

**Flexible model-based joint probabilistic  
clustering of binary and continuous inputs and  
its application to genetic regulation and cancer**

by

Fatin Nurzahirah Binti Zainul Abidin

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

**THE UNIVERSITY OF LEEDS**

in the

Faculty of Biological Sciences

School of Molecular and Cellular Biology

August 2017

## Intellectual Property and Publication Statements

The candidate confirms that the work submitted is his/her own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below

1st Authored, used in this thesis,

Chapter 2, 3 and 4: Fatin N. Zainul Abidin and David R. Westhead. "**Flexible model-based clustering of mixed binary and continuous data: application to genetic regulation and cancer**". *Nucleic Acids Res* (2017) 45 (7):e53.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

The right of Fatin N. Zainul Abidin to be identified as Author of this work has been asserted by her in accordance with the copyright, Designs and Patents Act 1988.

## **Acknowledgements**

First and foremost, thanks to God for giving me the chance and strength to complete this thesis in time although I faced some personal difficulties along the way to complete this thesis, but I am glad I still manage to complete it. Then, thanks to my lovely family in Malaysia for their understanding and allowing me to further my study here and being away from home with a great distance for three and a half years.

Next, I would like to express the deepest appreciation to my supervisor, Professor David Westhead, head of the School of Molecular and Cellular Biology who introduced me to the Bioinformatics field and for his patient, guidance and support whenever I ran into trouble or had a question about my research or writing. I was lucky to have learned a lot from Dave and his presence in each step of my PhD progressions and encouraging and kind to me whenever I get confused and frustrated was really a blessing for me.

In addition, thank you to my fellow lab mates especially Vijay, Nisar, Francis, Chulin and Matt for providing me with unfailing support and continuous encouragement throughout my years of study. I would also like to acknowledge Dr. Joan Boyes as my secondary supervisor and also as the second reader of this thesis, and I am gratefully indebted for Dave's and Joan's very valuable comments in this thesis.

Finally, I would like to thank the Majlis Amanah Rakyat (MARA), or the Council of Trust for the People, an agency under the Ministry of Rural and Regional Development Malaysia for their trust in me and willing to sponsor a huge amount of fund so that I am able to study here because of the lack of Bioinformatics concentration back home.

## Abstract

Clustering is used widely in 'omics' studies and is often tackled with standard methods such as hierarchical clustering or k-means which are limited to a single data type. In addition, these methods are further limited by having to select a cut-off point at specific level of dendrogram- a tree diagram or needing a pre-defined number of clusters respectively. The increasing need for integration of multiple data sets leads to a requirement for clustering methods applicable to mixed data types, where the straightforward application of standard methods is not necessarily the best approach. A particularly common problem involves clustering entities characterized by a mixture of binary data, for example, presence or absence of mutations, binding, motifs, and/or epigenetic marks and continuous data, for example, gene expression, protein abundance and/or metabolite levels.

In this work, we presented a generic method based on a probabilistic model for clustering this mixture of data types, and illustrate its application to genetic regulation and the clustering of cancer samples. It uses penalized maximum likelihood (ML) estimation of mixture model parameters using information criteria (model selection objective function) and meta-heuristic searches for optimum clusters. Compatibility of several information criteria with our model-based joint clustering was tested, including the well-known Akaike Information Criterion (AIC) and its empirically determined derivatives ( $AIC_{\lambda}$ ), Bayesian Information Criterion (BIC) and its derivative (CAIC), and Hannan-Quinn Criterion (HQC). We have experimentally shown with simulated data that AIC and  $AIC_{\lambda}$  ( $\lambda=2.5$ ) worked well with our method.

We show that the resulting clusters lead to useful hypotheses: in the case of genetic regulation these concern regulation of groups of genes by specific sets of transcription factors and in the case of cancer samples combinations of gene mutations are related to patterns of gene expression. The clusters have potential mechanistic significance and in the latter case are significantly linked to survival.

## Table of Contents

<b>Acknowledgements</b> .....	<b>ii</b>
<b>Abstract</b> .....	<b>iii</b>
<b>List of Figures</b> .....	<b>vii</b>
<b>List of Tables</b> .....	<b>ix</b>
<b>List of Abbreviations</b> .....	<b>x</b>
<b>Chapter 1. Introduction</b> .....	<b>1</b>
1.1. Basic molecular biology .....	1
1.1.1. Central dogma.....	2
1.2. Chromatin.....	3
1.2.1. Transcription and transcription control .....	4
1.2.2. Sequence-specific TFs and combinatorial regulation.....	5
1.2.3. Epigenetics and its relationship to gene regulation .....	5
1.2.4. Chromatin modifications and DNA methylation.....	6
1.2.5. Open questions in eukaryotic gene regulation .....	6
1.3. Experimental techniques for studying regulation .....	6
1.3.1. Microarrays and RNA-seq.....	7
1.3.2. ChIP-chip and ChIP-seq .....	10
1.4. Cancer.....	11
1.4.1. Basics of cancer hallmarks and mechanisms.....	12
1.4.2. Different types of genetic aberrations .....	13
1.5. Genomic projects – TCGA, etc. ....	15
1.6. Data analytical techniques .....	16
1.6.1. Clustering .....	16
1.6.2. Dimension reduction.....	20
1.6.3. Supervised analysis (machine learning).....	20
1.6.4. Gene set analysis, GO, GSEA.....	21
1.7. Aims and objectives of the thesis.....	22

<b>Chapter 2. Developing an algorithm to jointly cluster binary and continuous inputs.....</b>	<b>23</b>
2.1. Introduction.....	23
2.1.1. Basic probability concepts .....	24
2.1.2. Likelihood and maximum-likelihood .....	27
2.1.3. Information criterion in model selection.....	28
2.1.4. Maximum-likelihood optimization .....	30
2.1.5. A review on mixture model-based clustering methods of mixed data types.....	32
2.2. Methodology .....	34
2.2.1. The model .....	34
2.2.2. Estimating model parameters .....	35
2.2.3. Testing the algorithm on simulated data .....	45
2.2.4. Summary of test data sets .....	46
2.3. Results .....	47
2.3.1. Optimization of runtime parameters .....	47
2.3.2. Running simulation on simulated data .....	50
2.4. Discussion .....	57
2.5. Conclusions.....	57
<b>Chapter 3. Modelling <i>S. cerevisiae</i> cell cycle transcriptional regulation using a model-based joint clustering algorithm .....</b>	<b>59</b>
3.1. Introduction.....	59
3.1.1. Why yeast?.....	60
3.1.2. Yeast cell cycle control .....	60
3.2. Methodology .....	63
3.2.1. Pre-processing of genes expression data .....	63
3.2.2. Pre-processing of TFs binding data .....	64
3.2.3. Comparison with an existing method .....	65
3.2.4. Functional analysis of clusters .....	69
3.2.5. Time-lagged correlation calculation for TF-gene interactions.....	69
3.3. Results .....	71

3.3.1.	Overall clusters statistics .....	71
3.3.2.	Marginal densities of modules found using SA.....	75
3.3.3.	Statistical analysis of the clusters .....	78
3.3.4.	Yeast cell cycle TRN and regulators interactions .....	85
3.3.5.	Biological analysis of the clusters .....	87
3.3.6.	TF-TF interactions inferred from the clustering output.....	90
3.3.7.	Time-lagged correlation analysis .....	91
3.4.	Discussion .....	96
3.5.	Conclusion.....	96
<b>Chapter 4.</b>	<b>Application of model-based joint clustering to cancer data.....</b>	<b>97</b>
4.1.	Introduction.....	97
4.1.1.	How AML develops.....	100
4.1.2.	Molecular aberrations of AML .....	101
4.2.	Methodology.....	102
4.2.1.	Selecting mutated genes and variably expressed genes .....	102
4.2.2.	Clusters analysis .....	105
4.3.	Results and discussion .....	107
4.3.1.	Comparison between our clusters and FAB classifications .....	117
4.4.	Conclusion.....	119
<b>Chapter 5.</b>	<b>Discussion and future work .....</b>	<b>120</b>
5.1.	Method development .....	120
5.2.	Transcriptional regulatory networks reconstruction .....	121
5.3.	Identification of cancer subtypes.....	122
5.4.	Future work.....	123
<b>References.....</b>		<b>124</b>
<b>Appendix A.....</b>		<b>135</b>
<b>Appendix B.....</b>		<b>143</b>

## List of Figures

<b>Figure 1.1</b>	A section of a chromosome (a gene) containing exon in between of introns.....	2
<b>Figure 1.2</b>	Prokaryotic vs eukaryotic gene expression central dogmas of molecular biology. ....	3
<b>Figure 1.3</b>	Modulation of transcription of a eukaryotic gene in an active state. ....	4
<b>Figure 1.4</b>	A genomic locus analysed by corresponding chromatin profiling experiments. ....	7
<b>Figure 1.5</b>	Workflow of RNA-seq from library preparation to RNA profiles quantification.....	9
<b>Figure 1.6</b>	Representation of ChIP-seq of a DNA-binding protein on DNA. ....	11
<b>Figure 1.7</b>	How DNA methylation silences a gene.....	14
<b>Figure 1.8</b>	Clustering methods applied for 40 genes which have been measured under two different experimental conditions. ....	18
<b>Figure 1.9</b>	Top-down (agglomerative) and bottom-up (divisive) strategies in hierarchical clustering of six data points in this example.....	19
<b>Figure 2.1</b>	Probability distribution for random variables $X = xk$ from tossing a fair coin twice. ....	25
<b>Figure 2.2</b>	Probability distribution for probability density function of random variables $X = xk$ from a list of continuous values, $k$ . ....	25
<b>Figure 2.3</b>	Representation of finite mixture distributions for multiple components..	27
<b>Figure 2.4</b>	A graphical representation of the simulated annealing process.....	31
<b>Figure 2.5</b>	Refinement of model parameters using EM.....	40
<b>Figure 2.6</b>	A snapshot of the FlexiCoClustering GUI upon running the application. ....	42
<b>Figure 2.7</b>	Comparison of scores between different starting points. ....	48
<b>Figure 2.8</b>	The real-time simulated temperature and acceptance ratio from clustering using AIC2.0. ....	49
<b>Figure 2.9</b>	The real time simulated score and number of clusters from clustering using AIC2.0. ....	49
<b>Figure 2.10</b>	Difference in scores result from our algorithm using a tightly clustered data and relatively little 'noise' data set simulated from the probability distribution assumed in our method.....	50
<b>Figure 2.11</b>	Difference in number of clusters result from our algorithm using a tightly clustered data and relatively little 'noise' data set simulated from the probability distribution assumed in our method.....	51



<b>Figure 2.12</b>	Difference in scores result from our algorithm using a noisier data and less tight clusters data set simulated from the probability distribution assumed in our method. ....	52
<b>Figure 2.13</b>	Difference in number of clusters result from our algorithm using a noisier data and less tight clusters data set simulated from the probability distribution assumed in our method.....	53
<b>Figure 2.14</b>	Standard AIC ( $\lambda = 2.0$ ) clusters membership when compared with AIC ( $\lambda = 2.5$ ) clusters member. ....	54
<b>Figure 2.15</b>	Expectation maximization result. ....	55
<b>Figure 2.16</b>	CPU times taken to run the simulation. ....	56
<b>Figure 3.1</b>	The events during the eukaryotic yeast cell cycle.....	61
<b>Figure 3.2</b>	A representation of co-expression between a TF gene and an average expression pattern of genes in a cluster. ....	70
<b>Figure 3.3</b>	Two examples of clusters showing both expression patterns and regulatory binding patterns. ....	72
<b>Figure 3.4</b>	Expectation maximization results. ....	76
<b>Figure 3.5</b>	Differences in Bernoulli's parameter estimates.....	77
<b>Figure 3.6</b>	Transcriptional regulatory networks and regulatory interaction of our 'clear' clusters. ....	86
<b>Figure 3.7</b>	Cluster 27, 42, and 48 expression and binding patterns.....	88
<b>Figure 3.8</b>	Combinatorial regulatory interactions found in AIC clear clusters. ....	91
<b>Figure 3.9</b>	Examples of time-lag correlation of average target genes expression and its corresponding TF(s) gene expression in cluster 1 and 6.....	92
<b>Figure 3.10</b>	Time-lagged correlation starting from 0 time-point lag until 6 time-points lag for TFs co-operation. ....	93
<b>Figure 3.11</b>	Number of TF-Gene cluster interactions supported by time-lag correlation. ....	94
<b>Figure 3.12</b>	Distributions of number of interactions with positive time-lag correlation ( $r > 0.5$ ) for all 14 TFs.....	95
<b>Figure 4.1</b>	Basic normal blood cell development in bone marrow and abnormal blood cell production which leads to AML. ....	100
<b>Figure 4.2</b>	Gene expression patterns for AML patients.....	103
<b>Figure 4.3</b>	An example of the Kaplan-Meier survival plot.....	106
<b>Figure 4.4</b>	18 clusters found from using AIC ( $\lambda = 2.5$ ).....	108
<b>Figure 4.5</b>	Comparison between our clustering method on AML patients and from using FAB classifications.....	118

## List of Tables

<b>Table 2.1</b>	Penalty terms used in different information criteria. ....	28
<b>Table 2.2</b>	A list of runtime parameters that were simulated in optimizing the model. .....	43
<b>Table 2.3</b>	Eight sets of simulated data with increasing number of genes and genes per module. ....	46
<b>Table 3.1</b>	The filtered TFs and their functional information. ....	66
<b>Table 3.2</b>	Statistics of clusters found by joint clustering of regulation and expression with different objective functions. ....	73
<b>Table 3.3</b>	Example of small sizes clusters with significantly enriched GO terms. ....	79
<b>Table 3.4</b>	Example of clusters which are related in expression and regulation and with significantly enriched GO terms. ....	82
<b>Table 3.5</b>	A summary statistics for the regulatory network of transcription factors and other regulators. ....	85
<b>Table 4.1</b>	Input data points and variables for AML dataset. ....	104
<b>Table 4.2</b>	Genes that are significantly differentially expressed between clusters determined using the one-versus-all phenotype test in the Gene Pattern tool .....	109
<b>Table 4.3</b>	Poor prognosis clusters enriched with at least one statistically significant biological process GO term (P-value < 0.05). ....	114
<b>Table 4.4</b>	Intermediate (C13 and C16) and good prognosis clusters enriched with at least one statistically significant biological process GO term (P- value<0.05). ....	116

## List of Abbreviations

### General abbreviations

AIC	Akaike Information Criterion
AML	Acute Myeloid Leukemia
BIC	Bayesian Information Criterion
BioGRID	Biological General Repository for Interaction Database
CAIC	Corrected Akaike Information Criterion
ChIP-chip	Chromatin Immuno Precipitation followed by Chip
ChIP-seq	Chromatin Immuno Precipitation followed by Sequencing
CNA	Copy Number Aberration
CRE	Cis Regulatory Element
DAVID	The Database for Annotation, Visualization and Integrated Discovery
EM	Expectation Maximization
ENCODE	ENCyclopedia Of DNA Elements
FAB	French-American_British
FDR	False Discovery Rate
FPKM	Fragments Per Kilobase of exon per Million reads
GO	Gene Ontology
GOSemSim	GO Semantic Similarity
GSEA	Gene Set Enrichment Analysis
GUI	Graphical User Interphase
HQC	Hannan-Quinn information Criterion
IDE	Interactive Development Environment
KEGG	Kyoto Encyclopaedia of Genes and Genomes
LeTICE	Learning Transcriptional networks from the Integration of ChIP-chip and Expression data
ML	Maximum Likelihood
mRNA	Messenger RiboNucleic Acid
NCI	National Cancer Institute

NGS	Next Generation Sequencing
NHGRI	National Human Genome Research Institute
NOS	Non Otherwise Specified
RNA-seq	RiboNucleic Acid followed by Sequencing
RPKM	Reads Per Kilobase of exon per Million reads
SA	Simulated Annealing
SGD	Saccharomyces Genome Database
TCGA	The Cancer Genome Atlas
TFs	Transcription Factors
TRN	Transcriptional Regulatory Network
TSS	Transcription Start Site
WHO	World Health Organization

### **Gene Abbreviations**

ACE2	Activator of CUP1 Expression 2
AFT1	Activator of Ferrous Transport 1
ASH1	Asymmetric Synthesis of HO
ASXL1	Additional Sex Combs Like 1
CBFB	Core-Binding Factor Beta Subunit
CBL	Cbl Proto-Oncogene
CDC28	Cell Division Cycle
CDK	Cyclin Dependent Kinase
CEBPA	CCAAT/Enhancer Binding Protein Alpha
CLB	Cyclin B
CLN	Cyclin
DAL80	Degradation of Allantoin 80
DEK	DEK Proto-Oncogene
DNMT3A	DNA Methyltransferase 3 Alpha
ELL	Elongation Factor For RNA Polymerase II

EZH2	Enhancer of zeste homolog 2
FKH1	ForK head Homolog 1
FKH2	ForK head Homolog 2
FLT3	Fms Related Tyrosine Kinase 3
GATA2	GATA Binding Protein 2
GIN4	Growth Inhibitory
GZF3	Gata Zinc Finger protein 3
HOX	Homeotic genes
IDH1	Isocitrate Dehydrogenase (NADP(+)) 1
IDH2	Isocitrate Dehydrogenase (NADP(+)) 2
JAK1	Janus Kinase 1
JAK3	Janus Kinase 3
KMT2A	Lysine Methyltransferase 2A
MBF	MCB-binding Factor
MBP1	MluI-box Binding Protein 1
MCM1	MiniChromosome Maintenance 1
MET31	METHionine requiring 31
MYC	V-Myc Avian Myelocytomatosis Viral Oncogene Homolog
MYH11	Myosin Heavy Chain 11
NDD1	Nuclear Division Defective 1
NPM1	Nucleophosmin 1
NSD1	Nuclear Receptor Binding SET Domain Protein 1
NUP214	Nucleoporin 214
PDR1	Pleiotropic Drug Resistance 1
PHF6	PHD Finger Protein 6
PML	Promyelocytic Leukemia
PTPN11	Protein Tyrosine Phosphatase 11
PU1	Hematopoietic Transcription Factor PU.1
RARA	Retinoic Acid Receptor Alpha

RUNX1	Runt-related transcription factor 1
RUNX1T1	RUNX1 Translocation Partner 1
SBF	SCP-Binding Factor
SIC1	Substrate/Subunit Inhibitor of Cyclin-dependent protein kinase 1
SLITRK	SLIT and NTRK-like protein 5
STAG2	Stromal Antigen 2
STB1	Sin Three Binding protein
SWE1	Saccharomyces WEe1
SWI4	SWItching deficient 4
SWI5	SWItching deficient 5
SWI6	SWItching deficient 6
TET2	Tet Methylcytosine Dioxygenase 2
TP53	Tumour Protein 53
WT1	Wilms Tumor 1
YHP1	Yeast Homeo-Protein 1

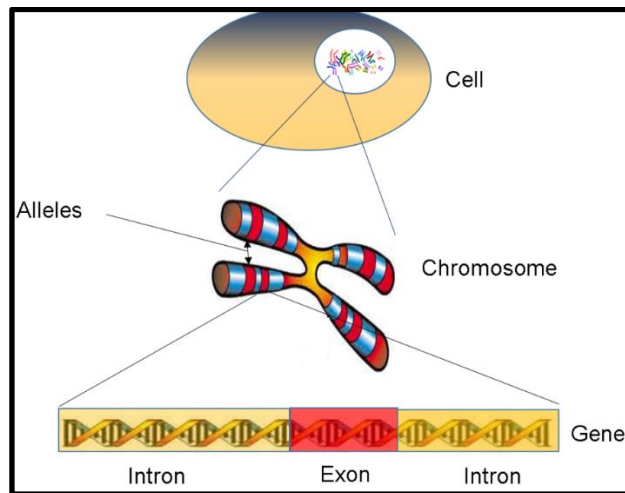
## Chapter 1. Introduction

The enormous amount of biological data produced on a genomic scale requires analytical tools to understand the bigger picture of biological function, be it at the level of the cell or the level of the organism. This is also known as systems biology. In contrast with reductionist biology, which separates biological systems into their components in elucidating the biochemical basis of living processes, systems biology is a top down approach which deals with this enormous data in explaining complex cellular processes and larger organismal system functions [1]. Bioinformatics is a branch of biological study and also a computational study, where applications such as statistical tools are being developed to analyse and make sense of large and diverse genomic datasets (i.e. genes, proteins, and epigenetic states) produced from microarrays and high-throughput technologies.

### 1.1. Basic molecular biology

A cell is a basic unit of life and recognized as a cell because it is surrounded by a membrane, also referred to as the plasma membrane. A cell's interior environment is in liquid form and called cytoplasm. There is the hereditary unit of life known as deoxyribonucleic acid (DNA) within a membrane-bound nucleus in eukaryotic multicellular organisms, or within the cytoplasm itself in a prokaryotic unicellular organisms. DNA is a blueprint that encompasses all the information required to build and maintain an organism's biological system. It is made up of long chains of bases: Adenine (A), Guanine (G), Thymine (T) and Cytosine (C). Apart from the nucleus, the cytoplasm also accommodates organelles, proteins, carbohydrates, lipids which perform the cell's functions.

A functional unit of heredity of life is known as a gene. A gene is a section/region of the DNA which provides instruction to make a protein and other transcribed ribonucleic acids (RNAs) that do not undergo translation into proteins (e.g. rRNA, tRNA, miRNA, etc.). In humans, genes are arranged on chromosomes that are found in the cell's nucleus. A human cell has 23 pairs of chromosomes and are made up of coding DNA (i.e. exons) that are interspaced with non-coding DNA (i.e. introns) (see Figure 1.1). Individual genes are separated by intergenic regions. Chromosomes are contained in every cell in human body except blood cells, and genes are carried on all chromosomes. Moreover all chromosomes exist in pairs except the XY sex chromosomes in males. The two copies of each gene is called an allele.



**Figure 1.1** A section of a chromosome (a gene) containing exon in between of introns.

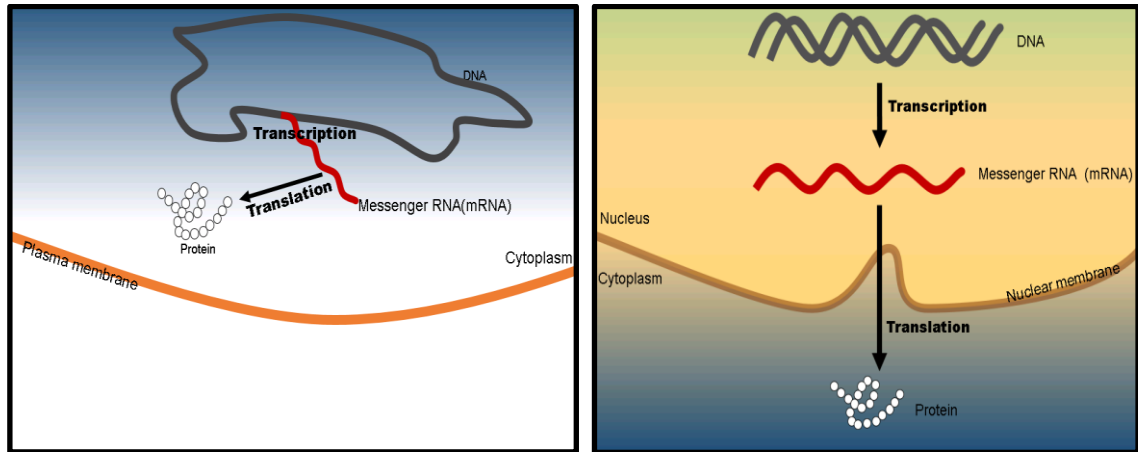
Molecular biology explores cells, their parts and biological and chemical processes between biomolecules which includes DNA, RNAs and proteins. Regulation of the biosynthesis of proteins at the molecular level, specifically at the transcriptional level is important to dictate the function of a cell.

### 1.1.1. Central dogma

A central dogma of biology states that, in each and every cell which makes up an organism, the flow of biological information is from DNA to messenger RNA (mRNA) and subsequently the formation of the protein [2]. The phenotypes observed in an organism result from the working forces of this central dogma. This process is quite straight forward in lower prokaryotic organisms such as bacteria where their gene expression is mainly controlled at the level of transcription [3]. However, for organisms with higher levels of organization such as mammals, it not as straightforward. Eukaryotic gene expression is controlled at the levels of transcription, epigenetics - any process other than DNA sequence that could alter gene activity which leads to heritable modifications - post-transcription, translation and post-translation [3]. Eukaryotic cells function differently, and a multitude of combinations of different gene regulations, metabolic reactions, cell-cell signaling/interactions and responses to stimuli at the cellular, tissue, organ and organ system levels lead to the different phenotypes observed.

The first portion of the central dogma is known as the transcriptome. This covers the transcription of DNA to mRNA inside the nucleus of a cell in eukaryote or in the cytoplasm itself in prokaryote. The second portion is known as the proteome, which is the translation from mRNA to protein in the cytoplasm of both prokaryotic and eukaryotic cells (see Figure 1.2 below).





**Figure 1.2** Prokaryotic vs eukaryotic gene expression central dogmas of molecular biology.

Prokaryotic (**left**) vs eukaryotic (**right**) gene expression central dogmas of molecular biology. The hereditary information flows from DNA to messenger RNA, known as transcription and then forming a protein through the translation process. Prokaryotic transcription and translation occur simultaneously in the cytoplasm whilst eukaryotic gene expression and translation take place in the nucleus and cytoplasm respectively.

It is possible but often impractical to study both the transcriptome and proteome to uncover the working force behind each phenotypic observation due to the complexity of genome. However, some attempts on capturing the cross talk between omics levels have found that RNA levels can only explain a small portion of the protein abundance observed [4, 5], and metabolic/clinical traits were correlated better to RNA levels than the protein levels [4]. To this end, studies related to different omics layers have been done separately due to the difficulties mentioned above. This thesis will focus in understanding the transcriptome of organisms (i.e. yeast and human) and its related regulations. A method developed to assist the knowledge discovery from high-throughput data will be described and discussed later in the chapters.

