-Y., Dai, D.-Q., Zhang, X.-F., and Zhu, Y. (2013). Cancer subtype discovery and biomarker identification via a new robust network clustering algorithm. *PloS one*, 8(6).

[342] Wu, X., Somlo, G., Yu, Y., Palomares, M. R., Li, A. X., Zhou, W., Chow, A., Yen, Y., Rossi, J. J., Gao, H., et al. (2012). De novo sequencing of circulating mirnas identifies novel markers predicting clinical outcome of locally advanced breast cancer. *Journal of translational medicine*, 10(1):42.

[343] Wu, X., Zurita-Milla, R., Izquierdo Verdiguier, E., and Kraak, M.-J. (2018). Triclustering georeferenced time series for analyzing patterns of intra-annual variability in temperature. *Annals of the American Association of Geographers*, 108(1):71–87.

[344] Xie, X. L. and Beni, G. (1991). A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (8):841–847.

[345] Xu, X., Lu, Y., Tan, K.-L., and Tung, A. K. (2009). Finding time-lagged 3d clusters. In *2009 IEEE 25th International Conference on Data Engineering*, pages 445–456. IEEE.

[346] Xu, X., Lu, Y., Tung, A. K., and Wang, W. (2006). Mining shifting-and-scaling co-regulation patterns on gene expression profiles. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 89–89. IEEE.

[347] Xu, Y., She, Y., Li, Y., Li, H., Jia, Z., Jiang, G., Liang, L., and Duan, L. (2020). Multi-omics analysis at epigenomics and transcriptomics levels reveals prognostic subtypes of lung squamous cell carcinoma. *Biomedicine & Pharmacotherapy*, 125:109859.

[348] Xue, Y., Liao, Z., Li, M., Luo, J., Kuang, Q., Hu, X., and Li, T. (2015). A new approach for mining order-preserving submatrices based on all common subsequences. *Computational and mathematical methods in medicine*, 2015.

[349] Yadav, A. and Singh, G. (2015). Incremental k-means clustering algorithms: A review. *International Journal of Latest Trends in Engineering and Technology*, 5:140.

[350] Yang, H., Nepusz, T., and Paccanaro, A. (2012). Improving go semantic similarity measures by exploring the ontology beneath the terms and modelling uncertainty. *Bioinformatics*, 28(10):1383–1389.

[351] Yang, J., Wang, H., Wang, W., and Yu, P. (2003). Enhanced biclustering on expression data. In *Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium on*, pages 321–327. IEEE.

[352] Yang, J., Wang, H., Wang, W., and Yu, P. S. (2005). An improved biclustering method for analyzing gene expression profiles. *International Journal on Artificial Intelligence Tools*, 14(05):771–789.

[353] Yang, J., Wang, W., Wang, H., and Yu, P. (2002). δ-clusters: Capturing subspace correlation in a large data set. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 517–528. IEEE.

[354] Yang, L., Shen, Y., Yuan, X., Zhang, J., and Wei, J. (2017a). Analysis of breast cancer subtypes by ap-isa biclustering. *BMC bioinformatics*, 18(1):481.

[355] Yang, W.-H., Dai, D.-Q., and Yan, H. (2011). Finding correlated biclusters from gene expression data. *Knowledge and Data Engineering, IEEE Transactions on*, 23(4):568–584.

[356] Yang, Y., Huang, N., Hao, L., and Kong, W. (2017b). A clustering-based approach for efficient identification of microrna combinatorial biomarkers. *BMC genomics*, 18(2):1–14.

[357] Yoon, S., Kim, J., Kim, S.-K., Baik, B., Chi, S.-M., Kim, S.-Y., and Nam, D. (2019). Gscluster: network-weighted gene-set clustering analysis. *BMC genomics*, 20(1):1–14.

[358] Yu, G., Li, F., Qin, Y., Bo, X., Wu, Y., and Wang, S. (2010). Gosemsim: an r package for measuring semantic similarity among go terms and gene products. *Bioinformatics*, 26(7):976–978.

[359] Yu, G., Yin, C., Jiang, L., Xu, D., Zheng, Z., Wang, Z., Wang, C., Zhou, H., Jiang, X., Liu, Q., et al. (2018a). Amyloid precursor protein has clinical and prognostic significance in aml1-eto-positive acute myeloid leukemia. *Oncology letters*, 15(1):917–925.

[360] Yu, H., Ding, J., Zhu, H., Jing, Y., Zhou, H., Tian, H., Tang, K., Wang, G., and Wang, X. (2020a). Loxl1 confers antiapoptosis and promotes gliomagenesis through stabilizing bag2. *Cell Death & Differentiation*, pages 1–16.

[361] Yu, J., Zhou, Z., Wei, Z., Wu, J., OuYang, J., Huang, W., He, Y., and Zhang, C. (2020b). Fyn promotes gastric cancer metastasis by activating stat3-mediated epithelial-mesenchymal transition. *Translational Oncology*, 13(11):100841.

[362] Yu, Y., Liu, D., Liu, Z., Li, S., Ge, Y., Sun, W., and Liu, B. (2018b). The inhibitory effects of col1a2 on colorectal cancer cell proliferation, migration, and invasion. *Journal of Cancer*, 9(16):2953.

[363] Zhang, M., Wang, W., and Liu, J. (2008). Mining approximate order preserving clusters in the presence of noise. In *2008 IEEE 24th International Conference on Data Engineering*, pages 160–168. IEEE.

[364] Zhao, L. and Zaki, M. J. (2005). Tricluster: an effective algorithm for mining coherent clusters in 3d microarray data. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 694–705. ACM.

[365] Zhao, X., Zhong, S., Zuo, X., Lin, M., Qin, J., Luan, Y., Zhang, N., Liang, Y., and Rao, S. (2014). Pathway-based analysis of the hidden genetic

heterogeneities in cancers. *Genomics, proteomics & bioinformatics*, 12(1):31–38.

[366] Zhao, Y., Li, Y., Lou, G., Zhao, L., Xu, Z., Zhang, Y., and He, F. (2012). Mir-137 targets estrogen-related receptor alpha and impairs the proliferative and migratory capacity of breast cancer cells. *PloS one*, 7(6):e39102.

[367] Zhao, Y., Si, L., Zhang, W., Huang, W., and Wang, R. (2019). Elane is highly expressed in leukemia patients and predicts poor survival. *Int J Clin Exp Med*, 12(4):3153–3160.

[368] Zhao, Y.-H., Wang, G.-R., Yin, Y., and Xu, G.-Y. (2007). A novel approach to revealing positive and negative co-regulated genes. *Journal of Computer Science and Technology*, 22(2):261–272.

[369] Zhou, W. and Dickerson, J. A. (2014). A novel class dependent feature selection method for cancer biomarker discovery. *Computers in biology and medicine*, 47:66–75.

[370] Zhou, X., Sun, H., Wang, D.-P., Zhang, Y., and Zhou, Y. (2010). Analysis of gene expression data based on density and biological knowledge. In *2010 Fifth International Conference on Frontier of Computer Science and Technology*, pages 448–453. IEEE.

Publications based on the Thesis Works

Journals

1. **Mandal, K.**, Sarmah, R., and Bhattacharyya, D. K., "Biomarker Identification for Cancer Disease Using Biclustering Approach: An Empirical Study", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 2, pp. 490–509, 2018, DOI: 10.1109/TCBB.2018.2820695.

2. **Mandal, K.**, Sarmah, R., and Bhattacharyya, D. K., "POPBic: Pathway-based Order Preserving Biclustering Algorithm Towards the Analysis of Gene Expression Data", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 6, pp. 2659–2670, 2020, DOI: 10.1109/TCBB.2020.2980816.

3. **Mandal, K.**, Sarmah, R., Bhattacharyya, D. K., Kalita J. K., and Borah B., "Rank-preserving biclustering algorithm: a case study on miRNA breast cancer", *Medical & Biological Engineering & Computing*, vol. 59, no. 4, pp. 989–1004, 2021, DOI:10.1007/s11517-020-02271-0.

4. **Mandal, K.**, Sarmah, R., and Bhattacharyya, D. K., "POPTric: Pathway-based Order Preserving Triclustering for gene sample time data analysis", *Expert Systems with Applications*, vol. 192, pp. 116336, 2022, DOI: 10.1016/j.eswa.2021.116336.