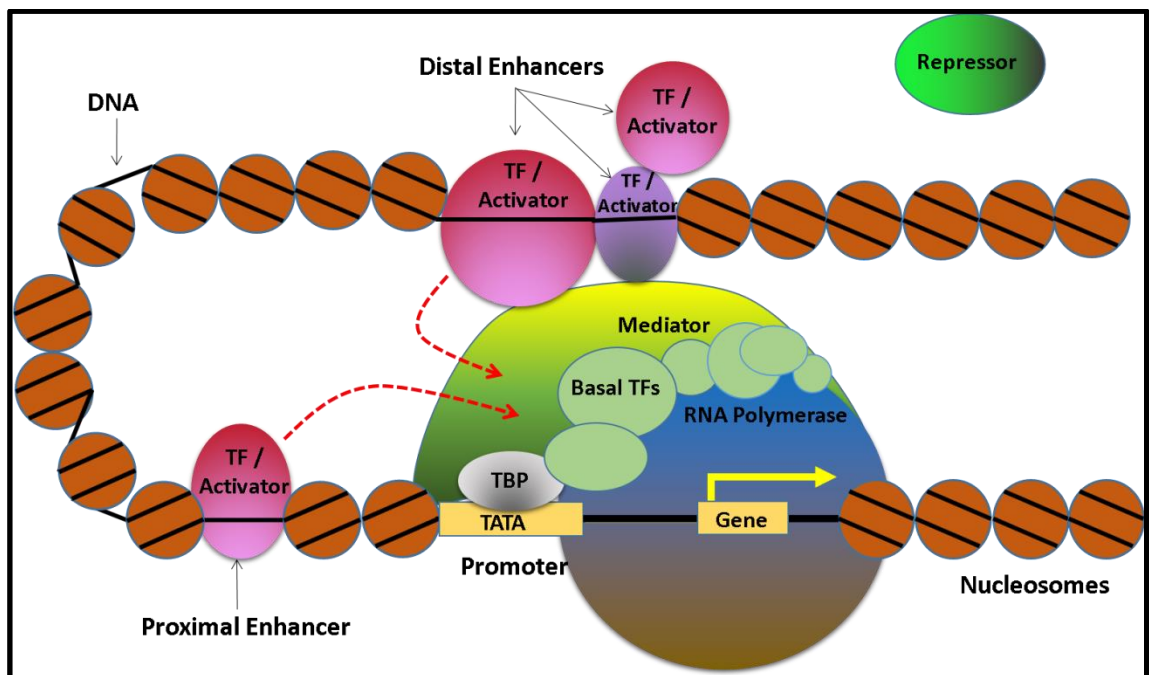
## 1.2. Chromatin

A complete set of DNA in an organism is known as a genome. The genome of a prokaryote is mostly contained in a single chromosome, which is usually circular, whereas, the genome of eukaryote is composed of multiple chromosomes, each containing a linear double helix molecule of DNA [6]. A human contains approximately three billion nucleic acids base pairs [7]. The completion of Human Genome Project in 2003 which was first articulated in 1988, had estimated there are about 19 000 to 20 000 protein coding genes in the human genome [7, 8]. Genes encode protein and cell functions. The number of human protein coding genes is only moderately bigger than that of much simpler organisms (e.g. fruit fly; ~13 600 genes [9], round worm; ~20 500 genes [10]). Thus, the size of the genome does not reflect the complexity of an organism.

Apart from this, a gene can also yield different proteins because of mRNA alternative splicing [6, 11]. In humans, each cell contains about 2 meters of DNA when (all chromosomes are included), and this long stretch of DNA can fit into the microscopic space of eukaryotic nucleus with the help of proteins known as histones which compact chromosomal DNA. The complex of DNA and histones formed then is called chromatin. Histones (positively charged proteins, i.e. H1, H2A, H2B, H3 and H4) alter the negatively charged DNA conformation by interacting with it and coiling and folding the DNA. DNA is packaged by a repeating unit of histones octamer which packages 147bp of DNA to give the nucleosome -a nucleosome is a unit of chromatin.

### 1.2.1. Transcription and transcription control

The control of the transcriptional process is important in dictating the amount and which proteins will be produced. The timely expression of genes in a cell results from transcription regulator activity and gives rise to a variety of cell types. As mentioned previously, eukaryotic transcription is more complex than prokaryotic transcription where the transcription of eukaryotic gene has to deal with introns, the presence of activators/repressors as well as chromatin accessibility. Here we will deal mainly with eukaryotic cells, since these are the main subject of the remainder of the thesis.



**Figure 1.3** Modulation of transcription of a eukaryotic gene in an active state.

Figure 1.3 above shows the transcriptional regulation machineries in which the basal TFs and RNA polymerase II are recruited by transcription factors on a distal enhancer to the core promoter where the TATA box lies adjacent to the transcription start site (TSS). This distal interaction with the help of chromatin remodelers. Chromatin remodelers form a

loop which then initiates the transcription [12]. These kind of looping interactions have been captured using genome-wide chromosome conformation capture (Hi-C) and related techniques [13].

### **1.2.2. Sequence-specific TFs and combinatorial regulation**

The transcriptional regulator also known as a transcription factor (TF) can either be an activator, a repressor or silencer (trans-acting elements) that bind to a particular binding motif in the non-coding DNA regions, which are also known as cis-regulatory elements (CREs). A TF is often composed of two domains: DNA-binding domain and activating regions which recognize and bind to DNA and interact directly/indirectly with the basal transcription machinery and other factors. When there are more than one or combinations of cis-regulatory elements occurring in regions on the DNA, they are called cis-regulatory modules. Cis-regulatory modules can affect transcription independently of location relative to the promoter [14].

Current studies have addressed the combinatorial regulation of transcription by combinations of TFs and their co-regulator(s) (see Figure 1.3 for illustration of combinatorial binding) in exerting effects on gene expression, in many model organisms such as man, mouse, and fruit fly genomic systems [15-18]. TF binding patterns are important in elucidating the biological function of organisms. While a TF can bind onto many CREs, many CREs can also be bound by more than one TF, and not all binding events are important/relevant for genes expression. These prove to be a challenge in inferring functional regulation by TFs.

### **1.2.3. Epigenetics and its relationship to gene regulation**

We as human have the same number of genes in each of us, but we have different phenotypes. There are many factors that contribute to our diversity, from different spatial and temporal expression of genes and DNA differences (including mutations and copy number variations) in different cell types to the much less well-understood underlying factors such as environmental and epigenetic changes. Epigenetics is the second dimension to the genome and also known as epigenome, and it contains key information specific to every cell type [19]. Epigenetics literally refer to 'outside conventional genetics' [20] or in other words any process other than DNA sequence itself that modulate gene activity in a cell. Many reviews that discuss epigenetics have explained in detail the epigenetic alterations that could lead to downstream biological effects which include transcription factors, non-coding RNAs, DNA methylation, and histone modifications along the genome [19-21].

#### **1.2.4. Chromatin modifications and DNA methylation**

As stated above, the primary components of chromatin are histones. Histones make up the chromosome and any alteration in histones will effect the DNA packaging and its accessibility. Chromatin in an active form for example, as in Figure 1.3, is where the important regulatory regions such as promoters and enhancers are more accessible, to allow transcription to occur. Histones can be post-translationally modified by acetylation and methylation of conserved lysine residues on the amino-terminal tail domain of histone proteins [22]. Active (open-chromatin) and inactive (heterochromatin) states of chromatin are usually associated with acetylation and deacetylation of histones respectively. On the other hand, methylation of different lysine residues of a histone protein can be markers for both active and inactive chromatin states. For example, methylation of lysine (K9) of H3 histone marks the silent DNA in heterochromatin. In contrast, methylation of lysine (K4) on the same histone protein marks active chromatin [22]. Similar to chromatin modifications by histones, DNA methylation of the promoter regions can lead to an aberrant transcription. Some promoter regions in mammalian genomes contain short regions (0.5-4 kb in length) which are rich in cytosine and guanine nucleotides also known as CpG islands [23] and mostly located proximal to the TSS. The addition of a methyl group cytosine residues, known as methylation often results in the repression of transcription. In normal cells, almost half of the genes promoters are found to be unmethylated [23].

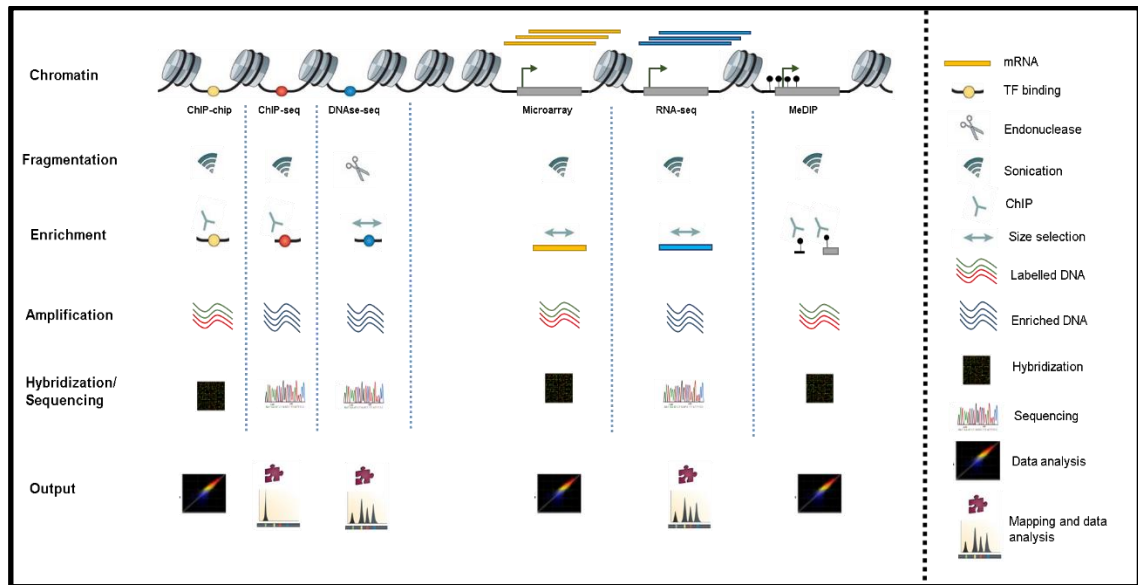
#### **1.2.5. Open questions in eukaryotic gene regulation**

Some open questions in eukaryotic gene regulation that need to be answered are, 1) With the combinatorial binding of transcriptional factors, how do we differentiate between relevant and irrelevant TF binding which actually drive the transcription?, 2) More than one epigenetic alteration could be responsible for the altered gene expression levels, thus, what is the best way to compare epigenetic alterations to the expression levels observed?

### **1.3. Experimental techniques for studying regulation**

All of the possible ways of regulating mRNA levels mentioned above can be measured using high-throughput technologies (i.e. array based or sequencing based technologies). Microarrays and array techniques were the pioneering technologies to quantify the abundance of transcripts as well as estimating the physical interaction of TF with DNA. These have now been superseded by the development of the next-generation sequencing (NGS) technologies. NGS technology is a platform to do massive parallel short-read DNA sequencing and this comes at greater reduction of cost per base since

the first final draft of human genome was completed [7]. Both, microarray and NGS technologies adopt a similar framework in their library preparation.



**Figure 1.4** A genomic locus analysed by corresponding chromatin profiling experiments.

A genomic locus analysed by corresponding chromatin profiling experiments. Order of steps varies on different chromatin profiling experiments. Figure was adopted from [24].

Figure 1.4 above shows different types of experiment that can be done to analyze a genomic locus for TF binding (ChIP-chip and ChIP-seq), chromatin accessibility (DNase-seq), transcripts/mRNA expression (Microarray and RNA-seq) and DNA methylation (MeDIP). The general consensus of steps in NGS and microarray includes fragmentation of targeted DNA using sonication or nucleases, enrichment of targeted DNA or reverse transcribed RNA (complementary DNA, cDNA), amplification of enriched cDNA/targeted DNA using PCR to prepare a sufficient cDNA/targeted DNA library. The library products are then either hybridized to an array or are sequenced. Reads produced by the sequencing technique need to be mapped to a reference genome before being quantified and analyzed. This is contrast with microarray where array probes are known or already annotated with locus information.

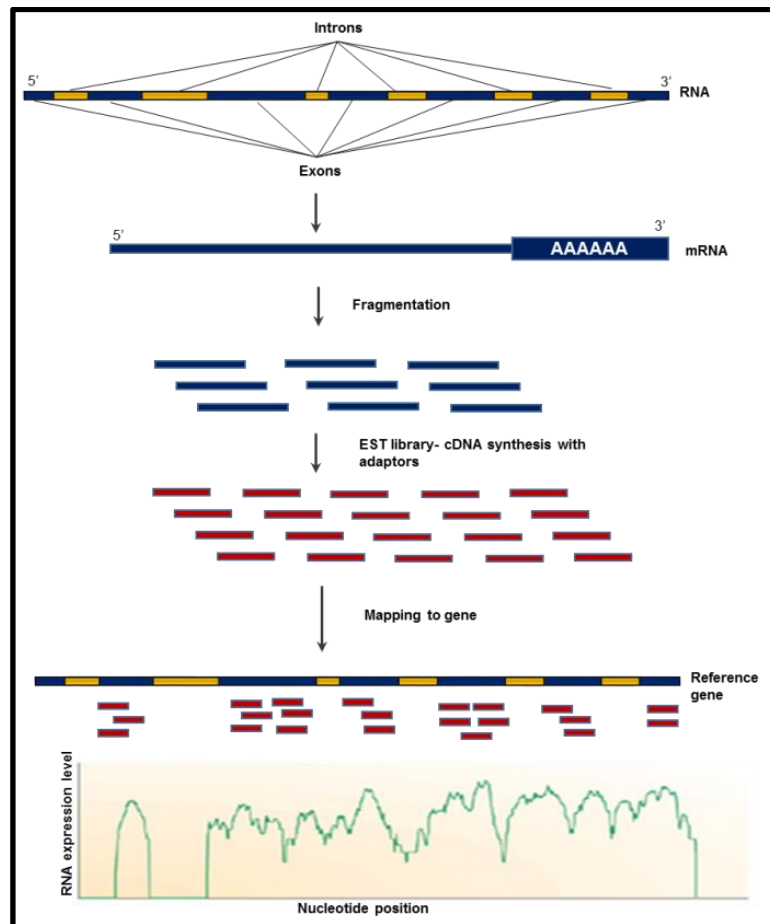
### 1.3.1. Microarrays and RNA-seq

Array-based technologies provide researchers with robust tools to measure the binding of TFs and the expression of genes, where thousands of probe intensities are analyzed in a single assay. Microarray probes represent collections of promoters, coding regions, transcripts 3' ends, alternatively spliced exons, single nucleotide polymorphisms (SNPs) and disease-gene arrays [25]. Microarrays have been used for decades in profiling gene expression. Thousands of oligonucleotides (short cDNA molecules (25 to 60-bp)) are 8

immobilized on a solid support where either the array is spotted-on-glass array, an in-situ synthesized array, or self-assembled microbeads in micro-wells [26]. In microarray technology, poly (A)-tailed mRNAs are first isolated, reverse transcribed to cDNA, purified and enriched, and followed by amplification by PCR and labeling (fluorescent tag or biotin). The labeled sample is then hybridized to a microarray. Depending on the type of labeling, the array is treated and scanned differently. A biotin-labelled sample array is stained with fluorescently labeled streptavidin to label cDNA and the fluorescent signal at each spot is measured. If different fluorescent dyes (Cy5- red and Cy3- green- to label cDNA samples from two different experimental conditions) are used, the differently labeled arrays are scanned using lasers with two different wavelengths corresponding to the dyes and the intensities are measured.

There are several limitations in performing microarray such as its design requires a priori knowledge of the genomic features, cross-hybridization between similar sequences, high signal-to-noise ratios, bias from PCR-based amplification and reproducibility of microarray data due to many formats, methods and analytical approaches available [25].

Next generation sequencing based approaches have overcome the limitations of microarray in many ways. For example, it has been used to assemble a genome de novo-without any a *priori* knowledge of genomic features. Moreover, DNA is directly sequenced hence, removing any issue of cross-hybridization. In addition, signals are quantified by counting the sequence tags rather than using relative measures between samples, and minute amounts of sample (nanograms) are sufficient, often without the need of PCR amplification [25]. However, the prominent feature of NGS is that the binding of TF (ChIP-seq) to the DNA and RNAs abundance (RNA-seq) data are collected genome-wide and from all genomic regions. Last but not least, NGS is more reproducible in term of the bioinformatics analysis where all NGS platforms have same data output and in similar formats.



**Figure 1.5** Workflow of RNA-seq from library preparation to RNA profiles quantification.

Workflow of RNA-seq from library preparation to RNA profiles quantification. Figure was redrawn based on [27].

Briefly, in the RNA-seq process, total mRNAs undergo a fragmentation process and are reverse transcribed into cDNA fragments. Adaptors are added to the 5' end of each cDNA (red fragments in Figure 1.5), these fragments can be exons, junctions between exons, and poly (A) tails. Then, using high-throughput sequencing, short sequences obtained from cDNA fragments known as short reads are mapped to the reference genes [27]. cDNA fragments are generally sequenced at the 5' ends. However, they can also be sequenced at both ends. These reads are used to generate a nucleotide resolution expression profiles for each gene as represented in Figure 1.5 (bottom most section). The abundance of reads (target transcripts) is quantitatively approximated in the form of counts. By taking the sequencing depth and other technical biases from RNA-seq preparation steps as well as the transcript lengths into consideration, normalization methods were developed known as reads per kilobase per million (RPKM) and fragments per kilobase per million (FPKM).

$$\text{RPKM} = \frac{\text{number of mapped reads}}{(\text{length of transcript in kilobase})/(\text{million mapped reads})}$$

$$\text{FPKM} = \frac{\text{number of fragments}}{(\text{length of transcript in kilobase})/(\text{million mapped reads})}$$

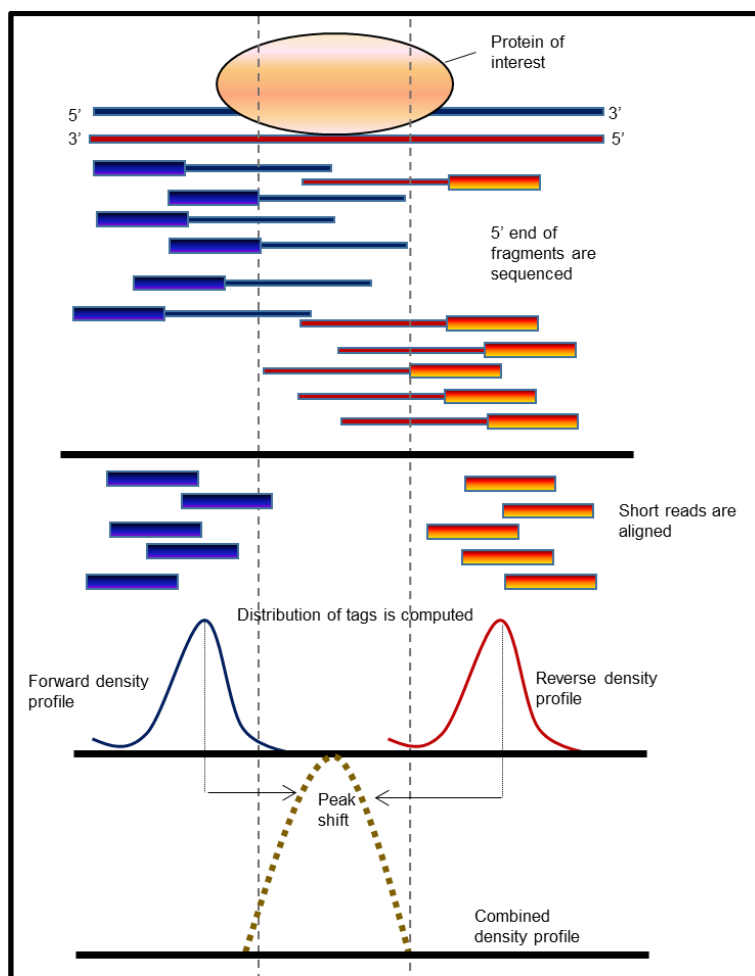
RPKM and FPKM are analogous to each other and were made for single-end and paired-end sequencing respectively. In RPKM, each read corresponds to a single fragment that was sequenced whereas, the latter one involves both 5' ends reads of the same fragment that was sequenced. For FPKM, fragments are used to approximate the abundance of transcripts rather than read counts so fragments are not counted twice [28].

### 1.3.2. ChIP-chip and ChIP-seq

Chromatin immunoprecipitation on a chip (ChIP-chip) has a similar overall framework to the gene expression using microarray except that at the earlier stage in library preparation, formaldehyde is used to crosslink the DNA binding protein to DNA, followed by sonication to shear the bound DNA. The DNA fragments are subjected to an immunoprecipitation reaction where an antibody specific to the bound protein is used to precipitate the protein-DNA complexes [29]. Steps following the purification of the bound DNA after immunoprecipitation are similar to the gene expression microarray.

Chromatin immunoprecipitation followed by sequencing offered higher resolution, less noise, and more coverage than ChIP-chip and can be used to profile DNA-binding proteins, histone modifications or nucleosomes on a genome-wide scale [30]. In ChIP-seq, the crosslinking of DNA-binding protein to DNA, fragmentation, and immunoprecipitation steps are similar to ChIP-chip. The immunoprecipitated fragments are sequenced from the 5' end. Sequenced reads are aligned to the genome (e.g. using Mapping and Assembly with Qualities (MAQ)) [31]. This is followed by the identification of enriched regions relative to the control with statistical significance using a peak caller (e.g. MACS) [32]. As tags are sequenced from both strands, the alignment of the sequenced tags to the genome results in two peaks (one on each strand) that flank the binding location of the protein or nucleosome of interest [30].





**Figure 1.6** Representation of ChIP-seq of a DNA-binding protein on DNA. Representation of ChIP-seq of a DNA-binding protein on DNA. Figure was redrawn with permission from [30].

This should form two distributions (forward and reverse density profiles) and a smoothed profiles from each strand are combined by shifting each profile to the center [30]. Peaks then can be scored using different models for the tag distribution (e.g. Poisson model or the binomial model).

#### 1.4. Cancer

Cancer cells are different from normal cells in that they no longer respond to cellular growth and death signals and are independent of signals from other cells [11]. A group of cancer cells forms tumour and in the early stages of cancer, tumours are benign or occupy a specific region of tissue. However, as these benign tumours accelerate in growth, they can spread to other tissues or organ systems and become malignant. This process of invasion is known as metastasis [11]. There are many possible ways for normal cells to transition into cancerous cells. Cancer cells often result from the accumulation of mutations and copy number aberrations (CNAs) of genes that control

pathways related to cell proliferation. The main challenge in understanding mutations and CNAs is to distinguish the driver events that contribute to cancer progression from the passenger mutations and CNAs. Apart from mutations and copy number aberrations, it is well-known that epigenetics also contributes to human disease.

#### **1.4.1. Basics of cancer hallmarks and mechanisms**

The six hallmarks found to be influencing cancer that were originally proposed by Hanahan and Weinberg in their review on cancer hallmarks [33] are as follows:

1. Sustaining proliferative signaling
2. Evading growth suppressors
3. Activating invasion and metastasis
4. Enabling replicative immortality
5. Inducing angiogenesis
6. Resisting cell death

The ability of cancer cells to sustain proliferation independence may be acquired through deregulation of growth-promoting signals, deregulation of receptor signalling, constitutive activation of signalling pathways operating downstream of growth ligand receptors, constitutive activation of signalling circuits triggered by somatic mutations, and compromised negative-feedback loops that are supposed to weaken the proliferative signalling [33]. In contrast to the cell proliferation independence, disruption in negative regulation of cell proliferation usually, by the tumour suppressors (i.e. Tp53- a gene that provides instructions for making a protein called tumour protein *p53*) can hamper the decisions of cells to either proliferate or commit to apoptosis/death. Cell adhesion formed by dense populations of cells using, for example, E-cadherin is abolished in cancer cells [33]. Loss of E-cadherin affects tissue integrity which could lead to tumour abnormal cellular architecture and invasion [34]. In addition to uncontrolled cell proliferation, cancer cells progress in malignancy by invading local and distant cells or tissues (metastasis). This involves a multistep process where at first, localized invasion by cancer cells enter the nearby blood stream and lymphatic vessels (intravasation). Cancer cells in these vessels will transit through lymphatic and blood systems and then escape into distant tissues (extravasation) and forming a small colony of cancer cells. This small colony will grow into tumors (metastases), and this final step is known as colonization [33].

In normal cells following embryogenesis, the sprouting of new blood vessel from the existing one (angiogenesis) is largely dormant except during wound healing and the female reproductive cycling system [33]. In contrast, angiogenesis is always activated

during tumor progression to help sustain tumor growth. As mentioned previously, Tp53 plays an important role in triggering cellular apoptosis. This is mainly by up-regulating several apoptotic factors. Another oncoprotein such as Myc plays a pivotal role in cell growth, increasing cell proliferation, tumorigenesis and in reprogramming stem cell state [35]. Tumor cells evolve by either loss of Tp53 or activation of Myc, increasing the expression of anti-apoptotic regulators, increasing survival signals, downregulation of apoptotic factors, or by disrupting the ligand-induced death pathway [33].

Clinicians are trying to diagnose, prevent and/or eliminate cancer by implementing invasive surgery together with the therapeutic treatments such as radio-, chemo-, immuno-, targeted-, and/or hormone-therapy, as well as precision medicine. Parallel to the standard therapeutic strategies which generally kill proliferative cancerous cells as well as non-malignant fast-growing cells (e.g. hair, intestine, and buccal cells), targeted-therapy involves small-molecule drugs that can enter cells to targeting specific targets (e.g. oncogenic drivers) inside cancer cells or monoclonal antibodies that specifically attach to the outside of cancer cells without destroying the healthy cells [36]. Targeted-therapy underpins the precision medicine where clinicians and scientists are trying to tailor the suitable drugs based on patient's genetic aberrations (e.g. genes expression, mutations, methylations, and copy number) that drive cancer or block their effects [36]. By understanding precision medicine, more systematic and accurate diagnosis and cancer treatment could be realized in the future.