5. **Mandal, K.** and Sarmah, R., "SGAClust: Semi-supervised Graph Attraction Clustering of gene expression data", *Network Modeling Analysis in Health Informatics and Bioinformatics*. (Under second review)

Conference/Workshop

6. **Mandal, K.**, Sarmah, R., and Borah B., "A Fuzzy Graph Based Cluster

Affinity Search Technique for clustering of gene expression data", *IEEE International Conference on Systems in Medicine and Biology (ICSMB)*, pp. 78–82, January 4-7, 2016, Kharagpur, India.

Book Chapter

7. **Mandal, K.** and Sarmah, R., "A Density-Based Clustering for Gene Expression Data Using Gene Ontology", *Proceedings of the International Conference on Computing and Communication Systems (I3CS)*, ISBN: 978-981-10-6890-4, In: Mandal, J. K., Saha, G., Kandar, D., and Maji, A. K., Editors. Springer Singapore, pp. 757–765, 2018.

Appendix

The POPBic tool has been developed in a MATLAB environment to work on microarray gene expression data for extracting biclusters. The tool also helps to identify biomarkers from identified biclusters. To run the tool MATLAB software is required. The tool can be accessed at `http://agnigarh.tezu.ernet.in/~rosy8/shared.html`. The stepwise instructions to use this tool are given below.

(I) Extract the zipped folder named "POPBic_Tool.rar" in to your computer. The POPBic_Tool folder contains the following files and folder.

(a) Sample_data (folder)

(b) POPBic_tool.fig

(c) POPBic_tool.p

(II) Start MATLAB and change the working directory of the MATLAB to the specified location where you have extracted the files.

(III) In the MATLAB console type POPBic_tool and hit enter. After that, the following GUI in Figure A1 will appear.

(IV) The main window has three panels, viz. Input, Output, visualization, and Biomarker identification as shown in the Figure A1.

(V) Load the gene expression dataset in the input panel by clicking the Browse button which will display a dialog box where the user can select the file. The dataset should be in the '.xlsx' format, where rows represent genes and columns represent experimental conditions. The sample file "gene_expression.xlsx" is given in the sample folder.

(VI) Load the KEGG pathway information in the input panel by clicking the Browse button which will display a dialog box where the user can select the file. The file should be in the '.txt' format. The sample file "pathway.txt" is given in the sample folder. The pathway information can easily be downloaded from the

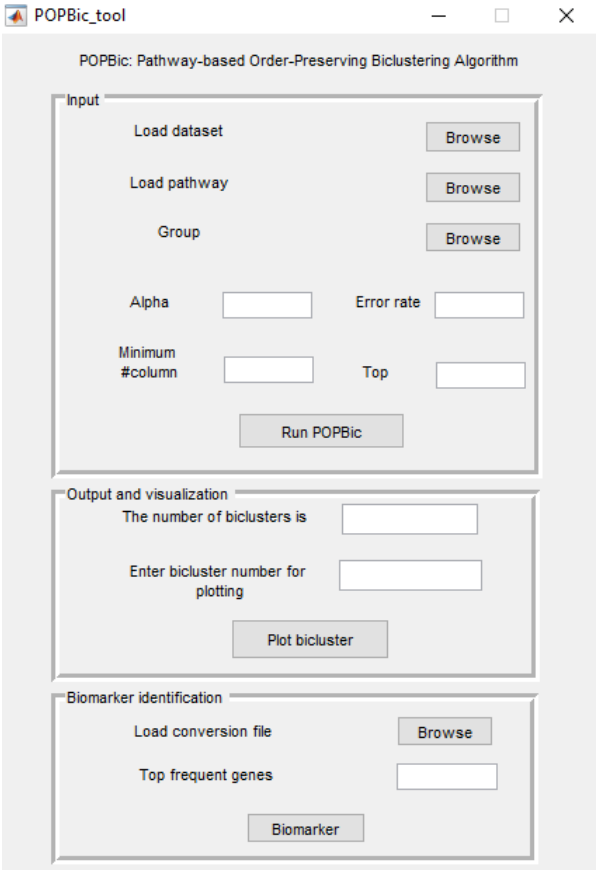


Figure A1: The screen-shot of POPBic tool.