#### **1.4.2. Different types of genetic aberrations**

##### **Somatic mutations**

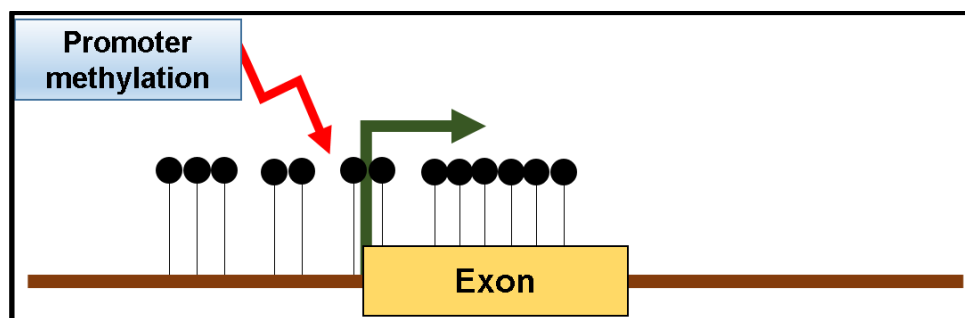
Humans are all unique because we all have small variations in our genetic code. Genetic variations can either be inherited from our ancestors or occur during our life from a variety of sources such as from the exposure to radiation, chemicals or even just by chance. Variation in genetic codes can be from insertion- addition of nucleotide base(s), deletion-removal of nucleotide base(s) or/and substitution-change of a nucleotide base. Variations sometimes do not affect the normal function of cells but sometimes can be associated with cancer and human diseases. Different cancer types can have different mutation signatures, but some genes appeared to be frequently mutated across different cancer types, for example, the tumor suppressor genes, Tp53 [6, 11]. Any mutations occurring in the cell in developing somatic tissue are not transmitted to progeny. On the other hand, mutations in the germ cells which used during reproduction may be transmitted to progeny.

### Copy number variations

Somatic copy number variations (CNVs) play a major role in most of cancer types. They act by activating oncogenes and deactivating tumor suppressors [37]. In diploid organisms such as man, our chromosomes have two copies of each gene. However, recent discoveries have revealed that large segments of DNA have varied copy number of genes (e.g. only one and more than two copies) and this could lead to biological imbalances and diseases [37, 38]. The imbalance in copy numbers affects each individual differently. For example, an individual who is heterozygous for a tumor suppressor gene, loss of the normal allele will produce a locus with malfunctioning tumor suppressor protein also known as the loss of heterozygosity (LOH). LOH is common in cancer.

### Aberrant methylations

Hypermethylation of promoter regions in cells often silences the expression of linked genes. Figure 1.7 below shows how the hypermethylation blocks the proximal promoter from the binding of transcriptional machineries to initiate the transcription. DNA methyltransferase enzyme is responsible in adding a methyl group to the C5 position of the cytosine ring of DNA [22, 23, 39]. Aberrant methylations associated with cancer are widely studied, and it is found that disruption in the maintenance of methylation patterns causes the inactivation of broad range of genes including tumor suppressor gene expression. Different patterns of DNA methylation also correlate with the expression of genes in different cancer types [40].



**Figure 1.7** How DNA methylation silences a gene.

Hypermethylation of promoter regions in cells often silences the expression of linked genes. Figure 1.7 above shows how the hypermethylation blocks the proximal promoter from the binding of transcriptional machineries to initiate the transcription. DNA methyltransferase enzyme is responsible in adding a methyl group to the C5 position of the cytosine ring of DNA [22, 23, 39]. Aberrant methylations associated with cancer are

widely studied, and it is found that disruption in the maintenance of methylation patterns causes the inactivation of broad range of genes including tumor suppressor gene expression. Different patterns of DNA methylation also correlate with the expression of genes in different cancer types [40].

### **1.5. Genomic projects – TCGA, etc.**

An abundance of datasets has been produced from high throughput technologies, and these are driving more novel discoveries associated with human and other model organisms' transcriptional regulation as well as the regulation and subtypes of cancers in human. This usually involves collaboration of many research groups around the world such as the Encyclopedia of DNA Elements (ENCODE) and The Cancer Genome Atlas (TCGA) consortia. The benefit of such efforts is that data are made publicly available to the research community for them to come up with novel findings or comparison of common patterns existing in the publicly available data and locally generated data [41, 42]. In addition, publicly available datasets can be used to test newly developed algorithms.

The ENCODE project is an international collaboration of research groups funded by the National Human Genome Research Institute (NHGRI) [41]. It provides information about functional epigenetics of the human genome that is vital to the development and function of a human. This includes data on DNA methylation, histone modifications and TF binding that influence mRNA production. With the advancement of techniques, ENCODE also examines the accessibility of the genome using the DNA-cleavage protein DNase I as well as long-range chromatin interactions that could change the chromatin structural conformation and thus affect transcription [41].

The Cancer Genome Atlas (TCGA) is a collaboration between the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI) that currently has generated multi-dimensional maps corresponding to different key genomic changes in thirty-three types of cancer [42]. Different types of data including DNA methylation, gene expression sequencing data, microRNA and protein expression (RPPA) have been made publicly available. The available and readily accessible data allows the cancer research community to improve the prevention, diagnosis, and treatment of cancer accordingly. New methodologies to learn and predict the prognosis of the cancer markers discovered using this datasets have contributed considerably to the diagnosis and treatment for cancer patients.

## **1.6. Data analytical techniques**

ChIP-seq and RNA-seq have produced wealth of information on the interaction of DNA-binding proteins to DNA and the expression levels of genes, respectively. Some of them are irrelevant or do not contribute to the biological processes affecting the function of an organism. Hence, suitable methods need to be tailored to separate relevant from all irrelevant and relevant information and to put relevant information together (e.g. clustering, dimensional reduction and machine learning).

### **1.6.1. Clustering**

Clustering is an approach to extract useful information from a large dataset by collectively grouping objects which have similar features/properties together and separate them from other groups which are dissimilar. Clustering has been used earlier by chemists and biologists in constructing the periodic table of chemical elements as well as the classification of animals and plants into the hierarchies of kingdom, phylum, class, order, family, genus, and species [43]. Clustering provides researchers with an alternative to explore patterns existing in high dimensional data derived from microarray and next generation sequencing technology. With the advancement of algorithm development with more sophisticated computational techniques, various data types can be analysed and visualised, and integration of different datasets often promotes the understanding of the biological function of genetic components which make up the observable phenotypes effects in organisms.

#### **1.6.1.1. Brief background on clustering methods**

Clustering of biological information (e.g. gene expression) could shed light into the regulation of genes as well as the functional biochemical components inside the gene regulatory network [44]. It has been used widely to annotate gene functions, predict diseases, derive gene regulatory networks, and guide the direction of experiments, as well as generating new hypotheses for further investigation [44, 45]. A process of organizing multivariate data into different classes is often achieved using a clustering algorithm [46]. Unsupervised clustering procedures (i.e. k-means, hierarchical clustering and self-organizing maps, see Figure 1.8 below) are the main methods used in analyzing genomic data compared to supervised machine learning due to its simplicity, and the fact that less computational power is required.

### 1.6.1.2. The three main clustering methods

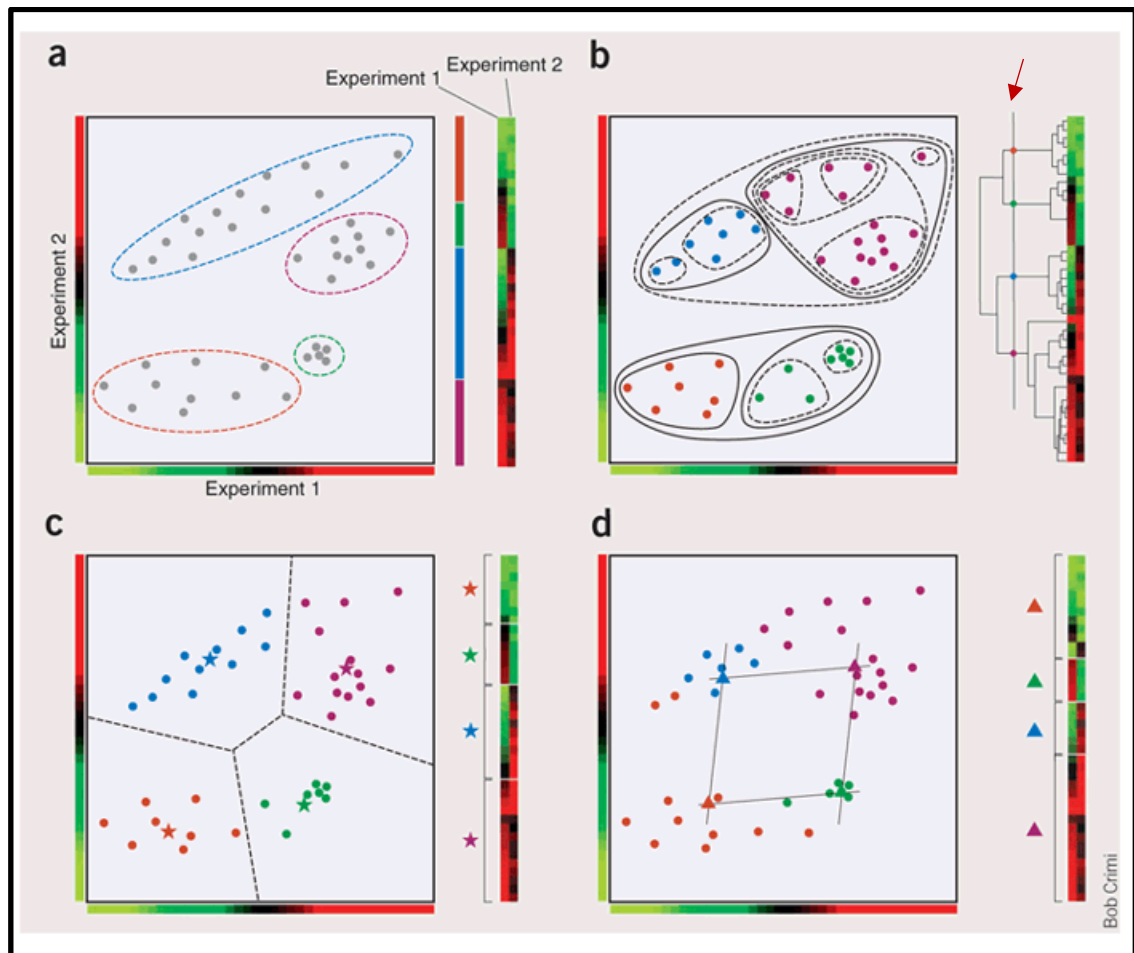
Many similarity measures can be adopted in calculating the distance or similarity between objects, such as Euclidean distance (Equation 1) and Pearson correlation (Equation 2) below. A matrix of vectors can be described as  $n \times d$  matrix, where  $n$  is a set of objects (e.g. genes) and each object represented by a set of  $d$  measurements/conditions (e.g. gene expression value across time points). For example, when clustering genes with expression values across time points (expression patterns), the clustering algorithm will calculate the distance between the gene expression patterns of pairs of genes and find the shortest distance between any two genes to be recognised as exerting a similar expression patterns.

$$d_{fg} = \sqrt{\sum_c (e_{fc} - e_{gc})^2} \dots \dots \dots \text{Equation 1}$$

Here,  $d_{fg}$  is the Euclidean distance between object  $f$  and object  $g$ .  $c$  is the condition and  $e$  is the measurement value.

$$d_{fg} = 1 - r_{fg} \quad \text{with} \quad r_{fg} = \frac{\sum_c (e_{fc} - \bar{e}_f)(e_{gc} - \bar{e}_g)}{\sqrt{\sum_c (e_{fc} - \bar{e}_f)^2 \sum_c (e_{gc} - \bar{e}_g)^2}} \dots \dots \dots \text{Equation 2}$$

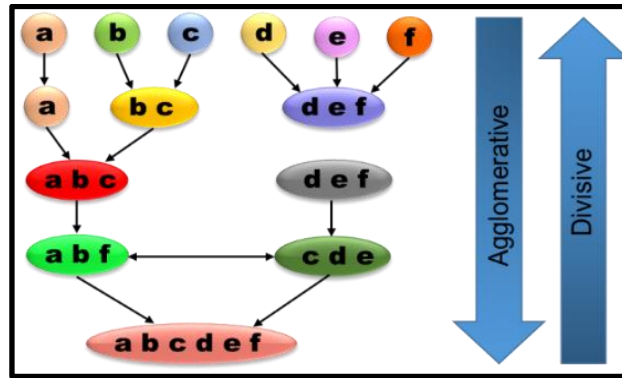
Here,  $d_{fg}$  is the Pearson correlation between object  $f$  and object  $g$ .  $\bar{e}_g$  is the mean of object  $g$  and  $\bar{e}_f$  is the mean of object  $f$ .  $e_{fc}$  and  $e_{gc}$  are the measurement values of object  $f$  and  $g$  under condition  $c$ , respectively.



**Figure 1.8** Clustering methods applied for 40 genes which have been measured under two different experimental conditions. Clustering methods applied for 40 genes which have been measured under two different experimental conditions. **a)** Known underlying cluster set of four clusters. **b)** The hierarchical clustering produces different level of clusters and the dendrogram shown on the right side of figure was cut at the level which produced four clusters. **c)** The partitioning method using k-means (k=4) based on the shortest distance to the cluster centroids (shown using stars). **d)** The self-organising map (SOM) method find clusters which are organized into a grid structure. Figure was reproduced with permission from [47].

There are two main approaches to cluster data: partitioning the data and hierarchical based approaches. Both types of clustering require either distance or correlation measurements to calculate the similarity and dissimilarity between and within clusters respectively. There are two sub-methods in the hierarchical based approach namely, agglomerative or bottom-up and divisive or top-down (shown in Figure 1.9) in building a hierarchical clustering tree.





**Figure 1.9** Top-down (agglomerative) and bottom-up (divisive) strategies in hierarchical clustering of six data points in this example.

Here, partitioning of data set can be obtained by cutting the dendrogram at a certain level (e.g. red arrow in Figure 1.8 b); usually the level where there is a large difference in the dendrogram would give a unique clustering. However, each cluster can have sub-clusters even after cutting the dendrogram at a certain level, and the dendrogram might need to be cut at different levels for each branch to obtain sensible partitioning. In contrast to hierarchical clustering, k-means divides the data set into clusters by trying to minimize an error function (Equation 3) using pre-defined number of clusters (centroids) [48]. If the number of clusters is unknown, a k-means algorithm can be repeated for a set of different number of clusters, typically from two to  $\sqrt{N}$  where  $N$  is the number of samples in the data set [47, 48]. In a k-means algorithm, the error function is minimized

$$E = \sum_{k=1}^C \sum_{x \in Q_k} \|x - c_k\|^2 \dots \dots \dots \text{Equation 3}$$

where  $C$  is the number of clusters,  $c_k$  and  $x$  are the center of and sample in cluster  $k$ ,  $Q_k$  respectively. There are several approaches on how  $c_k$  could be initialized. For example, we could start with a hierarchical clustering, cut the dendrogram at a certain level and use the mean calculated from each cluster produced. As for self-organising maps (SOMs), a user-defined number of centroids is required and these centroids are linked via a grid structure. At each iteration, a gene is chosen, and instead of moving the genes to the centroid, the centroid closest to the gene will be moved towards it as well as its neighboring centroids on the grid. Over time or iterations, the flexibility of grid of centroid will be reduced as the radius of this neighborhood shrinks resulting in a grid of cluster (see Figure 1.8 d ) [47]. The neighboring clusters found using SOM show related expression patterns.

An unsupervised model-based clustering is a generalization standard clustering method mentioned above where each object has a different degree of membership in all clusters and can also be based on mixture models [48]. Here, the data are assumed to be generated by probability distributions (e.g. a Gaussian distribution), and the parameters

of the probability distribution can be estimated using, for an example, the expectation maximization method. Data points are assigned to different clusters based on their probabilities in these distributions.

Of all the methods discussed above, mixture model-based clustering provides more natural weighting of the underlying clusters as it takes into account of the different degree of membership of a data point in all clusters based on the parameters estimated from the data. This will be discussed in more depth in the next chapter.

### **1.6.2. Dimension reduction**

The high dimensionality of data sets and types per sample that we have make it difficult to currently visualize and limit the data exploration. However, a reduction to only most important features has been proven to be useful using principle component analysis (PCA) [49, 50]. PCA is a mathematical algorithm that reduces the dimensionality of data using a feature selection approach. The features (e.g. genes) are reduced by projecting each sample into different planes/directions called principal components (they are statistically independent from each other) in the n-dimensional plot (number principle components is less than the number of features). Then, the principal components that could explain the variations best in the dataset are selected [49]. Each principle component will have its proportion of variance based on the features contained within it. Selection to only subset of components that could capture ~90% of the original variance would reduce the number of features while retaining most of the variation in the dataset. In addition, it is also common practice to reduce the dataset to only most variably expressed genes as these genes usually being the key players in important cellular pathways (i.e. cells differentiation pathway, tumorigenesis pathway) in comparison with the housekeeping genes expression.

### **1.6.3. Supervised analysis (machine learning)**

Machine learning is a computational algorithm that improves the outcomes with experience. It is one of the useful methods for the interpretation of the genomic 'big data' by learning to recognize patterns in the problem given. One of the most useful uses of machine learning is to use input from the high-throughput technologies in distinguishing between different disease or sample phenotypes which leads to the identification of disease biomarkers [51]. In addition, it also has been used to predict gene expression using DNA sequence alone and sometimes to take into account other epigenetic information at the gene promoter [51]. There are three stages in the machine learning method. First is the development of the algorithm itself that would produce a successful learning. Second, divide or prepare the data set into a training set and a testing set. The

data points in training data set are annotated, and this process known as labeling, and the annotation is called 'label'. The labeled data points in the training data set are processed and stored as a model. Third, the testing data set are introduced to the algorithm, and which uses the model in second step to predict the labels. Successful prediction would predict most labels correctly. Given the known label of the testing set, the performance of the algorithm can be measured directly [51]. The learning process in machine learning algorithm can be in the form of generative models or and discriminative models (i.e. support vector machine). Given the two classes in the learning model, the generative approach constructs a full model based on the distribution of features in each class and then compares the difference between them. In contrast, the discriminative approach only focuses on modeling the boundary between those two classes [51].

The decision of researcher regarding which data to provide as an input to the algorithm is important, as the prior knowledge of the data relevance could produce a good prediction. For example, in cancer patient classifications, the feature selection in the learning step which involves finding subsets of genes based on gene expression measurements, could be valuable in providing accurate diagnoses for patients.

#### **1.6.4. Gene set analysis, GO, GSEA**

Clusters found using unsupervised/supervised clustering are usually subjected to a systematic evaluation to see if the objects in the same group are biologically correlated. Gene ontologies (GO) and gene set enrichment analysis (GSEA) are the most widely used criteria to see if the members of clusters are biologically homogeneous or not. The GO consortium provides a standard system to produce a structured, precisely defined, common, controlled vocabulary for describing the roles of genes and gene products in any organism [52]. There are three categories of GO namely, biological process, molecular function, and cellular component. Biological process refers to a biological objective to which the gene or gene product contributes. Molecular function is defined as the biochemical activity of a gene product. Lastly, cellular component refers to the place in the cell where a gene product is active [52]. There are several methods published to calculate the enrichment of GO terms for clusters such as The Database for Annotation, Visualization and Integrated Discovery (DAVID) [53] and GSEA [54]. Apart from GO terms, GSEA also incorporates eight major collections of annotated gene sets deposited as The Molecular Signature Database (MsigDB) [54]. A good clustering would result in clusters which have distinct and homogenous biological function enrichment between and within each cluster respectively.

## 1.7. Aims and objectives of the thesis

The main goal of the whole Ph.D. research was to develop, test and apply a model-based joint clustering algorithm which is generic in nature where it accepts binary (e.g. TFs binding, mutation status) and continuous (e.g. gene expressions) inputs and predicts the relationship between these inputs based on the clusters found. There are few effective tools (will be explained in the next chapter) existed to perform the analysis that our tool now allows. This involved the following steps,

- a) Develop a joint clustering algorithm using a mixture model and simulated annealing
- b) Generate and run the program using simulated data to optimise the runtime parameters
- c) Test the program using suitable publicly available data sets
- d) Infer the transcriptional regulation of genes and classification of cancer patients
- e) Biological and statistical evaluation of the clusters found
- f) Publish the software on a software repository in the form of command-line-interphase and graphical-user-interphase

In this thesis, there are four results chapters to cover all of the objectives stated above, and they are as follows:

Chapter 2: Developing and algorithm to jointly cluster binary and continuous inputs

This chapter covers the background, mathematical representation, and simulation of the model-based joint clustering algorithm.

Chapter 3: Modelling *S. cerevisiae* cell cycle transcriptional regulations using model-based joint clustering algorithm

Using relatively simple dataset of TF bindings and cell-cycle gene expression data from yeast, this chapter covers the benefits of integrating these data types in inferring the transcriptional regulatory networks in yeast cell-cycle.

Chapter 4: Application of model-based joint clustering to cancer data

In this chapter, we apply our method to another research area by identifying the sub-types of cancer from integration of mutation with the gene expression data from Acute Myeloid Leukaemia patients which are publicly available from TCGA.

## Chapter 2. Developing an algorithm to jointly cluster binary and continuous inputs

### 2.1. Introduction

Scientists have predicted that by 2025, with the advancement of high-throughput technologies and drop in sequencing costs, between 100 million and 2 billion human genomes could have been sequenced [55]. This will generate a massive amount of data, also known as genomic 'Big Data'. Clustering is one of the data mining methods always chosen by scientists to make sense of this massive amount of data. In our field, there are many examples of the need to cluster entities described by mixed variable types – mutations (discrete), binding (discrete), gene expression (continuous) etc.

The main statistical basis used in well-known clustering methods (i.e. k-means, self-organizing map, and hierarchical clustering) are either distance-, correlation-, or model-based clustering of a single data type (i.e. continuous, order, nominal or binary data types). Clustering of a mixture of data types on the other hand is a much more complex process and it is possible to execute this using a stepwise approach for distance/correlation based clustering. Model-based clustering, however, can be manipulated using a mixture model approach and accordingly, in a single step instead of stepwise approach.

We therefore set out to develop a method to cluster such entities that would be generically applicable to a range of different problems. Our specific goals were to generate a method able handle variable numbers and data set sizes common in the field, and where the optimum number of clusters is unknown and difficult to estimate, making manual experimentation impractical. We sought a method that would give clusters with clear biological interpretability, for instance a pattern of mutation or TF binding that relates to a shared pattern of expression in a cluster of genes.

In this chapter, we introduce a novel method to cluster entities described by combinations of binary and continuous variables, for applications to several different problems in genomics research. Bernoulli and Gaussian distribution for categorical variables and continuous variables, respectively will be applied to our mixture model clustering. Models found by clustering at this stage can be subjected to expectation maximization (EM) for further refinement, in our context this would be optimizing the clusters membership with the pre-defined cluster numbers. This method would be complementary to those discussed above and would be applicable to several realistic current problems.

### 2.1.1. Basic probability concepts

A probability is a measure associated with an event/outcome and how likely this is to happen and presented as follow,

$$\text{Probability of an outcome} = \frac{\text{number of ways the outcome can happen}}{\text{total number of possible outcomes}}$$

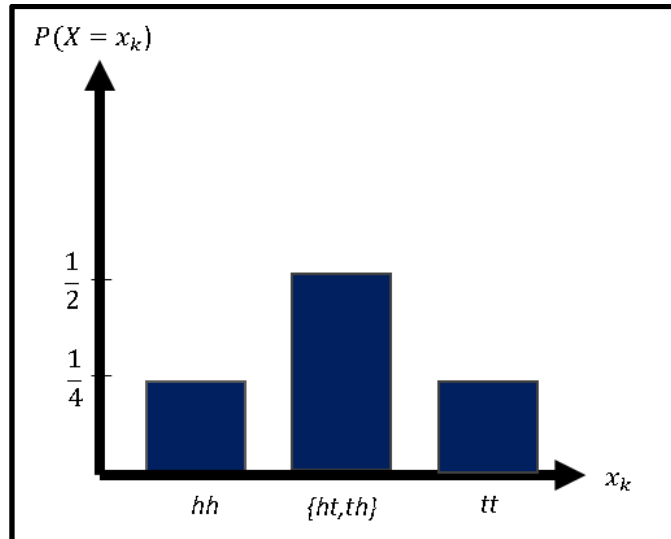
Taking the well-known example of tossing a fair coin, event A is the occurrence of a 'head' when a coin is tossed and the probability of event A will occur, denoted by  $P(A)$  would be the fraction of number of head to the all sides on a coin (a head and a tail) which equals to  $1/2$  or  $P(A) = 0.5$ .  $P(A)$  or probability of an outcome in general takes a value between 0 and 1,  $0 \leq P(A) \leq 1$  [56]. A probability density function is a probability measure that gives us probabilities of the possible values for a random variable. Given  $k$  number of tosses,  $X$  is a discrete/countable random variable (head or tail), a sample space,  $S_x = \{x_1, x_2, x_3, \dots, x_k\}$  are possible values of the random variable  $X$ . Here, we are interested in finding the probabilities of  $X = x_k$ . For example, if we toss a coin twice, a sample space we will get is,  $S = \{hh, ht, th, tt\}$  where  $h$  and  $t$  are head and tail respectively. Here, we are interested in probabilities of observing head(s). The probability density functions of  $X$  equal to  $hh$ ,  $\{ht, th\}$ , or  $tt$  are

1.  $P(hh) = \frac{1}{4}$

2.  $P(\{ht, th\}) = \frac{1}{4} + \frac{1}{4}$

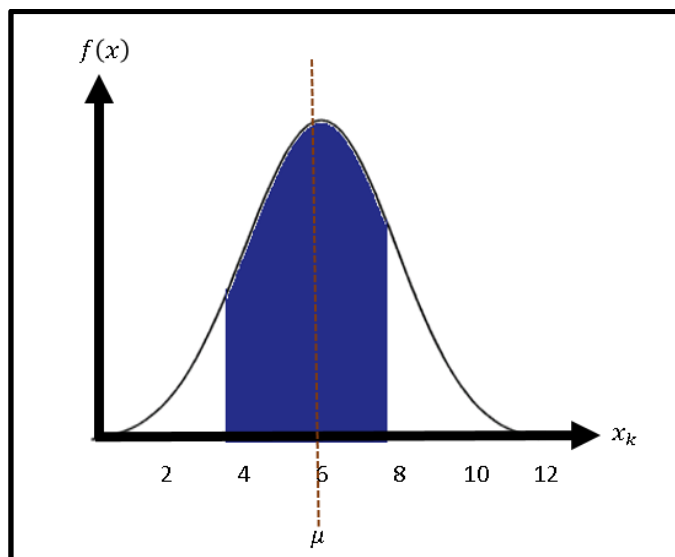
3.  $P(tt) = \frac{1}{4}$

The probability density functions above can be presented in the form of a probability density distribution, as in Figure 2.1 below.



**Figure 2.1** Probability distribution for random variables  $X = x_k$  from tossing a fair coin twice.

For discrete variables given in the example above as well as other types, including binary variables, the probability distribution does give the probability of each value (i.e.  $hh$ ,  $\{ht, th\}$ , and  $tt$ ). In contrast to discrete variables where we can assign probability to a single value, it is not possible to apply the same logic with the continuous variables. However, we can use probability density function to specify the probability of a random variable falling within a particular range of values or often represented as the area under the density function curve, above the horizontal axis and in between lowest and greatest values of the range (see Figure 2.2 below).



**Figure 2.2** Probability distribution for probability density function of random variables  $X = x_k$  from a list of continuous values,  $k$ .

A probability distribution function also denoted by  $f(x)$  rather than  $P(x)$  is a mathematical formula that gives the probability of each value of random variable which can be either continuous or discrete [56]. There are many statistical distributions established by statisticians to accommodate different underlying data type's and structures such as the Gaussian/normal, Bernoulli and mixture distribution.

### **Gaussian distribution**

The Gaussian/normal distribution is a very common continuous probability distribution and it is described by the bell-shaped curve which is symmetrical about the mean [56]. The probability density function of the Gaussian distribution is,

$$N(x) = \left( \frac{1}{\sigma\sqrt{2\pi}} \right) \exp\left(-\frac{1}{2} \cdot \frac{(x-\mu)^2}{\sigma^2}\right) \dots \dots \dots \text{Equation 4}$$

Here,  $x$  is the value of a random continuous variable, and  $\mu$  and  $\sigma^2$  are the mean and variance of  $x$ .

### **Bernoulli's distribution**

The Bernoulli distribution is a probability distribution of a binary random variable, where

$$B(x) = p^x(1-p)^{1-x} \quad \text{and} \quad p = P(x=1) \dots \dots \dots \text{Equation 5}$$

Here,  $x$  is the value of binary variable which is either '1' or '0' and  $p$  is the probability of getting '1'.

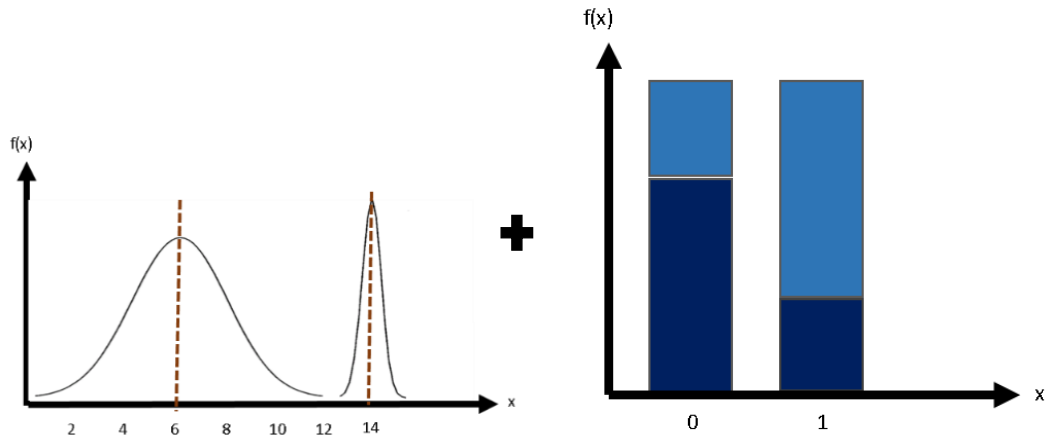
### **Mixture distribution**

Clustering of multiple data types simultaneously using a probabilistic approach can be done using a finite mixture model. A finite mixture model is a linear combination of two or more component probability distributions. It is a natural representation of populations thought to contain relatively distinct groups of observations where each observation can be characterized by a number of variables which may be binary, continuous ordinal or nominal [56]. Finite mixture distributions are of the form:

$$f(x; \theta_i) = \sum_{i=1}^k \alpha_i p_i(x; \theta_i) \dots \dots \dots \text{Equation 6}$$

Where  $x$  is a  $n$ -dimensional random variable,  $\alpha_i$  are the mixing proportions, where  $\sum \alpha_i = 1$  and  $p_i(x; \theta_i)$ , for  $i = 1, \dots, k$ , are the component densities and  $\theta$  is a set of component density parameters. Figure 2.3 below shows a representation of finite mixture distributions with  $k = 2$ , for multiple components or distinct groups of observations.





**Figure 2.3** Representation of finite mixture distributions for multiple components. Representation of finite mixture distributions for multiple components. Left: Finite mixture distributions for two normal components (one with lower mean and higher variation, and the other one with smaller variation and higher mean). Right: Finite mixture distributions for two Bernoulli components (lighter blue- higher  $p$  and darker blue- lower  $p$ ).

**2.1.2. Likelihood and maximum-likelihood**

A measurable characteristic of a population, such as a mean,  $\mu$  or standard deviation,  $\sigma^2$  is called a parameter. However, for a sample we acquired from a population, the measurable characteristic is known as estimated parameter. When the real population size is known and all are sampled ( $n \sim \infty$ ), we can calculate the probability of getting an observation using the distribution parameter(s) (e.g.  $N(x|\theta)$  for  $\theta = \{\mu, \sigma^2\}$ ). Often in a real-life scenario, it is almost impossible to sample data from all members of a population and most studies use finite sampling instead. Thus it is biased to use the word probability to represent finite sampling observations. Instead, a more proper terminology for this is ‘likelihood’. The likelihood of observing a set of parameters values,  $\theta$ , given observed outcomes  $x$ , is equal to the probability of those observed outcomes given those parameters values [57]. The log-likelihood of finite mixture distributions is of form:

$$\log L(P, \theta|x) = \log f(x|P, \theta) = \log \sum_{i=1}^k \alpha_i p_i(x; \hat{\theta}_i) \dots\dots\dots \text{Equation 7}$$

Where the left hand side of the formula is the likelihood of the parameters given the data and the right hand side is the probability of data given the estimated parameter,  $\hat{\theta}$ . According to Moon (1996), taking the logarithm of the likelihood often simplifies the maximization and yields equivalent results since log is an increasing function [58].

The main idea of maximum-likelihood (ML) is estimating the parameters of a distribution based upon observed data drawn according to that distribution. This involves finding parameter  $\theta$  for which the probability of observing  $x$  is as high as possible.

### 2.1.3. Information criterion in model selection

Application of ML in clustering using a pre-defined number of clusters, is used ultimately to calculate likelihood from a fixed number of parameters and is fairly straightforward. However, with heuristic approach where the number of clusters needs to be optimized comes the problem of comparing models with different numbers of parameters between different optimization steps. Likelihood values are not comparable when the models have different numbers of parameters. On average, a higher likelihood results when more parameters are introduced and we have to take account of this. Penalizing this would discourage the over-estimation of the number of parameters. Information criterion (IC) based clustering is one way where extra parameters can be penalized. A few model selection criteria exist in the literature and many model selection criteria are in the form:

$$O(L, k) = -2L + k\lambda(N) \dots \dots \dots \text{Equation 8}$$

Where  $L$  is the maximized log-likelihood,  $\lambda$  is a function of the number of data points  $N$  or in other word, cost for fitting an additional parameter, and  $k$  is the number of parameters in the model.

	Criterion	$\lambda(N)$	Equation	Reference
<b>a.</b>	AIC1	1	$-2L + k$	-
<b>b.</b>	AIC1.5	1.5	$-2L + 1.5k$	-
<b>c.</b>	AIC	2	$-2L + 2k$	Akaike, 1973
<b>d.</b>	AIC2.5	2.5	$-2L + 2.5k$	-
<b>e.</b>	AIC3	3	$-2L + 3k$	Bozdogan, 1993
<b>f.</b>	HQC	$2 \ln \ln(N)$	$-2L + 2k(\ln(\ln(N)))$	Hannan and Quinn, 1979
<b>g.</b>	AIC4	4	$-2L + 4k$	-
<b>h.</b>	BIC	$\ln N$	$-2L + k(\ln(N))$	Scwarz,1978
<b>i.</b>	CAIC	$\ln N + 1$	$-2L + k(\ln(N) + 1)$	Bozdogan, 1987

**Table 2.1** Penalty terms used in different information criteria.

Penalty terms used in different information criteria sorted ascendingly (from **a.** to **i.**) based on its stringency  $\lambda(N)$  in penalizing extra parameters and the number of data points in the model.

The most well-known information criterion was introduced by Akaike (1973) known as Akaike Information criterion (AIC) (see Table 2.1c.) and was used in estimating the Kullback-Leibler's distance between the estimated model and an underlying 'true' model of time series data [59]. Since then, this has given rise to several more information criteria which are based on different views of statistical theories in producing a parsimonious model [60]. The Bayesian information criterion (BIC) (see Table 2.1h.) was introduced later by Schwarz (1979) for the case of independent, identically distributed observations and linear models based on Bayesian point of view [61]. The penalty term of BIC,  $\lambda(N) = \ln(N)$  is more stringent than the penalty term of AIC,  $\lambda(N) = 2$ .

The major difference between AIC and BIC is that the BIC penalty increases with increasing  $N$  (more data) and BIC will always be the biggest penalty when  $N$  is large enough. This means that BIC converges to a single model as  $N$  gets very large, while AIC does not. Some statisticians like this property because it appeals to the idea that as the amount of data goes to infinity the model should converge on the 'correct' model. While the BIC penalty increases as  $\ln(N)$ , the likelihood term, on average increases faster (proportional to  $N$ ). Therefore with BIC, the likelihood dominates more over the penalty as  $N$  increases. Consequently, BIC tends to favor smaller number of parameters than AIC as  $N$  goes to infinity.

Bozdogan (1993) suggested using AIC3 criterion where  $\lambda(N) = 3$  instead of  $\lambda(N) = 2$  to get a minimum plausible model. AIC3 (see Table 2.1e.) was found to be the best criterion for selecting the number of latent classes with a binary dataset and it is a good compromise between AIC overestimation and BIC [62]. Bozdogan (1987) has proposed another version of AIC which penalises the number of parameters more heavily than AIC and BIC, known as consistent AIC (CAIC) (Table 2.1i.) [63].

Furthermore, if the penalty term increases quickly with an increasing number of parameters and  $N$  in the model, this will favour underestimation. Hannan and Quinn (1979) introduced Hannan-Quinn criterion (HQC) (see Table 2.1f.) that will underestimate the model parameters less for larger  $N$  than do CAIC and BIC. The HQC criterion is an attempt to keep the favourable statistical properties of BIC while reducing the rate of increase of penalty with  $N$  with more  $(\ln(\ln(N)))$  [64].

Since all of these information criteria have a minus sign (-) in front of them, the lower the  $O(L, k)$ , the better is the solution. By understanding the differences among the criteria and empirically testing them on our mixture model, a more succinct decision could be made on which is the best information criterion that should be applied.

### 2.1.4. Maximum-likelihood optimization

Maximum likelihood can be optimized or modelled either stochastically or deterministically. Usually, in deterministic modelling, with a known set of inputs, modelling will result in a unique set of outputs. On the other hand, stochastic modelling incorporates random inputs which then lead to random outputs. Generally, ML is deterministic when there is no missing information. In clustering however the information on which mixture component generated each data point is missing, and this needs to be handled by expectation maximization (EM).

EM is not fully deterministic because it often has to be run from several different start points to get the best answer. Maximum likelihood of mixture components can be modelled using EM or simulated annealing (SA) which will be discussed later. EM and SA are partially deterministic and stochastic ways of optimizing ML, respectively. As  $\theta$  is incomplete, we would like to find  $\theta$  to maximize  $\log f(x|P, \hat{\theta})$  by maximizing the expectation of  $\log f(x|P, \theta)$  given the data.

#### 2.1.4.1. Expectation maximization

EM can achieve clustering using a mixture distribution if the number of parameters (i.e. mixture components/clusters) is fixed. If the number of clusters is unknown, either EM needs to run with different candidate numbers of clusters, or some other solution is required. One possible solution is an optimization of the number of clusters, for example using simulated annealing.

Here, given initial parameter(s), EM maximizes  $\log f(x|P, \theta)$  by updating the initial parameter(s) using estimated data [58]. The basic steps involved in EM are as follows:

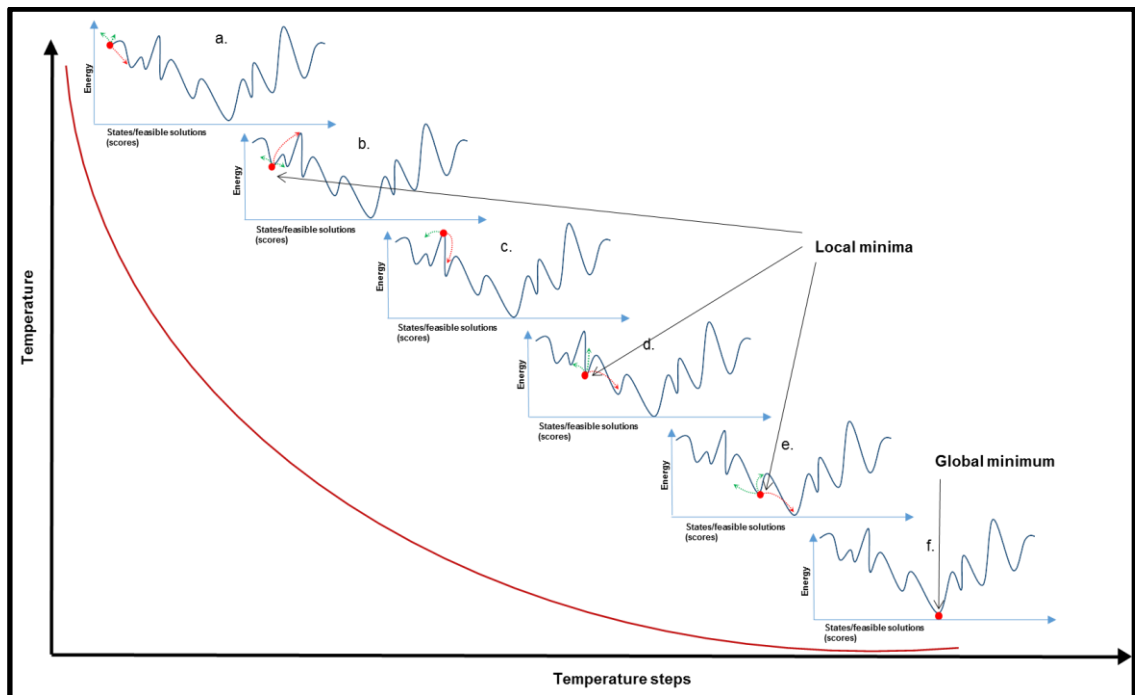
1. Choose initial parameter(s)  $\theta^{[0]}$
2. E-step : Estimate unobserved data using  $\theta^{[s]}$
3. M-step: Compute maximum likelihood estimate of parameter  $\theta^{[s+1]}$  using estimated data from step 2
4. E-step and the M-step are iterated ( $s \rightarrow s + 1$ ) until the parameter estimate has converged

### 2.1.4.2. Simulated annealing

Annealing is the process by which a metal cools and freezes into a minimum energy crystalline structure. Kirkpatrick and co-workers (1983) developed an algorithm known as simulated annealing (SA) which exploits the analogy between annealing of a metal and the search for a minimum energy rigid structure [65]. Using a random starting point, SA uses a random search that always accepts better (lower energy) solutions and occasionally accepts worse (higher energy) solutions. It uses also a control parameter  $T$ , which by analogy of cooling of a metal is known as the system "Temperature".  $T$  starts high and gradually decreases towards zero. At each temperature, the algorithm accepts higher energy solutions according to the acceptance probability,

$$P(\Delta E, T) = e^{-\frac{\Delta E}{T}} > R \dots \dots \dots \text{Equation 9}$$

where  $\Delta E$  is the increase in energy. Thus at high temperature many worse solutions are accepted, allowing the algorithm to escape from local energy minima, as illustrated in Figure 2.4 below.



**Figure 2.4** A graphical representation of the simulated annealing process. A graphical representation of the simulated annealing process. The red ball/circle is the solution and the small arrows in red and green are the moves which the red ball could make (solution acceptance). Here, at higher temperatures, bad move could be made (solution with higher energy) occasionally but not at the lower temperature.

As shown in Figure 2.4 above, SA explores the solution space and escapes local minima (false minimum) before reaching the global minimum solution. Given higher temperature steps and smaller temperature reduction rate, local minima could be escaped and global minimum would be reached. Simulated annealing can be used to optimize any criterion, for instance a likelihood (treating  $-L$  as the energy) or an information criterion (treating the criterion as the energy). In this chapter we employ it for clustering by optimization of information criteria to determine an optimal number of clusters and initial estimates of cluster parameters. These parameters are then refined by expectation maximization.

### **2.1.5. A review on mixture model-based clustering methods of mixed data types**

Clustering of multivariate data using a mixture of multivariate normals for continuous variable data or a mixture of multivariate Bernoulli densities for binary data as proposed by Wolfe (1970) and Everitt (1984) respectively are the essential technique known as latent class analysis [66]. There are several approaches to cluster data described by different types of variable or mixed data. Some possible approaches are to perform a separate clustering on each type of variables, convert all types of variables to a single type of variable followed by a clustering or clustering data set with both continuous variables and binary or ordinal values as proposed by Everitt (1988) [66]. The latter is important early work on model-based clustering of mixed mode data. Here, the authors proposed that the binary and ordinal variables come from an underlying continuous distribution of not-directly observable continuous variables. The method involves estimating the parameters of the unobservable continuous data by setting certain threshold values as the cut-off points.

A similar attempt was by Morlini [67], where author proposed a model-based clustering approach based on a multivariate Gaussian mixture model in clustering binary and continuous variables using a mixture of discrete (multinomial) and continuous (multivariate) distributions. With the assumption that the observed binary values '0' and '1' correspond to small and large latent continuous values/scores respectively, the author estimates the scores of the latent continuous variables which produces the observed binary values and then, combines this together with the scores of the original continuous variables for clustering. This involved deciding on the thresholds for each binary variable.

These ideas were extended to ordinal and nominal variables by McParland and Gormley [68] in their clustering method called ClustMD. They proposed a method using a latent variable model with underlying mixture Gaussian distributions to estimate the mixed type observed data of any combination of continuous, binary, ordinal or nominal variables.

The latent continuous data underlying both the ordinal and nominal data were assumed to be Gaussian, as were any observed continuous data. Thus, the joint vector of observed and latent continuous data,  $z_i$  was assumed to follow a multivariate Gaussian distribution,  $z_i \sim MVN_p(\mu, \Sigma)$ .

In all of the approaches mentioned above, an expectation maximization framework was adopted in estimating the maximum likelihood of the observed data which requiring manual specification of cluster numbers, and methods were applied to problems with relatively small numbers of variables (<10 continuous and <20 other). Similar ideas have been explored by Cai and co-workers in a Bayesian context [69]. Here, a generalized latent variable model was proposed with cumulative probabilities of various types of observed variables specified by a linear model of latent variables. Different density functions are applied for different types of data (i.e. Gaussian for continuous variable, Gaussian with thresholds for ordinal variable, Poisson or Binomial for count data, and multinomial logit link for nominal variable). Although this method can simultaneously model multiple data types, it is again dependent of the defined number of mixture components or prior Bayesian estimates and fairly large sample size is required to obtain accurate results.

Alternatively, addressing problems with large numbers of variables of different types and incorporating dimension reduction as an integral component, iCluster [70] and integrative phenotyping framework (iPF) [71] were developed specifically for integrating and clustering mixed genome-scale ('omics') data for disease subtype discovery. In iCluster, the link between data types was achieved by assuming a shared underlying latent variable model representing the disease subtypes. It also utilizes the k-means procedure to find the actual cluster assignments given latent variable values. iPF is a workflow developed to integrate independent homogenous clustering from different omics data in an agglomerative manner. It utilizes a dissimilarity matrix of features from clusters across omics data. This then followed by visualization of heterogeneous clustering of pairwise omics sources.

All of the approaches mentioned above assume a common clustering with a known/common set of clusters across all data types. A different approach was taken by the Bayesian MDI package [72, 73], which first cluster data sets based on pairwise relations (linking coefficients) between data sets, and then fusing entities together if they have same linking coefficients. MDI combines the *entities* into statistically distinct clusters while exploiting any latent structure in cluster allocations across data sets. A flexible Bayesian mixture modelling approach was applied. Although MDI provides adequate flexibility for grouping of fused entities, it does not clearly encourage any

sharing of clusters across more than pairs of data sets. Thus, we think our work would fill this gap by making an algorithm that is truly flexible in term of entities cluster allocations while exploiting all data sets of different types simultaneously.

## 2.2. Methodology

In this work, our intention was to use mixture-model clustering to cluster genes using both gene expression (continuous) and regulatory (binary) information (e.g. TF binding), and also to classify cancer patients into sub-classes based on gene expression (continuous) and mutation patterns (binary). The probabilistic model which will be explained below is in relation to the first one, where clustering expression of, and TF binding to genes is used to infer a relationship between them. This probabilistic model of genetic regulation for genes works the same way for clustering cancer patient data. In our genetic regulation framework, we refer to a set of genes that are regulated by a set of TFs as a regulatory module. Similarly, for the cancer patient classification, we refer a set of patients having different patterns of markers (i.e. gene expression and mutation) as a cluster.

### 2.2.1. The model

We consider a set of data-points (e.g. genes, tumour samples, etc.) each having a set of binary/regulatory inputs  $\{r_{ij}; j = 1, \dots, n_r\}$  where  $r_i \in \{0,1\}$  and a set of continuous/expression values  $\{e_{il}; l = 1, \dots, n_e\}$  (we will investigate both un-normalized and expression values normalized to zero mean and unit standard deviation for each data-point). There are therefore  $n_r$  regulatory inputs and  $n_e$  expression values. Within each module/cluster,  $m$ , we assume that the regulatory inputs are characterized by a set of probabilities  $\{p_{mj}; j = 1, \dots, n_r\}$  of the  $j^{th}$  regulatory input being equal to one for a data-point within this cluster. It is further assumed that the expression values follow normal distributions with means,  $\mu_{ml}$  and standard deviations,  $\sigma_{ml}$ . This leads to the following probabilistic model of genetic regulation for gene  $i$ .

$$p(r_{i1}, \dots, r_{in_r}, e_{i1}, \dots, e_{in_e}) = \sum_{m=1}^{N_m} \alpha_m \prod_{j=1}^{n_r} B(r_{ij}; p_{mj}) \prod_{l=1}^{n_e} N(e_{il}; \mu_{ml}, \sigma_{ml}) \dots \dots \dots \text{Equation 10}$$

Here  $B$  denotes the Bernoulli distribution with parameter  $p_{mj}$ , and  $N$  is the normal distribution with mean and standard deviation  $\mu_{ml}$  and  $\sigma_{ml}$ . An expression pattern is thus represented by a mean and standard deviation at each point the sequence. The  $\alpha_m$  are mixing coefficients where  $\sum_{m=1}^N \alpha_m = 1$  and  $N_m$  is the number of clusters.



Following this we use the log-likelihood

$$\ln L = \ln(\prod_{i=1}^{N_g} p(r_i, e_i)) = \sum_{i=1}^{N_g} \ln p(r_i, e_i) \dots\dots\dots \text{Equation 11}$$

where  $N_g$  is the number of genes in cluster  $m$  and we adopted vector notation for the regulatory inputs and expression levels for gene  $i$  for brevity.

### 2.2.2. Estimating model parameters

A standard approach to estimating the parameters in the model above would be to fix the number of mixture components or clusters,  $N_m$  and then fit the parameters of the mixture distribution by expectation maximization (EM). This process could be repeated for a selection of possible values of  $m$  and an optimum chosen, but this procedure is difficult for a number of reasons. First we have no information from the application domain about the likely value of  $N_m$ , and second the EM optimization algorithm is local in nature and therefore needs to be started from a number of different initial points to investigate possible alternative minima. Therefore we have adopted an alternative approach, beginning with a meta-heuristic search over models with  $1 < m < N_m$ , and finally using EM to refine the best model found. The initial heuristic approach makes the assumption that the components in the mixture model above (equation 9 and 10) from which it is assumed the data are generated, are 'well separated', so that the contribution of each gene (data point) to the likelihood is dominated by a single mixture component. With this assumption, given a solution comprising  $m$  clusters and the assignment of genes to clusters (let the set of genes assigned to cluster  $m$  be  $G_m$ ), the estimates of the maximum likelihood parameters for the component Bernoulli distributions are

$$P_{mj} = \frac{1}{|G_m|} \sum_{g_i \in G_m} r_{ij} \dots\dots\dots \text{Equation 12}$$

and the estimates of the parameters of the normal distributions are

$$\mu_{ml} = \frac{1}{|G_m|} \sum_{g_i \in G_m} e_{il} \dots\dots\dots \text{Equation 13}$$

$$\sigma^2_{ml} = \frac{1}{|G_m|} \sum_{g_i \in G_m} (e_{il} - \mu_{ml})^2 \dots\dots\dots \text{Equation 14}$$

and the estimates of the mixing coefficients are

$$\alpha_m = |G_m|/N_g \dots\dots\dots \text{Equation 15}$$

### 2.2.2.1. Model selection using simulated annealing

We can then seek to find mixture components parameters that would maximize the likelihood by a suitable search algorithm over assignments of data points to clusters. Without penalizing the extra number of parameters, ML maximizing the likelihood by increasing number of parameters. Number of parameters is directly proportional to the number of clusters, thus, this would result in each data point assigned to its own cluster. Penalizing this, discourages the over-estimation of parameter numbers. We have used information criterion (IC) based models optimization where extra parameters will be penalized.