Functional Annotation Table tab of the DAVID web browser by giving the list of genes in the upload section.

(VII) Load the group information in the input panel by clicking the Browse button which will display a dialog box where the user can select the file. The file should be in the '.xlsx' format. The sample file "group.xlsx" is given in the sample folder. The file is a column vector that is created depending on the experimental conditions in gene expression data. Here, we describe how to create the file by giving an example. In the 'gene_expression.xlsx' file there are 102 samples and two subtypes (50 normal samples and 52 tumor samples). Each normal sample is mapped to 1 and each tumor sample is mapped to 2. The sequence of experimental conditions given in the gene expression dataset is strictly followed in the 'group.xlsx'.

(VIII) Use following values of POPBic algorithm for sample data.
Alpha = 0.05, Error rate = 0.5, Minimum number of conditions = 5, and Top = 0.1

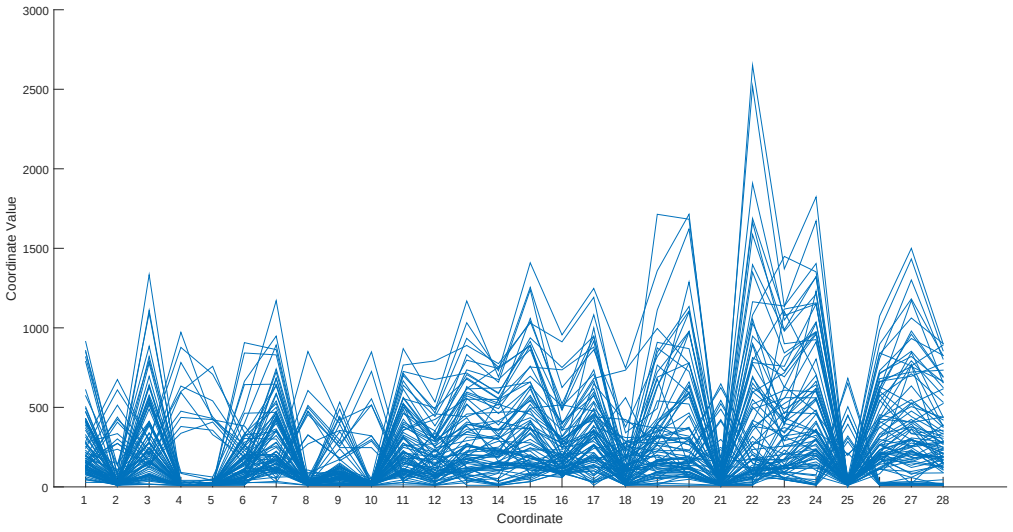


Figure A2: Co-expressed genes of a bicluster.

(IX) Press the button 'Run POPBic' to get the result. The execution may take time. So be patient while executing. Two files will be saved in the same folder one 'Intermediate_result.xlsx' and another 'POPBic_output.txt'. The first, second, and third sheet of 'intermediate_result.xlsx' consist of pathway identifier, the pathways corresponding to each of the genes, and p-value for each of the genes. In the case of p-values, we put '100' for some of the genes where we have not found the pathways in the 'Sheet2'.

(X) In the second panel i.e., output and visualization the number of biclusters will be shown after the completion of execution. Enter the number of biclusters (from 1 to number which is displayed in the previous text box) which you want to plot and then press 'Plot bicluster'. A separate window will pop up for plotting which can be saved for later use. One such example of a bicluster plotting is shown in Figure A2.

(XI) The third panel is for biomarker identification. To get the biomarkers from the extracted biclusters, browse the conversion file. The conversion file can be downloaded from the Gene ID Conversion Tool of the DAVID web browser by giving the list of genes in the upload section. In the sample example, we have converted our Affymetrix genes into official IDs.

(XII) Use 2 as a default value for the top frequent gene to get the desired biomarkers. The output file will be saved as 'biomarker.txt' in the same folder. The biomarkers can be verified from the literature.