We have compared different model selection criteria in selecting the best approximating model as there is no single best criterion for every underlying data structure. Moreover, as the number of data points,  $N_g$  increases, different information criteria imply different trade-offs between the goodness of fit and model complexity. In reference to the Equation 5, in our method,  $k = N_m (1 + n_r + 2N_e) - 1$ , accounting for  $N_m - 1$  independent mixing coefficients, and the Bernoulli parameters and Normal distribution parameters in each model.

Simulated annealing algorithm with Monte-Carlo moves are used to cluster simulated data by optimizing the objective function and different objective functions are tested on different level of data complexity (see Table 2.1 in the section 2.1.3). The representation/pseudo-code of the SA algorithm is as follows:

<b>Monte Carlo simulated annealing for clusters optimization</b>	
1:	Normalize expression (genes)*
2:	Clusters = Initialize clusters (agglomerative/divisive)
3:	Best clusters = [list]
4:	Old clusters = [list]
5:	Score = 0.0
6:	Old score = 0.0
7:	Best score = 0.0
8:	Difference = 0.0
9:	Count temperature = 0
10:	Temperature = Start temperature
11:	While Count temperature < Maximum temperature cycle:
12:	Temperature = Temperature * Temperature decreasing factor
13:	Score = Calculate modules score (clusters)
14:	beta = 1.0/temp
15:	Old clusters = clusters
16:	While Iteration < Maximum iterations:
17:	If random > Merge and split probability:
18:	Change (clusters)
19:	Else:
20:	If random > 0.5:
21:	Merge (clusters)
22:	Else: Split (clusters)
23:	Old score = Score
24:	Score = Calculate clusters score (Clusters)
25:	Difference = (Score-Old score)
26:	If Difference < 0.0 or random < exponent(-beta*Difference):
27:	'accept'
28:	if Score < Best score:
29:	Best score = Score
30:	Best clusters = Clusters
31:	Else:
32:	'reject'
33:	Clusters = Old clusters
34:	Score = Old score
35:	End while
36:	Count temperature + 1
37:	If Best score = Old Score:
38:	Count score = Count score + 1
39:	If Count score == MaxReplters:
40:	break
41:	End while
42:	Return Best score, Best clusters

**Algorithm 1**

Pseudo-code for the Monte-Carlo Simulated annealing algorithm.

The first step of the algorithm is to read and store the local copies binary inputs and continuous inputs. Then, if normalization of continuous inputs is chosen, it will be normalized to zero mean and one standard deviation also known as z-score (see line 1 in Algorithm 1). This is then followed by initialization/starting point of clustering. We represent a clustering solution as an integer array of length  $N_g$  where each element of the array holds the 'cluster number' to which the corresponding gene is assigned. Cluster numbers are thus integers in the range 1 to  $N_g$ . Depending on the initialization schedule either agglomerative or divisive, each data point will be given a different cluster number for agglomerative and one cluster number for all data points for divisive schedule. Below is a representation of the clusters initialization for 10 genes:

Divisive: 

1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---

Agglomerative: 

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

The initial clustering solution (see line 2 in Algorithm 1) are stored in an array as shown above (e.g. 10 data points). Next, we initialized arrays for storing the solutions (see lines 3 and 4) and values of 0.0 for scores and scores difference (lines 5 to 8). We then initialized the temperature for the simulation system as in lines 9 and 10.

The main method or loop for simulated annealing is starting from line 11 onwards. While count temperature less than the maximum number of temperatures to be simulated (Maximum temperature), this program will run Monte-Carlo moves (lines 17 to 22) and acceptance probability (line 26) evaluation as in Equation 6. Furthermore, with different options of Monte-Carlo moves available, change, merge and split cluster's member, we decided on arbitrary threshold of probability of either one of them. Here how it works:

If a random uniformly distributed number,  $r$  is greater than the specified probability of splitting (MergeSplitProbability is 0.25 as in Table 2.2), than any random data point in a cluster will be moved to another cluster. If  $r$  is more than 0.5, two random clusters are merged into one cluster or else, a random cluster will be split into two new clusters. For every iteration of Monte-Carlo moves (line 16), a score will be calculated and accepted as the best score and clusters corresponding to this score will be accepted as the best clusters. These steps in Maximum temperature loop will run as long as the Maximum temperature has not been reached and the repetition of the same best score is less than MaxRepters (line 40).

For each subsequent solution following Monte Carlo randomization moves and simulated annealing, two genes are in the same cluster if and only if they have the same cluster

number. There is no requirement that all cluster numbers be used in a solution, for example, the following instance of a clustering solution for 10 genes

Solution: 

1	1	1	3	4	4	4	3	4	1
---	---	---	---	---	---	---	---	---	---

represents a cluster solution with 3 clusters (labelled 1,3,4) where cluster 1 is genes 1,2,3 and 10, cluster 3 is genes 4 and 8, and cluster 4 is genes 5,6,7 and 9. Clearly this solution could also be represented using cluster numbers 1, 2, 3 as

Solution: 

1	1	1	2	3	3	3	2	3	1
---	---	---	---	---	---	---	---	---	---

but for algorithmic reasons it is easier to allow both representations to be used in the search procedure, only renumbering on completion.

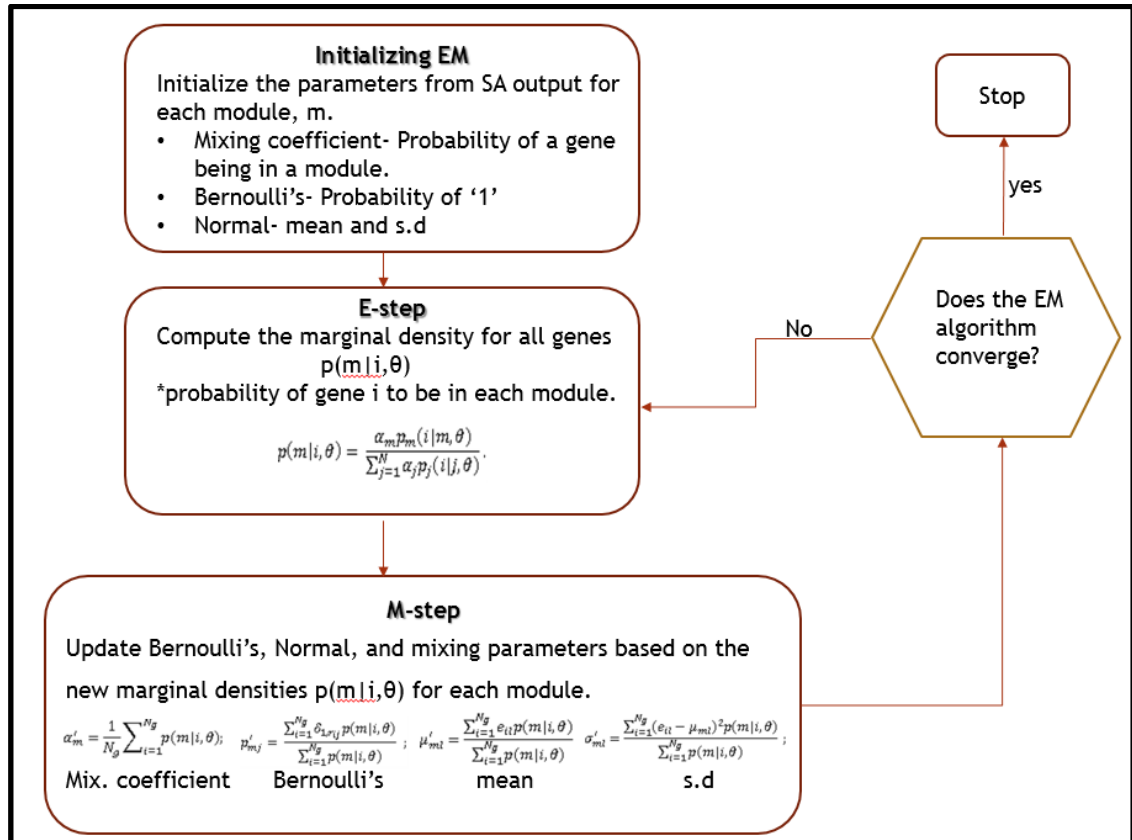
The configuration of clusters and its parameters will be sent to the EM to be further optimized locally.

**2.2.2.2. Model refinement using expectation maximization**

The mixture density parameters estimation problem can be solved globally using SA. However, the mixture density parameters estimated by SA could still be improved by local refinement. EM is well-known for finding ML densities parameter estimation and this would help in refining the clusters found by SA. With an assumption that the ML models found by SA are incomplete, and given the current vector of parameters for all clusters,  $\theta = (\alpha_1, \dots, \alpha_m, p_{1j}, \dots, p_{mj}, \mu_{1l}, \dots, \mu_{ml}, \sigma_{1l}, \dots, \sigma_{ml})$ , we can easily calculate the probability for gene  $i$  to be in for mixture component  $m$  defined in 2.2.1 by using an equation derived from Bayes' rule:

$$p(i \in m | \theta) = p(m | i, \theta) = \frac{\alpha_m p_m(i | m, \theta)}{\sum_{j=1}^M \alpha_j p_j(i | j, \theta)} \dots \dots \dots \text{Equation 16}$$

Equation 16 is for estimating the degree of mixing between clusters through the probability density that data point  $i$  is generated from mixture component  $m$  [74]. Mixing coefficients  $\alpha_m$  can be interpreted as prior probabilities for membership of each module. The steps of the EM algorithm are showed in the Figure 2.5 below:



**Figure 2.5** Refinement of model parameters using EM.

It starts with the prior probability densities from the SA output and refinement of its parameters until convergence. Convergence here means, until the parameters values do not change for two consecutive EM iterations or until the maximum number of iterations has been reached.

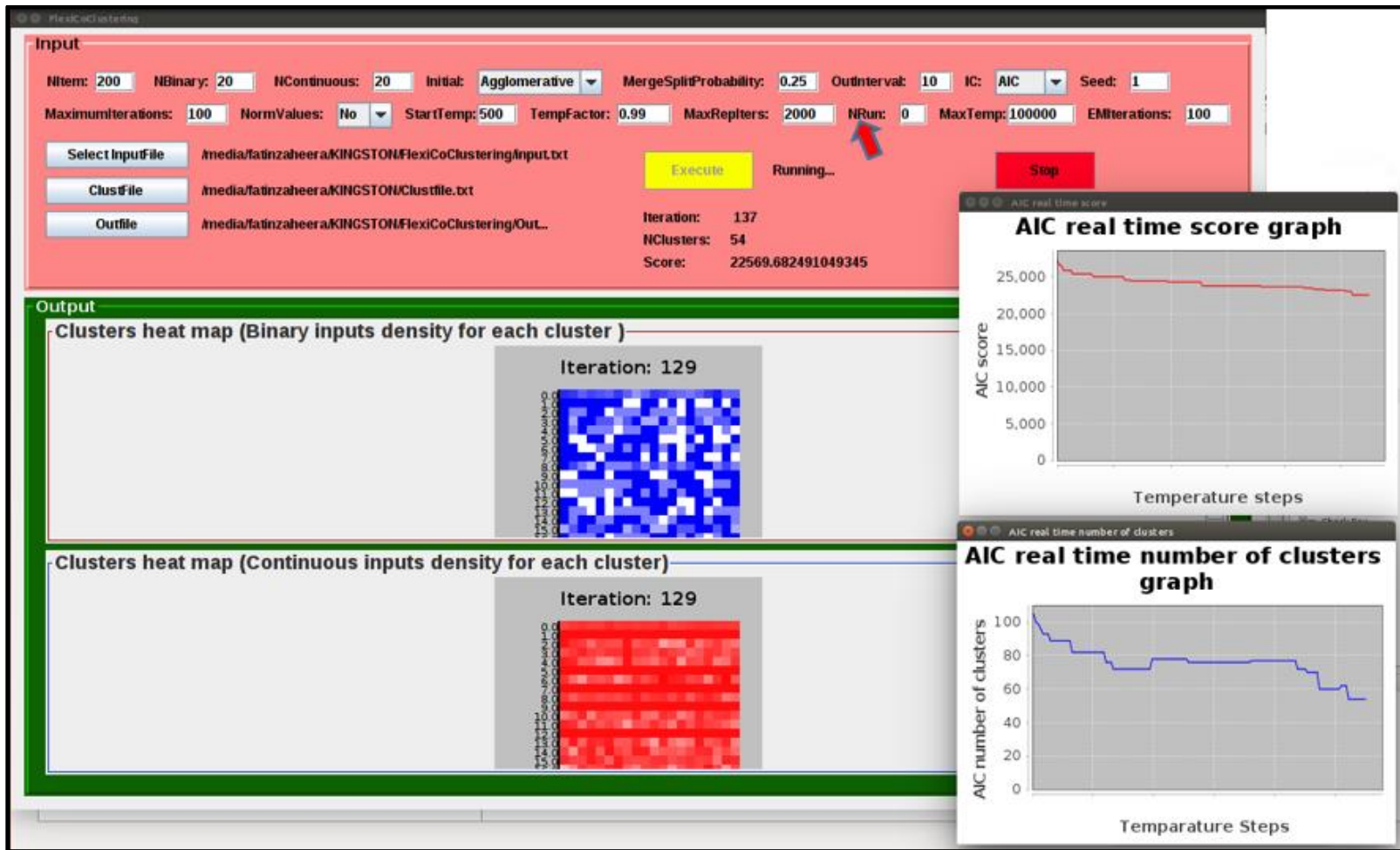
### **2.2.2.3. FlexiCoClustering: A package for model-based co-clustering of binary and continuous inputs**

With a method for model-based joint clustering of binary and continuous inputs in hand, we packaged this algorithm which we named as 'FlexiCoClustering' into both, a command line interphase and also a Graphical User Interphase (GUI) which have been made freely available at a public data repository in GitHub. The link to both packages is:

<https://github.com/BioToolsLeeds/FlexiCoClusteringPackage/>

For a GUI, we utilized the existing Java Graphics APIs called Swing which provides a huge set of reusable Graphical User Interphase (GUI) components, such as button, text field, label, choice, panel, and frame for building a user friendly GUI application. With these components which are the built-in components in NetBeans Interactive Development Environment (IDE) for Java [75], the process of building of this GUI application was straight forward.

Figure 2.6 below shows a snapshot of the GUI application with the real-time probability densities for binary and continuous input parameters of each cluster. Alongside these, this GUI produced a real-time score and number of clusters as well. For details on how this application works and types of output produced, please refer to the software user manual in Appendix A. A full list of algorithm parameters is given in Table 2.2. Suitable values for these parameters were determined by examining algorithm performance on simulated data.



**Figure 2.6** A snapshot of the FlexiCoClustering GUI upon running the application.

A snapshot of the FlexiCoClustering GUI upon running the application. Red arrow shows where user should change the NRun to '1' after the initial run (NRun=0) have finished if it is required at all.



**Table 2.2** A list of runtime parameters that were simulated in optimizing the model.

Values in bold are the default parameters setting. Runtime parameters in red could be changed from default values (in bold) to the user specified values depending upon datasets which will be used as input. Parameters with '\*' are with default values and optional to change.

Runtime parameter	Functional description
<b>NItems</b>	Number of data points
<b>NBinary</b>	Number of binary variables in the input files
<b>NContinuous</b>	Number of continuous variables in the input files
Agglomerative/Divisive	Starting point option for clustering, if agglomerative start with all data points in separate clusters, if divisive start with all in a single cluster.
<b>IC (AIC/AIC<math>\lambda</math> /BIC/HQC/CAIC)</b>	Scoring function/information criterion for the model selection (see section 2.2.2.1 for details).
MergeSplitProbability ( <b>0.25</b> )	There are 3 possible Monte Carlo moves, chosen according to the following scheme. A standard move (swapping a single entity into another cluster) occurs with probability 1-MergeSplitProbability. Otherwise either merging two clusters into one, or splitting one cluster into 2 are chosen with equal probability.
MaximumIterations ( <b>500</b> )	The maximum number iterations of the temperature loop, i.e. the maximum number of different temperatures (controls the lowest temperature used)
StartTemp ( <b>500</b> )	Starting temperature for simulated annealing. Fixed by experimentation to give a high move acceptance ratio.
TempFactor ( <b>0.999</b> )	Factor by which the temperature is reduced at each iteration of the temperature loop. Set to 0.999 by default (used in all optimizations reported in the paper).
<b>MaxTempCycle (100,000)</b>	Maximum number of temperature cycle (termination criterion)
<b>MaxRepters (2000)</b>	Maximum number of best score repetition (convergence criterion)

**Table 2.2** A list of runtime parameters that were simulated in optimizing the model.

Values in bold are the default parameters setting. Runtime parameters in red could be changed from default values (in bold) to the user specified values depending upon datasets which will be used as input. Parameters with '\*' are with default values and optional to change. **(Continued)**

Runtime parameter	Functional description
Seed ( <b>1</b> )	The seed of the random number generator used to produce permutations
EMIterations ( <b>100</b> )	The maximum number of EM iterations if EM parameters values do not converged
OutInterval	Intervals at which the solution is printed in the output and at which the heat maps are updated on GUI
ClustFile	The name of the final clusters output file
NormExp ( <b>1/0</b> )	Normalise continuous inputs to zero mean and a standard deviation for each data points- z-scores.
Nrun ( <b>0</b> /any integer)	Number of re-run of the program after initial run. The final clusters found by SA will be use as a new starting point if user decided to increase the simulated annealing run

### 2.2.3. Testing the algorithm on simulated data

As an initial test of the methodology and objective functions, we examined their ability to find correct solutions for the number of mixture components and the assignment of data points to components in data simulated from the probability distribution explained previously. Both, binary and continuous data sets were simulated using following steps:

1. The test case data sets were specified as 100, 200, 500 and 1000 number of data points with two sub sets each containing 10 and 20 clusters (e.g. the 500 data point sets were 25 clusters of size 20, and 50 clusters of size 10). We have used  $n_r = 20$  for binary inputs and  $n_e = 20$  for continuous values for all data sets.
2. The input file for simulated data generator is in the format of:

NItems : 200	
NBinary : 20	
NContinuous : 20	
NClusters : 20	
0.9 0.4 0.4 0.4 0.4	: 7.9 0.3 2.1 0.3 4.5 0.3 9.8 0.3 -8.0 0.3
0.4 0.4 0.4 0.4 0.9	: -2.9 0.3 8.1 0.3 5.5 0.3 9.7 0.3 -8.0 0.3
0.9 0.4 0.4 0.4 0.9	: 9.9 0.3 -3.1 0.3 8.5 0.3 7.3 0.3 -1.0 0.3

Here, NDataPoints is the number of data points ( $n_r$ ), RegulatoryInputs is  $n_r$ , ExpressionValues is  $n_e$  and NClusters is  $n_m$ . The rest of the lines in the input file are the parameters needed to produce clusters where each line corresponds to binary inputs (in dark blue),  $p_{mj}$  and continuous expression patterns (on the right hand side after colon ':') for each cluster to be simulated. For continuous inputs, values in black are the mean,  $\mu_{ml}$  and in red are the  $\sigma_{ml}$ .

3. Binary inputs from the Bernoulli distribution for each member of clusters were generated. Each cluster was specified with a distinct patterns of binary variables  $p_{mj}$ . To simulate the noise that exists in a real data set, the simulated data generator was seeded with two different combinations of Bernoulli parameters. The tight/less noisy simulated data sets were modelled using combination of 0.1 and 0.9. On the other hand, noisy data sets were generated with combination of 0.4 and 0.9 in the initial input vector.
4. Continuous values from Gaussian distribution were generated for each continuous variable in each cluster. Again, to simulate the noise effects, we used two different Gaussian parameters. The tight/less noisy simulated data sets are with standard deviations of 0.01 and noisy data sets are with standard deviations of 0.3. The continuous expression values for each cluster were generated randomly using uniform distribution between -15.0 and 15.0.

5. Appropriate measures were taken to make sure that continuous and binary patterns are unique between clusters and random numbers from appropriate probability distributions were generated using standard functions in the Java programming language.

#### 2.2.4. Summary of test data sets

Different data sets were simulated from using the simulated data generator with a range of values for the various parameters and were used to test the search procedure and its ability to find the optimal model. Initially, we wanted to test the efficiency of SA and objective functions with an increasing number of data points. Table 2.3 below shows sets of simulated data used for studying the effects of increasing number of number of data points where the number of data points per cluster was set to be 10 and 20 for each data set.

Data set	1	2	3	4	5	6	7	8
Number of clusters	5	10	10	20	25	50	50	100
Number of data points	100	100	200	200	500	500	1000	1000
Number of data points per cluster	20	10	20	10	20	10	20	10

**Table 2.3** Eight sets of simulated data with increasing number of genes and genes per module.

## **2.3. Results**

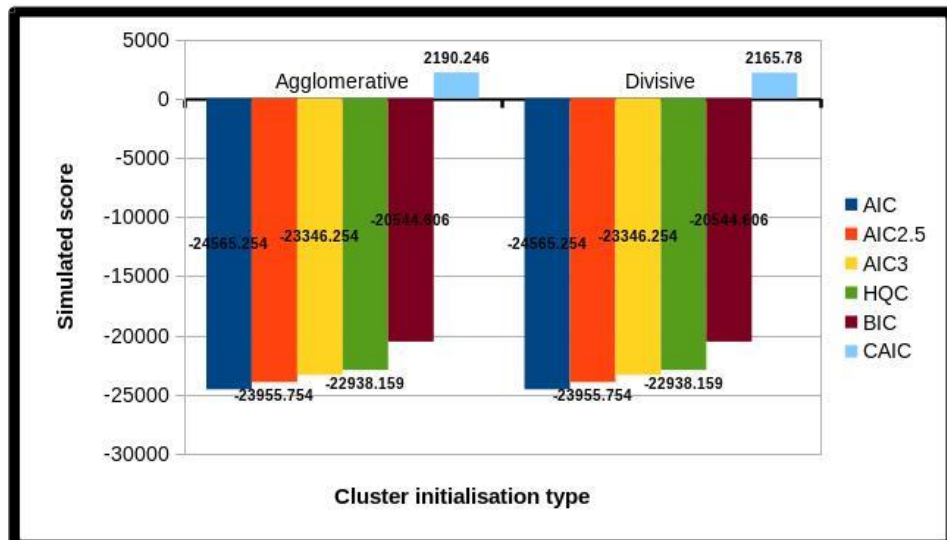
We investigated the ability of our algorithm to find probable solutions using simulated data. The simulated data consist of 100-1000 data points and 5-100 mixture components as shown in Table 2.3 above.

### **2.3.1. Optimization of runtime parameters**

We determined the best way to run our algorithm before applying it to the real dataset. Three main algorithm runtime parameters, namely, Agglomerative or Divisive, Starting and Maximum Temperature, and MaxRepters were optimised and the results are shown accordingly in the following sub-sections.

#### **Agglomerative/Divisive**

As our clustering is purely heuristic, random moves also known as Monte-Carlo moves such as splitting, shuffling and combining cluster members (data points) resulted in increasing and decreasing number of clusters as the algorithm runs. We are not sure if the starting points of clustering (i.e. agglomerative or divisive) would make any difference in the final solutions as it is important in other standard clustering methods that were mentioned in the introductory chapter. Based on Figure 2.7 below, agglomerative and divisive mode of clustering, starting points are not significantly differed from each other in term of scores produced as well as number of clusters found. The difference between them is the initialization of the clusters where agglomerative mode assigns every data point to its own cluster and divisive mode put all data points into a single cluster. This tells us that either agglomerative or divisive can be used with the clustering algorithm where both can find the true solutions independent of the starting points.



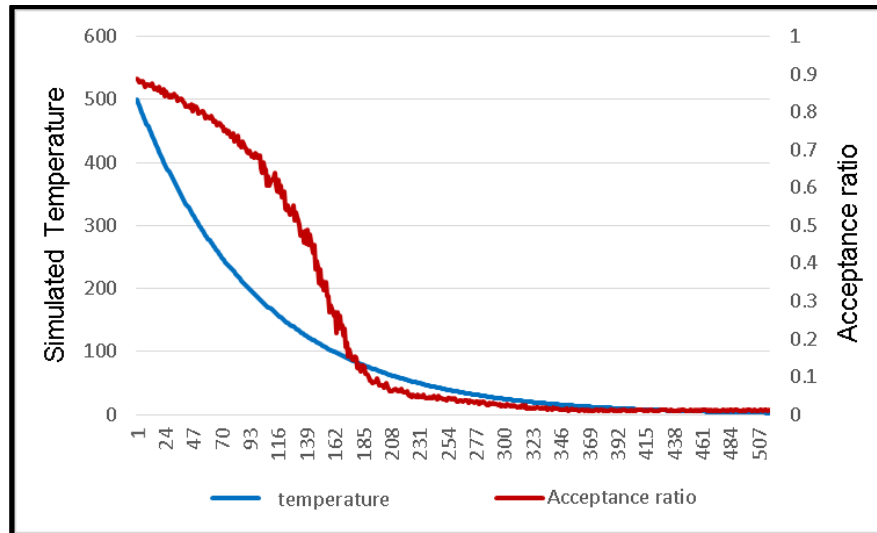
**Figure 2.7** Comparison of scores between different starting points.

The simulated scores for all relevant objective functions from starting the algorithm agglomeratively and divisively. A relevant set of runtime parameters used to produce this result: StartTemp=500; TempFactor=0.999; MaxTempCycle=100,000.

### Starting Temperature and Maximum Temperature

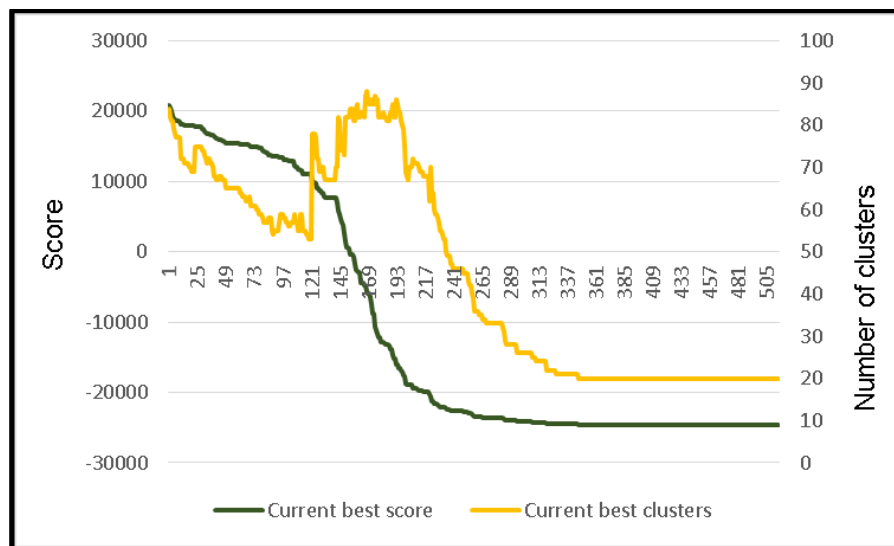
As mentioned in the methodology section, start temperature (StartTemp) and maximum temperature cycle (MaxTempCycle) are changeable parameters, and therefore we tried few starting temperatures at the beginning of simulation study (i.e. 150 and 500). Maximum temperature cycle on the other hand was arbitrarily decided to be as high as possible (100,000) because it is just a secondary convergence criterion/termination criterion where if the Maximum number of best score repetition (MaxRepters) have not been met after a long time, the algorithm will be terminated automatically. Although using both 150 and 500 as the starting temperature gave the same results, the times taken are slightly different. Higher starting temperature took longer time to converge but it did converge in the end, whereas with lower Starting temperature, it converged faster but with higher probability of false positive or converged to a local minimal (see Figure 2.4 for illustration).

Higher starting point takes a longer time because it explores more solutions at much higher temperatures in the beginning of the run. In our case, where we used a really high maximum temperature cycle or maximum number of temperatures to be simulated, this does help to overcome the problem observed at lower starting temperature. In Figure 2.8 below, we can clearly see that acceptance ratio-percentage of acceptance of either random solution or correct solution are high at the beginning of the run where the temperature of the system is high. As the system temperature reduces, the acceptance ratio also decreases and eventually converged to a correct solution (see Figure 2.9).



**Figure 2.8** The real-time simulated temperature and acceptance ratio from clustering using AIC2.0.

The real-time simulated temperature and acceptance ratio from clustering using AIC2.0. A relevant set of runtime parameters:  $N_g=200$ ;  $n_r=20$ ;  $n_e=20$ ;  $N_m=20$ ; StartTemp=500; TempFactor=0.999; MaxTempCycle=100,000.



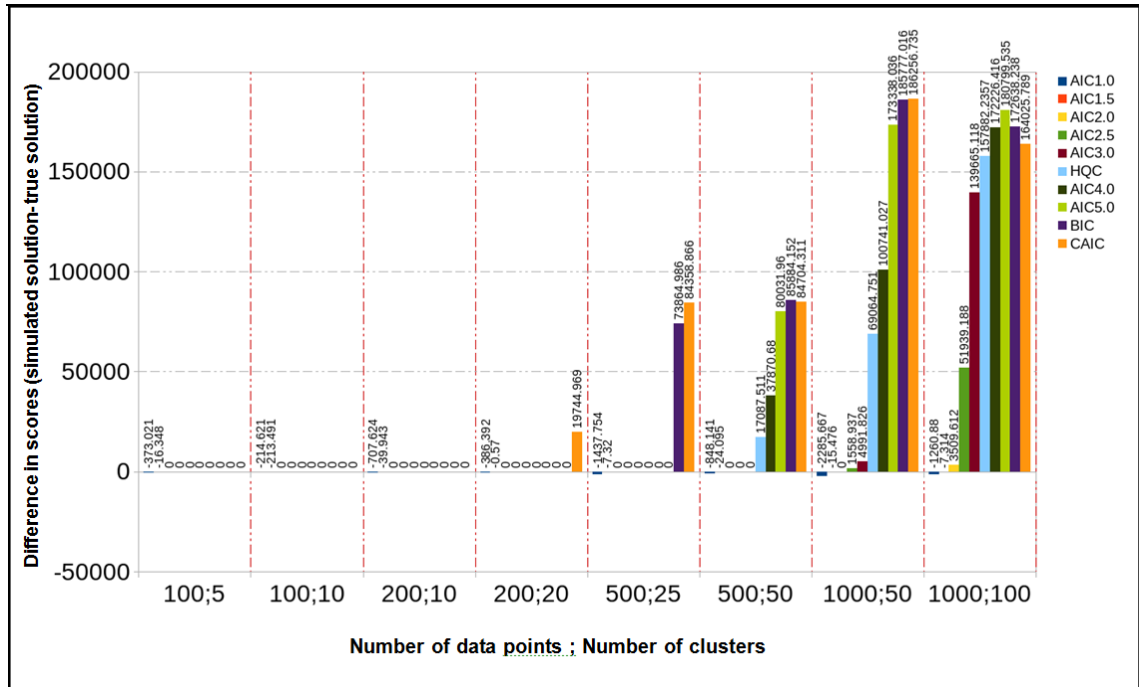
**Figure 2.9** The real time simulated score and number of clusters from clustering using AIC2.0.

The real time simulated score and number of clusters from clustering using AIC2.0. A relevant set of runtime parameters:  $N_g=200$ ;  $n_r=20$ ;  $n_e=20$ ;  $N_m=20$ ; StartTemp=500; TempFactor=0.999; MaxTempCycle=100,000.

As this is a meta-heuristic approach, the number of clusters are fluctuating during the first half of the simulation resulted from the acceptance criterion imposed as previously explained from Equation 6. Scores on the other hand, always decreased with time and temperature. Here we could say that our simulated annealing is working and have been able to produce correct results for the chosen simulated dataset.

### 2.3.2. Running simulation on simulated data

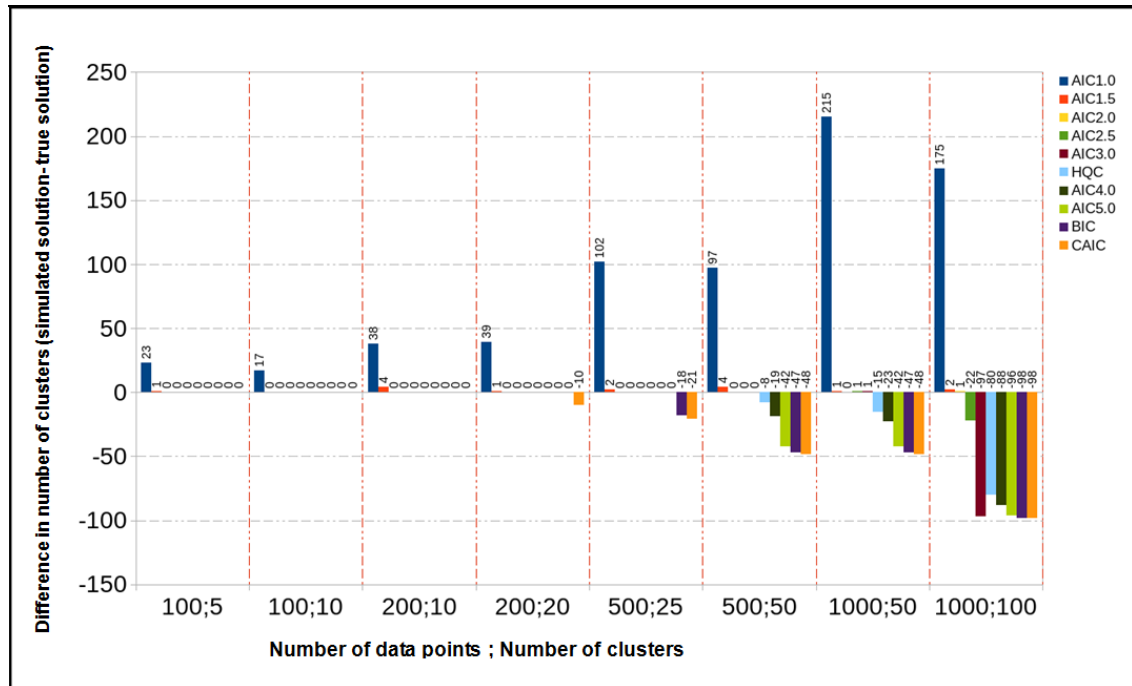
Data were simulated for both low and high level of variability and results are presented in (Figure 2.10 and Figure 2.11) and (Figure 2.12 and Figure 2.13), respectively.



**Figure 2.10** Difference in scores result from our algorithm using a tightly clustered data and relatively little ‘noise’ data set simulated from the probability distribution assumed in our method.

Result from our algorithm using a data set simulated from the probability distribution assumed in the method for  $n_r = 20$  regulatory inputs and  $n_e = 20$  expression values. In this case parameters of the simulation correspond to tightly clustered data and relatively little ‘noise’ (Bernoulli parameters of 0.1 or 0.9 at each regulatory input and expression standard deviations of 0.01). The cases simulated covering 100-1000 data points and 10 or 20 data points per cluster in each case. This figure shows the difference in score, between the solutions found by the algorithm and the known true solutions. Results are shown for several objective functions arranged in order of increasing penalty value,  $\lambda$ . Differences of zero in each case indicate that the algorithm found the true solution; negative score differences indicate objective function failures (solutions different to the true solution exist with better scores), and positive score differences indicate search algorithm failure (algorithm stopped at a solution scoring worse than the true solution). A relevant set of runtime parameters used to produce this result: StartTemp=500; TempFactor=0.999; MaxTempCycle=100,000.





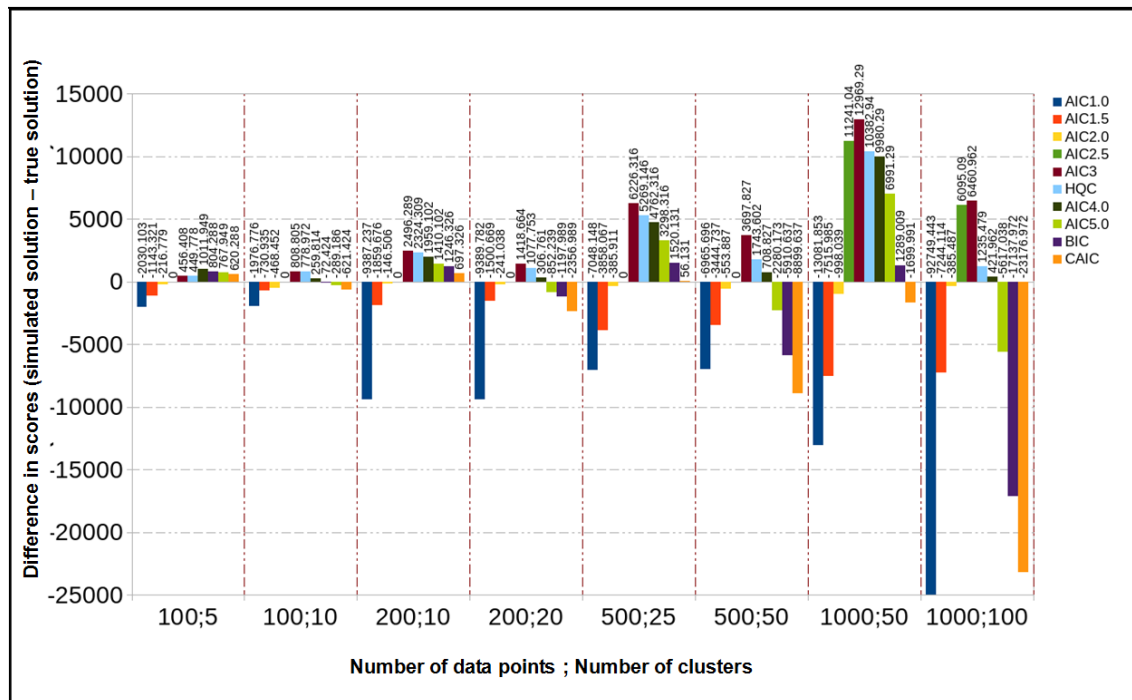
**Figure 2.11** Difference in number of clusters result from our algorithm using a tightly clustered data and relatively little ‘noise’ data set simulated from the probability distribution assumed in our method.

Result from our algorithm using a data set simulated from the probability distribution assumed in the method for  $n_r = 20$  regulatory inputs and  $n_e = 20$  expression values. In this case parameters of the simulation correspond to tightly clustered data and relatively little ‘noise’ (Bernoulli parameters of 0.1 or 0.9 at each regulatory input and expression standard deviations of 0.01). The cases simulated covering 100-1000 data points and 10 or 20 data points per cluster in each case. This figure shows the difference in the number of clusters between the solutions found by the algorithm and the known true solutions. Results are shown for several objective functions arranged in order of increasing penalty value,  $\lambda$ . Differences of zero in each case indicate that the algorithm found the true solution. A relevant set of runtime parameters used to produce this result: StartTemp=500; TempFactor=0.999; MaxTempCycle=100,000.

The results in Figure 2.10 and Figure 2.11 above show that for smaller numbers of data points (100-200), most of the objective functions with the exception of ( $\lambda = 1.0, 1.5$ ) successfully find the correct solution. Failure of  $\lambda = 1.0$  and 1.5 by finding solutions with more mixture components (clusters) with lower objective value (score) than the true solution score generated from the underlying distribution, indicate too low a penalty in the objective functions. With the increase in number of data points (500-1000), the optimization procedure found solutions very similar and equal to the correct solution for  $\lambda = 2.0$  and 2.5 respectively.

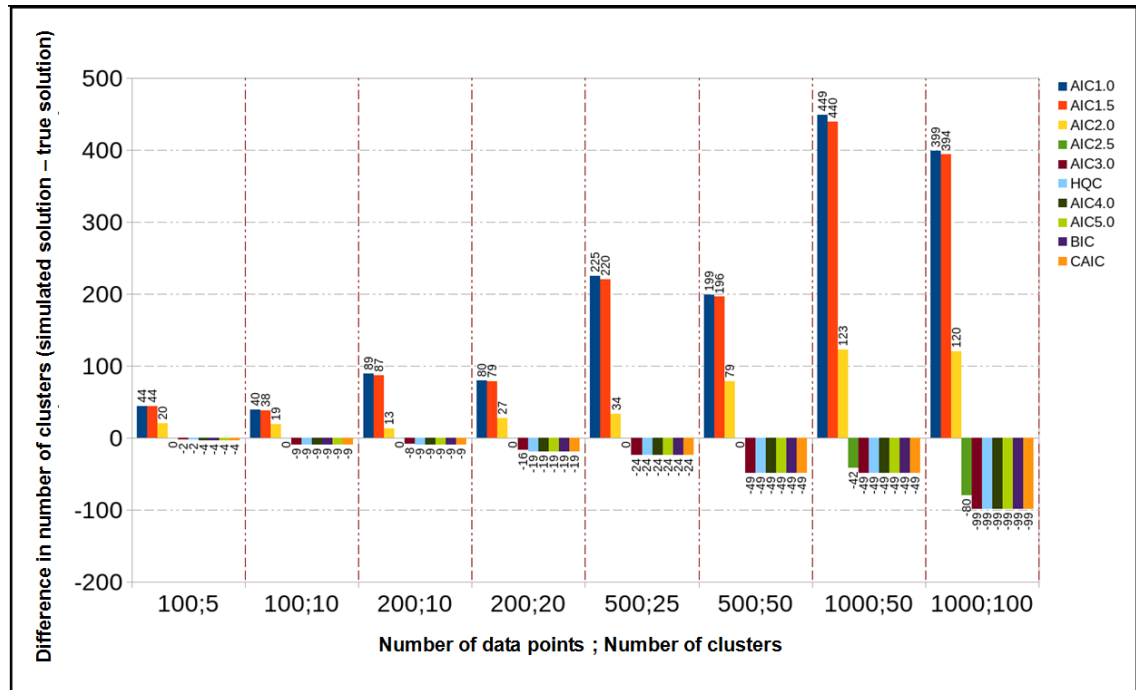
For larger numbers of data points, optimization algorithm with the stronger objective functions (AIC ( $\lambda = 3.0$ ) onwards, HQC, and BIC) have failed to find the correct solutions, generally finding alternatives with too few clusters. Further testing, by starting optimization algorithm at the correct solution in these cases, revealed the simulated solutions number of clusters are still smaller than the correct solution, indicating a failure

of the optimization algorithm rather than the objective function in these cases. By applying a more extensive annealing schedule with higher starting temperature, the solutions for these cases were not improved at all.



**Figure 2.12** Difference in scores result from our algorithm using a noisier data and less tight clusters data set simulated from the probability distribution assumed in our method.

Result from our algorithm using a data set simulated from the probability distribution assumed in the method for  $n_r = 20$  regulatory inputs and  $n_e = 20$  expression values. In this case parameters of the simulation correspond to noisier data and less tight clusters (Bernoulli parameters of 0.4 or 0.9 at each regulatory input and expression standard deviations of 0.3). The cases simulated covering 100-1000 data points and 10 or 20 data points per cluster in each case. This figure shows the difference in score, between the solutions found by the algorithm and the known true solutions. Results are shown for several objective functions arranged in order of increasing penalty value,  $\lambda$ . Differences of zero in each case indicate that the algorithm found the true solution; negative score differences indicate objective function failures (solutions different to the true solution exist with better scores), and positive score differences indicate search algorithm failure (algorithm stopped at a solution scoring worse than the true solution). A relevant set of runtime parameters used to produce this result: StartTemp=500; TempFactor=0.999; MaxTempCycle=100,000.



**Figure 2.13** Difference in number of clusters result from our algorithm using a noisier data and less tight clusters data set simulated from the probability distribution assumed in our method.

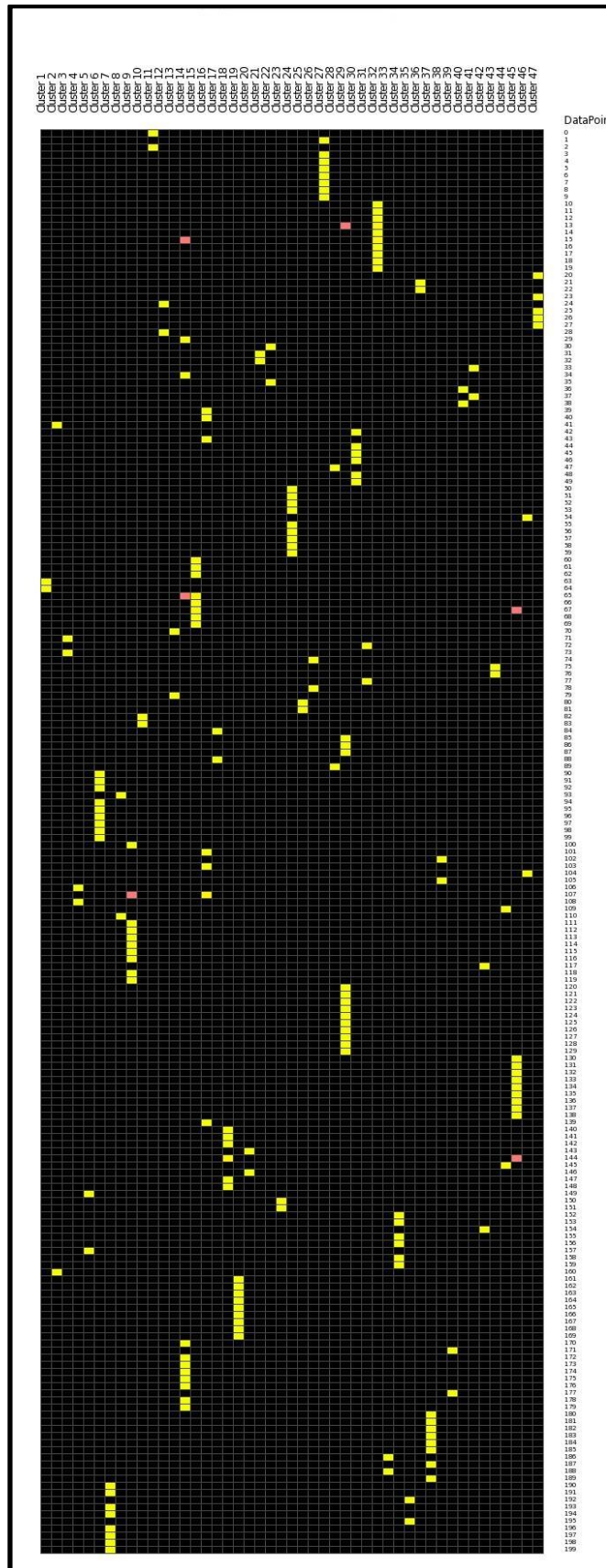
Result from our algorithm using a data set simulated from the probability distribution assumed in the method for  $n_r = 20$  regulatory inputs and  $n_e = 20$  expression values. In this case parameters of the simulation correspond to noisier data and less tight clusters (Bernoulli parameters of 0.4 or 0.9 at each regulatory input and expression standard deviations of 0.3). The cases simulated covering 100-1000 data points and 10 or 20 data points per cluster in each case. This figure shows the difference in the number of clusters between the solutions found by the algorithm and the known true solutions. Results are shown for several objective functions arranged in order of increasing penalty value,  $\lambda$ . Differences of zero in each case indicate that the algorithm found the true solution. A relevant set of runtime parameters used to produce this result: StartTemp=500; TempFactor=0.999; MaxTempCycle=100,000.

Using data simulated with higher variability within mixture components, similar observations which lead to the similar conclusions as with the lower variability data sets have been found (see Figure 2.12 and Figure 2.13 above). With  $\lambda = 2.0$  and 2.5, solutions found are equal or closer to the correct solutions for 100 to 500 data points and for higher number of data points (i.e. 1000 data points), standard AIC solutions are closer to the correct solution than  $\lambda = 2.5$ . However, for higher penalty criteria (i.e.  $\lambda = 4.0, 5.0$ , HQC, BIC and CAIC), they fail at the level of the objective function (solutions with lower scores as well as too few number of clusters than the correct solutions). Clusters found by using the standard AIC were actually clusters which are sub-clusters of AIC ( $\lambda = 2.5$ ) bigger clusters (see Figure 2.14 below) and EM refinement of standard AIC clusters (see following Figure 2.15) gave no improvement over the marginal density for current solution, thus suggesting that standard AIC allowing more number of mixing components parameters can be considered to be as optimum solution.

AIC 2.0 Cluster	AIC2.5 Cluster number of each member											
Cluster1	6	6										
Cluster2	4	16										
Cluster3	7	7										
Cluster4	10	10										
Cluster5	14	15										
Cluster6	9	9	9	9	9	9	9	9	9			
Cluster7	19	19	19	19	19	19	19	19				
Cluster8	9	11										
Cluster9	10	11	11	11	11	11	11	11	11			
Cluster10	8	8										
Cluster11	1	1										
Cluster12	2	2										
Cluster13	7	7										
Cluster14	2	3	17	17	17	17	17	17	17	17		
Cluster15	6	6	6	6	6	6	6	6	6			
Cluster16	3	4	4	10	10	10	13					
Cluster17	8	8										
Cluster18	14	14	14	14	14	14						
Cluster19	16	16	16	16	16	16	16	16	16			
Cluster20	14	14										
Cluster21	3	3										
Cluster22	3	3										
Cluster23	15	15										
Cluster24	5	5	5	5	5	5	5	5	5			
Cluster25	8	8										
Cluster26	7	7										
Cluster27	1	1	1	1	1	1	1	1				
Cluster28	4	8										
Cluster29	8	8	8	12	12	12	12	12	12	12	12	12
Cluster30	4	4	4	4	4	4						
Cluster31	7	7										
Cluster32	20	20	20	20	20	20	20	20	20	20		
Cluster33	18	18										
Cluster34	15	15	15	15	15	15						
Cluster35	19	19										
Cluster36	2	2										
Cluster37	18	18	18	18	18	18	18	18				
Cluster38	10	10										
Cluster39	17	17										
Cluster40	3	3										
Cluster41	3	3										
Cluster42	11	15										
Cluster43	7	7										
Cluster44	10	14										
Cluster45	13	13	13	13	13	13	13	13	13			
Cluster46	5	10										
Cluster47	2	2	2	2	2							

**Figure 2.14** Standard AIC ( $\lambda = 2.0$ ) clusters membership when compared with AIC ( $\lambda = 2.5$ ) clusters member.

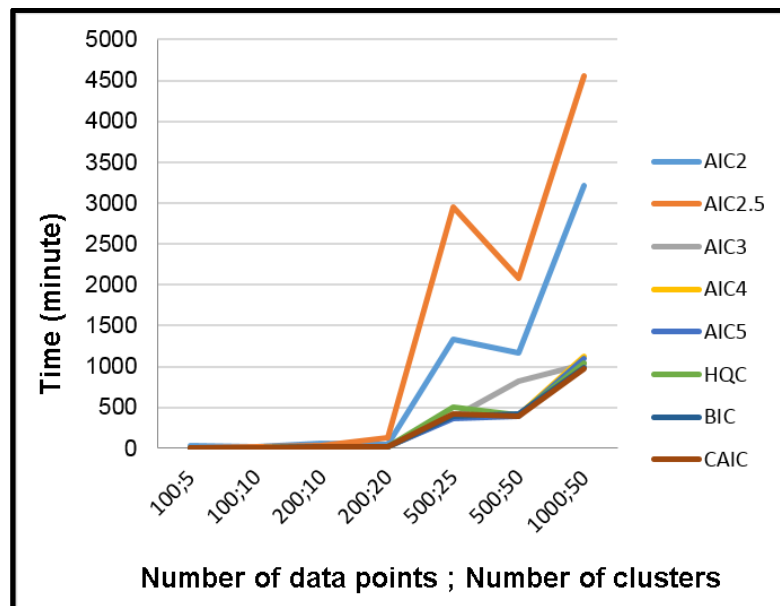
Numbers in AIC2.5 columns are the cluster number of AIC2.5 clusters members. With the exception of the cluster members highlighted in yellow, data points in standard AIC (200 data points and 20 clusters) are sub-clusters of AIC ( $\lambda = 2.5$ ) bigger clusters.



**Figure 2.15** Expectation maximization result.

Marginal densities  $p(m|i, \theta)$  for data point  $i$  being in each cluster  $m$  are shown for Standard AIC ( $\lambda = 2.0$ ). Colors: salmon (density approximately 0.0001), yellow (density approximately 1.0) and black (density approximately 0). Rows are the 200 data points and columns are the 47 clusters.

CPU time is another factor used to measure the performance of an algorithm and the result is shown in Figure 2.16 below. AIC ( $\lambda = 2$ ) and AIC ( $\lambda = 2.5$ ) perform similarly in terms of CPU times taken for small number of data points. However, a longer time needed by AIC ( $\lambda = 2.5$ ) to converge as the problem size gets bigger. In addition, with the exception of AIC ( $\lambda = 2$  and 2.5), the rest of the objective functions converged fairly quickly because their secondary convergence criterion (according to MaxReplters criteria of 2000 repetitions) has been met with solution of really small number of clusters ( $\sim 1$ ). AIC ( $\lambda = 2.5$ ) managed to find solutions equal or near to the correct solutions for handful of data complexity and variation, thus, needed a longer time to optimize its solution. Longer CPU time is needed as the data complexity increases due to the wider solutions space used in optimizing the algorithm.



**Figure 2.16** CPU times taken to run the simulation.

CPU times taken to run the simulation for several objective functions as a function of problem size (Number of data points; number of clusters) from using high variation data. A relevant set of runtime parameters used to produce this result: StartTemp=500; TempFactor=0.999; MaxTempCycle=100,000.

## 2.4. Discussion

We applied mixture model of Gaussian and Bernoulli distributions for continuous and binary variables respectively. As described previously, some published methods have used solely Gaussian distribution in describing the binary/categorical and continuous observations. Our method differed from the existing mixture model clustering where we have applied Bernoulli's distribution to represent the binary variables instead of using the latent variables approach where converting binary values to continuous values and applying the Gaussian distribution to model both continuous variables and continuous latent variable is deemed to be a better approach.

Some previous studies applied different modelling approach whereby nominal, binary and ordinal variables are represented as separate distributions (mostly derived from Gaussian distribution) and a mixture of these models which makes up the likelihood function. However, as more models are included, the computational cost will increase exponentially with the increase in model parameters. Hence, this will decrease the efficiency of the method. As a matter of simplicity and generality, nominal including binary variable and ordinal variable can be converted and represented by creating a separate binary variable for each nominal variable label and ordinal variable level. Furthermore, most existing approaches have used fixed cluster numbers followed by EM and we believe our method are more flexible in terms of this.

We have found that, with a larger number of data points, the number of mixture components also increases and this comes with much expensive computational cost with our optimization method especially for stronger penalties ( $\lambda = 3.0$  and above including those with  $N$  dependencies). The failure in optimization method suggests that it reflects optimization on a surface where the likelihood gives limited 'downhill' information compared the strong penalty on parameter numbers. The failure of the objective function on the other hand, reflects that the penalty (i.e.  $\lambda = 1.0, 1.5$ ) might be too small and that the use of a larger penalty ( $\lambda = 2.0, 2.5$ ) is a pragmatic correction. Of the criteria using sample size,  $N$  dependent corrections such as BIC, CAIC, and HQC, we found no significant advantage to the less penalized objective functions ( $\lambda = 2.0, 2.5$ ).

## 2.5. Conclusions

In relation with the theory discussed above, these results with simulated data suggest that the optimization method is most successful with AIC type objective functions without  $N$  dependency on the penalty term, and that the actual AIC ( $\lambda = 2.0$ ) or the use of slightly higher penalties for small data sets AIC ( $\lambda = 2.5$ ) are effective choices with this simulated data. Overall the results on simulated data indicate that the method is an effective way

of clustering data points described by a mixture of binary and continuous variables and support the its application to important problems in genomics as described in the Chapters which follow.



## Chapter 3. Modelling *S. cerevisiae* cell cycle transcriptional regulation using a model-based joint clustering algorithm

### 3.1. Introduction

The biology behind cellular behaviour is of fundamental interest most of the time in the cell biology field. It can be dictated by various complex factors such as interactions or relationships between molecules as well as responses from external environmental perturbations. Understanding some of the complex molecular interactions existing in cells has been made possible by representing them in sophisticated transcriptional regulatory networks (TRNs) using advanced molecular and computational biology methods. A TRN is a network that describes gene expression as a function of regulatory inputs specified by interactions between proteins and DNA [76]. This was inferred earlier just by using gene expression data. There is an enormous literature related to clustering of gene expression patterns, derived from measurements by microarrays or RNA-seq [77-79]. This has been applied to the discovery of transcriptional regulatory networks in *S. cerevisiae* [77, 78, 80, 81] as well as in other organisms. While this approach is useful, it is fundamentally limited by the fact that the regulators (transcription factors, signalling molecules) do not always share the expression patterns of the genes they regulate, making the discovery of some real regulatory interactions difficult.

Regulation almost certainly depends on combinatorial binding of several transcription factors, where positive and negative regulation may correspond to different combinations [82]. TFs are also often post-transcriptionally or/and post translationally regulated and thus, the binding of TF proteins to DNA may provide direct evidence of gene expression regulation. Recently the techniques of ChIP-chip, ChIP-seq and DNase-seq have enabled the direct measurement of regulation, at least in so far as regulation that is effected through the binding of relevant factors to genomic DNA in the vicinity of the regulated gene. Thus, a complementary approach in combining TF binding and gene expression profiling with the appropriate computational method could elucidate complex TRNs for yeast cellular processes (e.g. cell cycle progression).

Since genes with similar expression profiles often function similarly, parallel information on regulation or TF binding to these gene promoters/enhancers could provide meaningful insights into the orchestration of gene activities of a cell. Previously, shared regulation was assumed to correlate with the co-expression of the target genes [78]. However, temporal delays for a TF in exerting its condition-specific regulatory function has often not been taken into account, as well as the combinatorial effects of multiple TFs, thus, weakening this assumption [83]. This was then improved to include binding motifs

present in the upstream/promoter of co-expressed genes [84]. Motif data only indicates potential binding sites and thus provides less direct evidence of regulation.

Using the developed clustering method as explained in the previous chapter, we are now going to test it further and try to address the aforementioned limitations by applying this method to jointly cluster yeast cell cycle gene expression patterns and regulatory information. The recent approach to discover clusters/modules uses LeTICE [85]. This is a probabilistic approach where it maximizes the likelihood of binary binding matrix given the location data and expression data and removes genes to the background if it does not have similar expression and binding patterns. Our algorithm is different from LeTICE in the optimization approach where we sought to maximize the likelihood of a model or set of clusters using likelihood of binding and using meta-heuristic simulated annealing approach in searching for optimum model. In addition, we have applied information criteria in our model selection and we did not impose any pre-defined minimum number of genes in a cluster. A transcriptional regulatory network can be built by using confident regulatory predictions for each cluster by TF(s) whose binding probability/confidence is large enough to be considered as the cluster regulator.

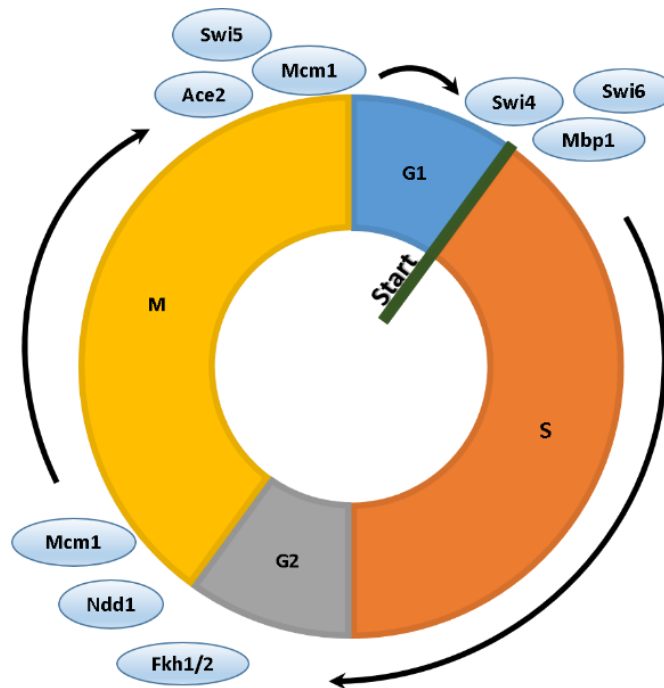
### **3.1.1. Why yeast?**

First, *Saccharomyces cerevisiae*, has been a popular model organism and its usefulness in biological research has been demonstrated widely especially in understanding molecular mechanisms that govern the cell division and cell cycle progression in higher organisms including humans. Yeast has approximately 6000 genes and a large proportion are still uncharacterized experimentally since its genome was first published in 1996 [86]. Perturbations of yeast cells can provide clues about how our cells behave or even benefit 'key players' in the food industry such as bakers and brewers. Given the importance of yeast to biologists and the fact that it can be easily manipulated, grows and copes with different environmental perturbations, any hypothesis generated from computational modelling of molecular mechanisms that regulate yeast cell behaviour could be tested experimentally. Our case study here is the reconstruction of the yeast cell cycle transcriptional regulatory network, more specifically, to model the relationships between transcription factors and their target genes that regulate the yeast cell cycle.

### **3.1.2. Yeast cell cycle control**

Yeast cells divide rapidly with a cell cycle time of between 90 minutes and 2 hours. Budding of yeast involves a cycle of mitosis and is generally studied in the haploid state. The stages in the yeast cell cycle are similar to other eukaryotic cells. It involves two

main phases, namely, S phase and M phase with two gap phases, the G1 and G2 between the main phases (see Figure 3.1 below).



**Figure 3.1** The events during the eukaryotic yeast cell cycle.

The main events of cell cycle are chromosome duplication (S phase), and chromosome segregation, nuclear division, and cell division (M phase). G1 phase is the gap phase between M and S phases, whereas G2 is the gap phase between S and M phases. A yeast cell decides whether to commit to a new cell cycle during the start-transition (START) in the G1 phase. Also shown on this diagram is the canonical model of yeast cell cycle regulation from transcription factor binding data of eight well-known cell cycle transcription factors [87, 88].

During S phase, DNA is replicated and chromosomes are duplicated by proteins carrying out DNA synthesis. Protein synthesis (e.g. histone proteins) is required as the DNA needs to be packaged into chromatin (chromatin condensation). The duplicated chromosomes are known as sister chromatids. Cytoplasmic components are duplicated as well throughout the cell cycle. The transition phase between S phase to the next M phase is known as G2 phase. In G2 phase, additional time is provided for cell growth, duplication and segregation as well as protein synthesis, as the cell prepares for mitosis.

During M phase, two major events occur which are mitosis and cytokinesis. During mitosis, sister chromatids are distributed equally into a pair of daughter nuclei [89]. This major phase is divided into two other sub-phases called metaphase and anaphase. In metaphase, pairs of sister chromatids are attached to the bipolar mitotic spindle oppositely. Contraction of spindle fibres forces sister chromatid separation towards opposite ends of the cell. During cytokinesis, the cell division occurs where a new plasma membrane and cell wall are generated and contraction of actin filaments and myosin

under the cell membrane takes place. The resulting two daughter cells fate will be determined at G1 phase which acts as the checkpoint of the cell cycle progression [89]. The start-transition (START) checkpoint at the end of the G1 phase will determine the cell cycle progression depending on cell mass as well as on environmental cues such as nutrient availability and mating pheromone [90].

Regulation of yeast cell-cycle dependent genes has been investigated rigorously. Transcription factors which regulate these genes have been identified and include Ace2, Mbp1, Mcm1, Ndd1, Swi4, Swi5, Fkh1, and Fkh2 [88, 89, 91]. MBF- a complex of Mbp1 and Swi6 and SBF-a complex of Swi4 and Swi6 control the activation of genes required in the transition between G1 phase to S phase by binding to the DNA sequence elements called MCB and SCB respectively [88, 89]. The genes activated during this phase include the cyclins (Cln1, Cln2 and Cln3). Cyclins regulate cyclin dependent kinases (Cdks) e.g. Cdk1, which promotes cell cycle progression to the S phase [87]. In addition, SBF/MBF heterodimer also promotes the activities of S phase cyclins, Clb5 and Clb6. At the transition between G2/M phase, another regulatory protein complex, Mcm1-Fkh1/2-Ndd1 activates the expression of G2/M genes responsible for mitotic regulatory proteins, e.g. Clb2 and Cdc20 which are required for mitotic entry and mitotic exit activity [87]. At the late mitosis M/G1 phase, TFs Swi5 and Ace2 stimulate expression of M/G1 genes responsible for mitotic exit and cytokinesis.

Around 204 transcription factors have been identified in *Saccharomyces cerevisiae* but its functional regulatory networks have not been fully discovered. Mapping functional regulatory networks requires characterized functional interaction of TFs to their targets. The degree of complexity involved in the functional regulatory network mapping of this simple organism is high, where the observed TF-DNA interactions are not necessarily direct (i.e. through interaction of TF with other proteins) - and involves a cascade of downstream gene activation.

Furthermore, not all TFs binding regulates gene expression. This is an inherent concept that should be noted in building gene regulatory networks. Haynes and co-workers (2013) have highlighted this issue of irrelevant binding of TFs where they found that 98% of yeast genes bound by TFs but only 45% of these genes were actually regulated by those TFs when perturbation on TFs were performed [92]. Joint clustering of gene regulatory information (i.e. TFs binding) with gene expression (i.e. cell cycle, knock out of the TFs, or other type of perturbations) could provide insights on functional as well as irrelevant binding of the TFs and could also discover new interactions in the networks. To date, investigations on regulatory control have been either in small-scale systems dealing with small genomic regions and a few genes, or focused on general properties

of genome scale systems neglecting the detail of control of any individual gene. We are aiming to generate models that yield greater mechanistic insight into the regulation of individual genes and groups of genes, using a novel probabilistic approach to jointly cluster regulatory information and gene expression patterns in yeast data.

### 3.2. Methodology

To test if our flexible model-based clustering can be applied to genetic regulation, we used the well-studied yeast cell cycle system. The experimental data set includes ~6000 genes from yeast cell cycle genes expression using microarray technology across 18 time-points from Spellman and co-workers [77]. The yeast cells in this experiment were sampled in rich media at each time point following  $\alpha$ -factor synchronization of cells at the very beginning of the experiment. For the transcription factor binding data set, 103 yeast transcription factors (TFs) from the regulatory map published by Harbison and co-workers [93] were retrieved. Yeast TFs binding from [93] and yeast cell cycle genes expression from [77] are widely used for constructing yeast functional regulatory networks. Our working hypothesis is that yeast has an underlying gene regulatory network that is always the same, thus, using a combination of datasets from different experiments is possible and has been used previously in research that constructed the yeast transcriptional regulatory networks. Before proceeding to the clustering, both genes expression and TF binding data were subjected to pre-processing.

#### 3.2.1. Pre-processing of genes expression data

The gene expression data set has missing values. Imputation of the missing values was done by replacing the missing values with the newly calculated expression value,  $e_i$  as follow,

$$e_i = e_{i-y} + dg$$

$$g = \frac{e_{i+x} - e_{i-y}}{(i+x) - (i-y)}$$

where  $g$  is the gradient between the nearest values before and after the missing value, ' $i - y$ ' and ' $i + x$ ' are the previous and next available expression value index for missing expression value  $i$ ,  $e_i$  respectively, and  $d$  is the difference between  $i$  and  $i - x$ . If the missing value located at the beginning and end of the dataset, we constitute the missing value with the nearest available value after or before them respectively. We reason that this is a viable approach as the data set involves a time series and generally, cell cycle gene expression gradually increases and decreases across small time interval, in this

case, an interval of 7 minutes. As a result, we have a complete expression dataset without any missing values.

### **3.2.2. Pre-processing of TFs binding data**

Given limited number of genes showing cell cycle expression patterns, not all TFs found by ChIP-chip regulate these yeast cell cycle genes. Based on the estimates from previous studies, around 10-20 TFs are likely to be involved in cell cycle related regulation [88, 94, 95]. We began with 525 genes identified by Spellman and co-workers as showing cell cycle related expression [77]. As for the TFs binding data, we used the publicly available data produced using the ChIP-chip technique. A pre-processed dataset for 103 TFs binding by Harbison and co-workers [93] was chosen and retrieved. Combining several motif discovery tools (i.e. AlignACE, MEME, Mdscore) with conservation information across yeast species and promoter regions bound by specific regulators, they discovered the putative sequence motifs that are bound by the transcription factors and mapped these binding sites (regulatory code) to the yeast genome. Each binding interaction was given a p-value by Harbison and co-workers and we have chosen a dataset with binary data matrix of TFs binding where '1' corresponds to TF binding with a p-value less than 0.005 and intermediate-confidence conservation criteria.

Our preselection of TFs was based on the pre-selection step for the LeTICE algorithm [85]. This is based on the hypothesis that if a TF is active in regulating any of the selected genes, then within the set of genes whose promoters it binds, there should be some gene pairs showing highly correlated expression patterns reflecting common regulation, even allowing for the possibility that the TF does not regulate all the genes that it binds. Therefore using the 95th percentile,  $\rho$ , of Pearson correlation coefficients over all gene pairs, the proportion of correlations greater than  $\rho$  in the gene set that the TF binds is calculated. This is then compared to the proportion of correlations greater than  $\rho$  in randomly selected gene sets of the same size, and an empirical p-value calculated. If this p-value is less than the generous threshold of 0.1 then it is assumed that the TF may regulate some genes and it is retained, otherwise the TF is removed from the set under consideration. In this case 17 TFs were retained for input to the main clustering algorithm, on the assumption that these TFs are the ones likely to be regulating cell cycle genes. The list of 17 TFs are shown in Table 3.1 below. Following this, the set of 525 genes was further reduced to 328 by eliminating genes not bound by any of the selected TFs.

### 3.2.3. Comparison with an existing method

LeTICE [85] was also used as an alternative method for comparison with our approach. LeTICE is not a generic clustering method but is designed specifically for the problem of genetic regulatory network prediction. It is based on integrating TF binding data with expression pattern data to define a genetic regulatory network, i.e. a set of clusters each comprising genes with a common TF binding pattern and a shared pattern of expression. This is achieved by finding the network,  $B$ , which maximizes

$$P(B|L, E)$$

where  $L$  is a matrix of TF binding probabilities and  $E$  a matrix of gene expression patterns. LeTICE location matrix consists of rows corresponding to genes and columns corresponding to TFs. According to LeTICE, the p-value (for the hypothesis that there is no interaction between a TF and the promoter of a gene) will be more pragmatic than deciding on a threshold for p-values.

As such LeTICE is a method based on a similar premise of integrating TF binding data and expression data to find regulatory relationships, but being based on different underlying methodology it is an ideal comparator, albeit only relevant to the problem of genetic regulation. To provide a direct comparison of algorithms, LeTICE was applied to the dataset described above. Note that LeTICE takes binding  $p$  values directly as input and that it has its own TF and gene pre-selection criteria, in this case it selected 18 TFs and 289 genes. LeTICE was then run with the optimum runtime parameters suggested in the original paper.

As part of this study we also examined the effect of using normalized (where each gene was normalized to zero mean and unit standard deviation) and un-normalized gene expression data. We also compared joint clustering to clustering expression data separately, which can be done by simply omitting binary variables in the input to our program.

**Table 3.1** The filtered TFs and their functional information.

With the exception of TFs in blue, the rest of the TFs are with known involvement in cell cycle. The functional descriptions were retrieved from the *Saccharomyces cerevisiae* Genome Database (SGD) [96].

TF	Cell cycle related functions
ACE2	Sequence-specific DNA binding RNA polymerase II transcription factor involved in G1/S transition of the mitotic cell cycle; activates cytokinetic cell separation; also regulates antisense transcription at diverse loci; localizes to both nucleus and cytosol
ASH1	Sequence-specific DNA binding RNA Pol II transcription factor that up and down regulates transcription; its role as transcription repressor negatively regulates mating type switching, G1/S transition of mitotic cell cycle; its role as transcription activator positively regulates pseudo-hyphal growth; subunit of the Rpd3L histone deacetylase complex; also localizes to the cellular bud
DAL80	A RNA polymerase II transcription factor that binds specific DNA sequence; negatively regulates transcription and is involved in nitrogen catabolite repression of transcription; localized to the nucleus
FKH1	Sequence-specific DNA binding transcription factor involved in chromatin remodeling, mitotic transcription regulation, transcription termination, mating-type switching, and pseudo hyphal growth; binds DNA replication origins and positively regulates replication initiation; also binds centromeres
FKH2	RNA polymerase II transcription factor involved in positive and negative regulation of transcription during mitotic cell cycle; positively regulates DNA replication initiation; binds replication origins and promoters in sequence-specific manner; localizes to cytosol and nucleus
GZF3	GATA zinc finger protein; negatively regulates nitrogen catabolic gene expression by competing with Gat1p for GATA site binding; function requires a repressive carbon source; dimerizes with Dal80p and binds to Tor1p



**Table 3.1** The filtered TFs and their functional information.

With the exception of TFs in blue, the rest of the TFs are with known involvement in cell cycle. The functional descriptions were retrieved from the *Saccharomyces cerevisiae* Genome Database (SGD) [96]. (**Continued**)

TF	Cell cycle related functions
MBP1	Sequence-specific DNA binding transcription factor that positively regulates transcription by RNA polymerase II involved in the G1/S transition of mitosis; subunit of the MBF (Mlu1 cell cycle box Binding Factor) transcription complex
MCM1	Transcription factor; involved in cell-type-specific transcription and pheromone response; plays a central role in the formation of both repressor and activator complexes; relocalizes to the cytosol in response to hypoxia
MET31	Zinc-finger DNA-binding transcription factor; targets strong transcriptional activator Met4p to promoters of sulfur metabolic genes; involved in transcriptional regulation of the methionine biosynthetic genes
NDD1	Transcriptional activator essential for nuclear division; localized to the nucleus; essential component of the mechanism that activates the expression of a set of late-S-phase-specific genes; turnover is tightly regulated during cell cycle and in response to DNA damage
PDR1	Sequence specific DNA-binding polymerase II transcription factor that activates expression of genes involved in drug response
RCS1	Sequence-specific DNA binding transcription factor that regulates chromatid cohesion, chromosome segregation, and cellular iron homeostasis; localizes to the cytoplasm, nucleus, and kinetochores
STB1	Protein with role in regulation of MBF-specific transcription at Start; phosphorylated by Cln-Cdc28p kinases in vitro; un-phosphorylated form binds Swi6p, which is required for Stb1p function; expression is cell-cycle regulated
SWI4	DNA binding component of the SBF complex (Swi4p-Swi6p); a transcriptional activator that in concert with MBF (Mbp1-Swi6p) regulates late G1-specific transcription of targets including cyclins and genes required for DNA synthesis and repair

**Table 3.1** The filtered TFs and their functional information.

With the exception of TFs in blue, the rest of the TFs are with known involvement in cell cycle. The functional descriptions were retrieved from the *Saccharomyces cerevisiae* Genome Database (SGD) [96]. **(Continued)**

TF	Cell cycle related functions
SWI5	Transcription factor that recruits Mediator and Swi/Snf complexes; activates transcription of genes expressed at the M/G1 phase boundary and in G1 phase; required for expression of the HO gene controlling mating type switching; localization to nucleus occurs during G1 and appears to be regulated by phosphorylation by Cdc28p kinase
SWI6	Transcription cofactor; forms complexes with Swi4p and Mbp1p to regulate transcription at the G1/S transition; involved in meiotic gene expression; also binds Stb1p to regulate transcription at START
YHP1	Homeobox transcriptional repressor; binds Mcm1p and early cell cycle box (ECB) elements of cell cycle regulated genes, thereby restricting ECB-mediated transcription to the M/G1 interval

### 3.2.4. Functional analysis of clusters

In the evaluation of our method, we considered comparison with the known literature on gene regulation in the yeast cell cycle, as well as measures of the functional coherence of clusters based on Gene Ontology (GO). The GO enrichment was performed for each cluster using GO annotation from DAVID Bioinformatics Resources 6.7 [53]. The statistical significance of GO term enrichment was measured by the EASE score- a modified Fisher Exact P-value using the pre-selected 328 yeast cell cycle genes as background. However, as an overall performance of clustering, we have used the semantic similarity measure using the 'Rel' method in GOSemSim package in R [97]. It gives a value between 0 (un-enriched/not functionally similar cluster) and 1.0 (enriched/functionally similar cluster) for GO terms (biological process) similarity for genes in a cluster. Higher score indicating higher similarity in GO terms for genes within a cluster.

In order to test the significance overall of the clusters average score of GO semantic similarity, we randomly shuffled the genes between clusters but keeping the cluster sizes intact for all clusters. By doing this, we have reduced the bias that could be introduced when using different cluster sizes. After repeating this randomization 10000 times, we then calculate an empirical p-value, which is the frequency of random overall clusters average semantic similarity score occurs greater than the real average similarity score.

Since our method can identify combinatorial regulation (a cluster of genes regulated by more than one TF), and this implies potential interactions between TFs, we also compared these implied interactions with physical and genetic evidence in BioGRID [98]. As for selecting a set of relevant regulators related to the well-known cell cycle regulators in the discussion section, KEGG was used to retrieve all genes related to the cell cycle pathway [99].

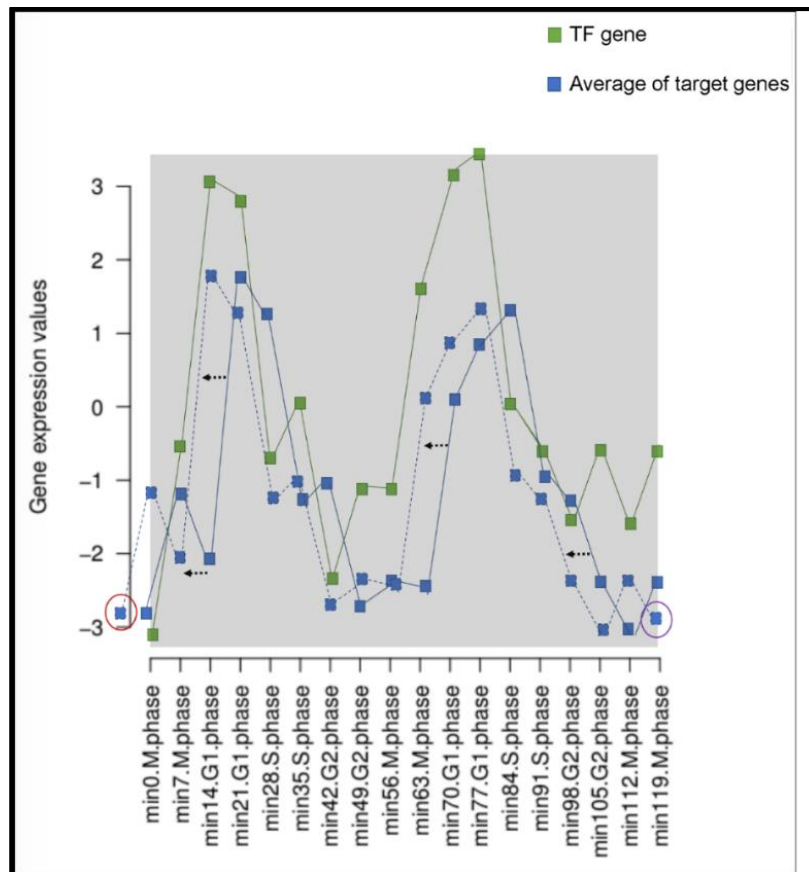
### 3.2.5. Time-lagged correlation calculation for TF-gene interactions

TF-gene interactions are widely inferred using the general framework of gene co-expression. However, it is also well established that a TF exerts its regulatory effects on its target genes in a time-lag manner [91, 100]. We would like to evaluate the proportion of our TF-cluster of gene interactions inferred from our clustering output that are supported by a generic co-expression method (i.e. correlation coefficient,  $r$ ) and as well as time-lag co-expression as the function for TF-gene interaction method [100]. Correlation between two signals or in our case, between TF gene expression and its target gene expression is a linear measure of similarity between them. Cross-correlation or what we referred to as time-lag correlation is somewhat a generalization of

the correlation measure where it takes into account the lag of one signal relative to the other. If the lag is equal to zero, then the time-lag correlation is equal to the generic correlation. Time-lag correlation is particularly important to assess the causal relationship between two signals in time, here, in our case, either the binding of the TF causing activation (positive time-lag correlation) or repression (negative time-lag correlation). Consider two time series,  $x(i)$  and  $y(i)$  where  $i = 0, 1, 2, \dots, N - 1$ . The time-lag correlation,  $r$  at lag,  $d$  is defined as

$$r_d = \frac{\sum_i [(x_{i+d} - \bar{x}) \cdot (y(i) - \bar{y})]}{\sqrt{\sum_i (x_{i+d} - \bar{x})^2} \cdot \sqrt{\sum_i (y(i) - \bar{y})^2}}$$

Where  $\bar{x}$  and  $\bar{y}$  are the means of the corresponding series. The above is computed for lags,  $d = 0, 1, \dots, 6$ . Here, we assumed that the time series is circular in nature in which case, the out of range indexes are 'wrapped' back within range, for example,  $x(-1) = x(N - 1)$  and  $x(N + 1) = x(1)$ . This is represented in Figure 3.2 where the out-of-bound data point circled in red is wrapped back into the range as circled in purple.



**Figure 3.2** A representation of co-expression between a TF gene and an average expression pattern of genes in a cluster.

Here, the average expression values of target genes is shifted to one time-point to the left (black arrows and blue dotted line '----'). Since cell cycle is cyclical in nature, we move the outlier circled in red to the same phase it is in the opposite/other side of the plot.

### 3.3. Results

We applied our method to yeast cell cycle data using parameters suggested from the simulation study. These parameters were set up same as with our simulation study prior to the program executions due to the small number of data points in yeast test case (328 genes with 17 TFs). A relevant set of runtime parameters is, StartTemp=500; TempFactor=0.999; MaxTemp=100,000; MaxReplters=2000. 328 yeast cell cycle genes considered here would fit with the simulated data of between 200-500 data points. Together with this, we have tested the method with expression only and regulatory input only data. LeTiCE was also tested using a similar input but with p-values as the location matrix not a binary binding matrix. We also examined the effect of refinement of clusters using the EM algorithm.

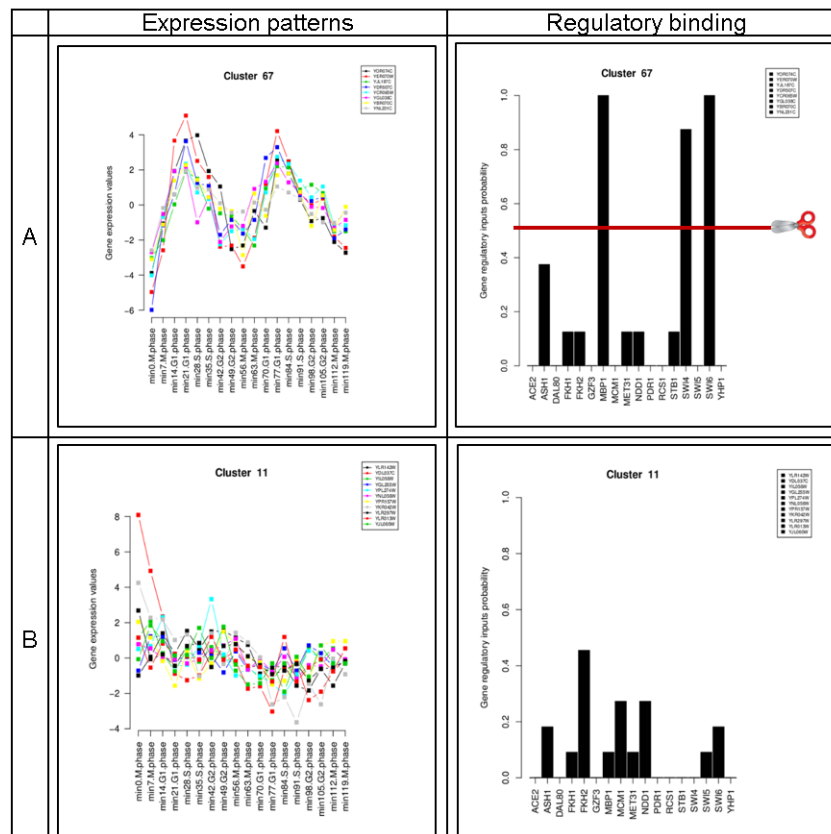
#### 3.3.1. Overall clusters statistics

The details of the results of clustering with different methods and parameters can be found in Table 3.2. The actual cluster solutions for AIC considered in this analysis can be found in Appendix B. Here, a different number of clusters were produced by using different clustering methods; AIC found the highest number of clusters (76 clusters) and both, using AIC ( $\lambda = 2$ ) and AIC ( $\lambda = 2.5$ ) with normalization produced smaller number of clusters than without normalization (see Table 3.2). Consistent with the simulation results, AIC ( $\lambda = 2$ ) produced a larger number of clusters with smaller cluster average size than AIC ( $\lambda = 2.5$ ). Theoretically, by normalizing expression values across time-points to the mean and one standard deviation (z-scores), we would expect that the number of clusters found will be smaller as genes with high and low expression levels but with similar patterns will be clustered together. In concordance with this theory, normalizing gene expression values results in fewer clusters compared to un-normalized expression patterns.

Number of clusters alone would not be enough dictate the performance of each clustering method, hence, we used the functional analysis of clusters as a measure of a method's performance. It is important to have good clusters as the basis for TRN construction. We acknowledge that some data points such as gene expressions and/or TF binding data from the chosen yeast datasets might be noisy and thus we could expect that some clusters will pose difficulty in interpretation. We defined a 'clear' cluster as a cluster which has at least one TF bound to more than half of the genes in the cluster and with correlation of genes expression patterns greater than 0.5. Essentially the 'unclear' clusters may represent genes that don't have a strongly cell cycle related expression pattern or genes possibly regulated by other TFs. In other words they represent cases

where our filtering of genes and TFs may not have worked optimally. We consider these un-clear clusters contain limited information about the regulation of gene expression of the corresponding genes.

Although most genes in clusters found using either AIC ( $\lambda = 2$  or 2.5) have correlated gene expression patterns, this is not the case with corresponding gene regulatory binding patterns. Most of the un-clear clusters have no TF bound to at least half of the genes. Examples of a clear cluster and an un-clear cluster are in Figure 3.3a and Figure 3.3b, respectively. Cluster 67 is a clear cluster with clear expression patterns and clear regulatory binding patterns. In contrast to cluster 67, cluster 11 has less clear binding patterns with the highest TF proportion of binding or TF binding probability (i.e. FKH2) less than 0.5. Only clear clusters were considered to be used in building the yeast cell cycle predicted TRN (see ‘Number of clear clusters (total genes)’ in Table 3.2).



**Figure 3.3** Two examples of clusters showing both expression patterns and regulatory binding patterns.

The two examples of clusters showing both expression patterns and regulatory binding patterns were generated using the standard AIC objective function and normalized gene expression patterns. Panel A shows expression and regulatory binding patterns plots for genes in cluster 67: here a clear gene expression pattern is associated to a clear regulatory hypothesis involving high probability of binding by Mbp1, Swi4 and Swi6 (more than half of the genes are bound by TF(s) - see the red line cut off). On the contrary, clear regulatory hypotheses could not be made for cluster 11 in panel B: these genes do not have a very clear cell cycle expression pattern nor do they show a high probability of binding any transcription factor.

**Table 3.2** Statistics of clusters found by joint clustering of regulation and expression with different objective functions.

Statistics of clusters found by joint clustering of regulation and expression with different objective functions, AIC ( $\lambda = 2$ ) and AIC2.5( $\lambda = 2.5$ ), with and without normalization of gene expression, compared to using LeTICE and using expression alone. Gene symbols in red are the nine well known yeast cell cycle transcription factors. 1 'Clear' clusters have clear expression patterns (average pairwise Pearson correlation of expression  $> 0.5$ ) and clear regulation (at least one transcription factor with binding probability  $> 0.5$ ). Such TFs in clear clusters are considered candidate regulators for the cluster.

\* Statistically significantly different the values obtained by random assignment of genes to clusters with the same size distribution,  $p < 0.01$ .

	AIC normalized expression	AIC un-normalized expression	AIC2.5 normalized expression	AIC2.5 Un-normalized expression	LeTICE	Expression only	Regulatory input only
Number of clusters	76	91	23	33	14	19	11
Number of clear <sup>1</sup> clusters (total genes)	52 (236)	64 (252)	15 (225)	26 (272)	14 (136)	14 (252)	1 (5)
Average (+/- s.d.) size of clear <sup>1</sup> clusters	5 ± 3	4 ± 3	15 ± 6	11 ± 10	10 ± 5	34 ± 14	5
TFs found as candidate regulators for clear <sup>1</sup> clusters	Ace2 Ash1 Fkh1 Fkh2 Gzf3 Mbp1 Mcm1 Met31 Ndd1 Pdr1 Rcs1 Swi4 Swi5 Swi6	<b>Ace2</b> Ash1 <b>Fkh1</b> <b>Fkh2</b> Gzf3 <b>Mbp1</b> <b>Mcm1</b> Met31 <b>Ndd1</b> Pdr1 Rcs1 Stb1 <b>Swi4 Swi5 Swi6</b>	Ace2 Fkh1 Fkh2 Gzf3 Mbp1 Mcm1 Ndd1 Pdr1 Swi4 Swi5 Swi6	Ace2 Fkh1 Fkh2 Gzf3 Mbp1 Mcm1 Met31 Ndd1 Pdr1 Swi4 Swi5 Swi6	Bas1 Fkh2 HAP4 <b>Mbp1</b> Ndd1 Stp1 Swi4 Swi5 Swi6	Ace2 Fkh1 Fkh2 Mbp1 Mcm1 Swi5 Swi6	Gzf3 Pdr1
Average GO Semantic Similarity (Mean +/- s.d. for random clusters)	0.34* (0.27 +/- 0.02)	0.32* (0.25 +/- 0.02)	0.32* (0.26 +/- 0.02)	0.30* (0.25 +/- 0.02)	0.25 (0.24 +/- 0.01)	0.33* (0.24 +/- 0.02)	- -

Next, we analysed the relevant TFs recovered using each method. Using LeTICE, we found that Ace2, Fkh1 and Mcm1 as not relevant and non-cell cycle specific TFs. In addition, clustering using expression or regulatory input only also failed to recapitulate all the known cell cycle specific TFs. Comparing our joint-clustering methods, AIC ( $\lambda = 2$ ) and AIC ( $\lambda = 2.5$ ), with the clustering using expression only and regulatory input only, both were performed using AIC ( $\lambda = 2$ ) which we change all gene expression values to zero and all regulatory input values to zero, respectively before running our algorithm, we found that these approaches missed a few important cell cycle TFs as relevant TFs. Interestingly, clustering using regulatory input only found only two TFs and using expression patterns alone recovered seven cell cycle TFs out of nine well-known cell cycle TFs.

An interesting finding here is that cell cycle genes with similar expression patterns tend to be regulated by similar sets of TFs but nonetheless, genes regulated by similar set of TFs are sometimes expressed differently. This has resulted in better recovery of clear clusters from using gene expression alone method compared to regulatory inputs alone. By combining both, gene expression and regulatory inputs, much smaller and specific clusters were recovered and in this case, clusters were found by both AICs. Furthermore, AIC ( $\lambda = 2$ ) and AIC ( $\lambda = 2.5$ ) both performed similarly in terms of cell cycle TFs recovered as relevant regulators.

We then measured the functional coherence of the clusters using the average semantic similarity of Gene Ontology annotations of the clustered genes. By this measure, most methods produce clusters that are significantly better than a random assignment of genes to clusters of the same size distribution (see Table 3.2). For neither AIC type objective function is there any strong evidence of a difference in the results based on normalization of the expression data. When using LeTICE, the average GO term similarity is lower than AIC and its derivative, AIC2.5 with AIC being the best criterion in explaining the functional relationship of gene regulation to gene expression.

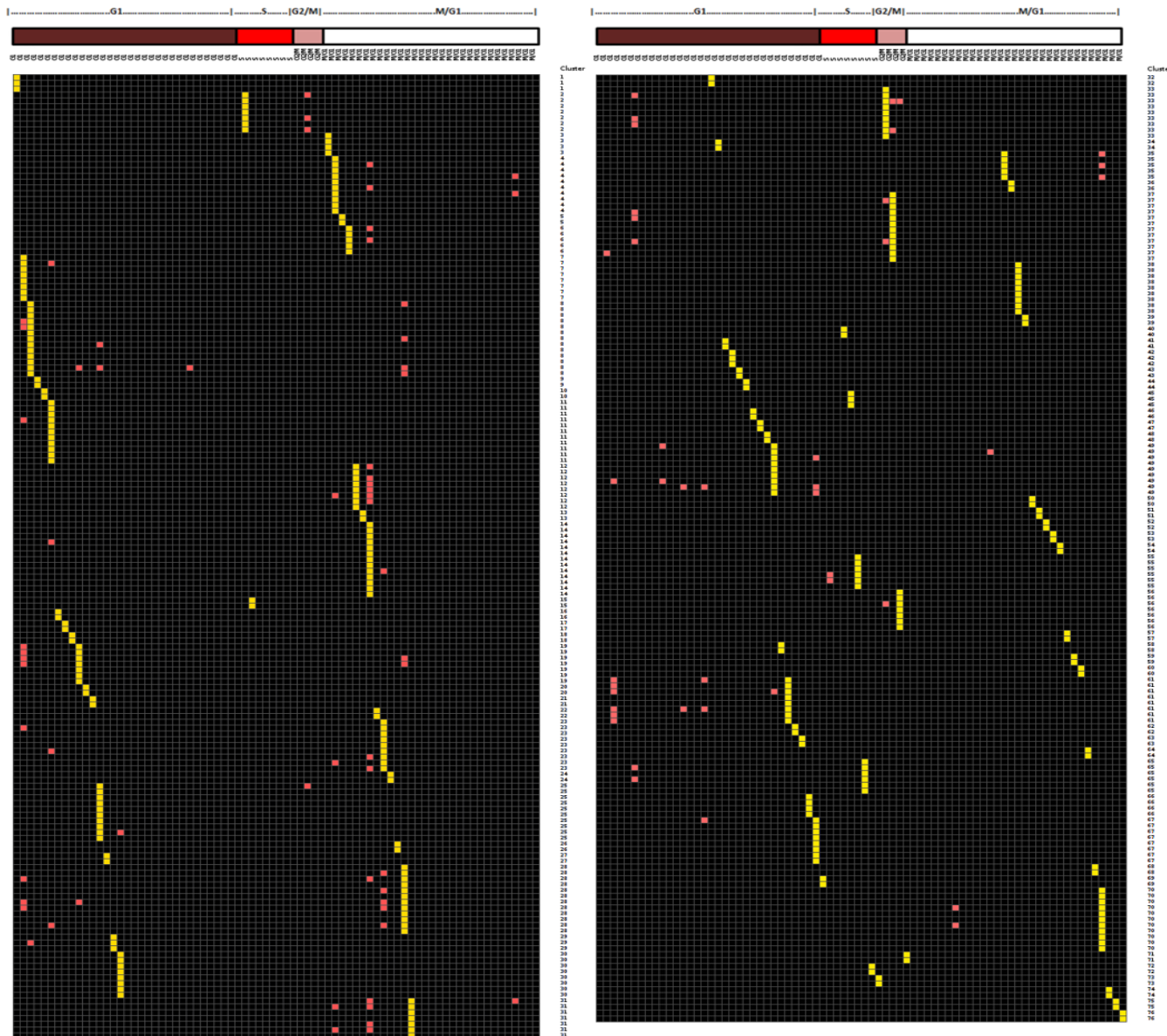
We have also found that expression only information could explain the functional relatedness of genes with similar expression and regulation better than LeTICE. Overall, by using AIC with normalization, better clusters were found compared to the rest of the objective functions and methods of clustering. Although clustering of normalized gene expression is generally a preferable method over using raw values, we have found and explained using evidence from our analysis that normalizing the gene expression across time-points resulted in more functionally related clusters membership compared to the un-normalized expression patterns in both AIC and AIC2.5.



Finally, based on GO criteria and the implication of more TFs in regulatory roles, we marginally preferred AIC ( $\lambda = 2.0$ ) with normalized expression and our subsequent analysis is based on these clusters. We chose to analyse our data in detail by extracting clear clusters found by AIC, in this case, of the 76 clusters produced (see Appendix B), 52 (see Appendix C) met the 'clear' cluster criteria.

### **3.3.2. Marginal densities of modules found using SA**

Upon convergence, our Expectation Maximization (EM) refinement of cluster output from heuristic search, the simulated annealing (SA) algorithm is able to suggest the probability of a gene being in each and every cluster found by SA or generally known as marginal densities. Figure 3.4 shows the marginal densities for each gene in all AIC clusters sorted horizontally based on cluster numbers and clusters are sorted vertically based on the phases of the cell cycle (e.g. G1, S, G2/M and M/G1). Using yeast as the test case, biological interpretations of marginal densities could be made which we could not do with simulated data because in yeast data, we are able to capture the marginal densities for some genes. In this test case, none of yeast cell cycle genes have equal probability to be in more than one module (marginal density is equal or greater than 0.5) and most of the genes with marginal densities have minute marginal density value ( $\sim 0.0001$ ) for other cluster(s). Upon detailed inspection of genes marginal densities, we have found that genes with marginal densities and their alternative clusters are all either expressed in the same cell cycle phase or with adjacent cell cycle phase without skipping a phase.



**Figure 3.4** Expectation maximization results.

Marginal densities ( $p(m|i, \theta)$  for gene  $i$  and cluster  $m$ ) are shown for all clusters found by AIC distributed over two heat maps. Figure legends are as follow:

**Salmon** : density approximately 0.0001

**Yellow** : density approximately 1.0

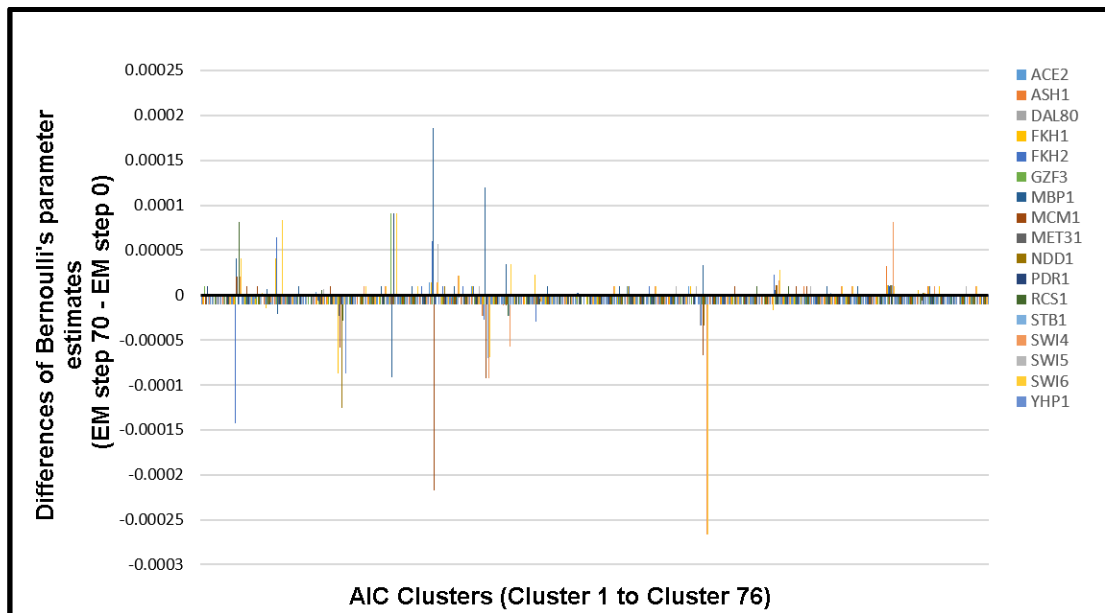
**Black** : density approximately 0

Rows : Genes in the 76 clusters.

Columns : 76 clusters sorted by peak of cell cycle phase from gene expression pattern in the cluster.

From Figure 3.4, we can see that all genes in cluster 1 have a high propensity to be in its own cluster (marginal density equals to 1.0). On the other hand, three genes in cluster 2 (salmon in colour) have higher propensity to be in its own cluster and with smaller probabilities to be in alternative cluster, and genes in this alternative cluster are expressed in G2/M phase, which is a neighbouring phase to the current cluster S phase of the yeast cell cycle.

In term of binding/Bernoulli's parameter estimates, upon EM convergence, the updated Bernoulli's parameter estimates (EM step 70) did not change much compared to the original clusters (EM step 0) as shown in Figure 3.5 below. Thus, the relevant TFs found after the EM refinement are still the same and this corresponds to the one presented in Table 3.2.



**Figure 3.5** Differences in Bernoulli's parameter estimates.

Cluster's negative/positive value of differences in Bernoulli's parameter estimates shows that the EM refined marginal densities have been updated since the SA solution. On average, there are really minute differences in the updated solution by EM (maximum of 0.0002 and minimum of -0.00025).

We have come to a conclusion that EM refinement reveals little overlap of the clusters where no genes have significant probability of membership of clusters other than the one assigned by the SA algorithm and parameter estimates for the clusters did not change significantly after refinement genes in each cluster are closely related to its own cluster than to the other cluster(s).

### 3.3.3. Statistical analysis of the clusters

Usually, when working with distance-based clustering [101], or model-based clustering like LeTICE [102], a thresholding on the minimum cluster size is implemented for example five genes per cluster. This is a generic robust assumption that at least five genes are needed for a cluster with genes having higher chance to be functionally related. It is notable that our choice of AIC as objective function produces a relatively large number of clusters, some of which are quite small. However, even very small clusters can sometime contain genes with similar biological functionality.

For example, clusters 40, 42, 48, 57 and 59 containing 2-3 genes each, have clear regulation (see Table 3.3) and contain genes with related functions. These clusters have statistically significant functional enrichment in cell division and cellular budding (clus. 48), drug transport and response to drug (clus. 57), chromatin assembly and disassembly (clus. 40), cell separation after cytokinesis (clus. 42), and cell wall organization (clus. 59). Hence, not limiting the minimum size of cluster during clustering is beneficial. Equally, there are often several clusters which are related in expression and regulation (see Table 3.4), for instance clusters 18, 29, 30, 49 and 67 whose expression patterns all peak in G1 phase and all show a high probability of regulation by the TFs SWI4, SWI6 and MBP1. These separate clusters have clearly different GO annotations: regulation of transcription (clus. 18), organelle organization (clus. 29), conjugation with cellular fusion (clus. 30), cellular budding (clus. 49) and deoxyribonucleotide biosynthetic processes (clus. 67), and their separation reflects differences in the detail of the expression pattern and regulatory probabilities.

**Table 3.3** Example of small sizes clusters with significantly enriched GO terms.

Clus.	Expression	Regulation	Regulator	Ensembl gene ID	GO term enrichments (Biological process) P-value < 0.05 ; p-value > 0.05
40			Swi4 Swi6	YDR224C YDR225W	Negative regulation of transcription, Chromatin assembly or disassembly
42			Ace2 Fkh1 Fkh2	YER124C YLR286C YHR143W	Cell separation after cytokinesis, cell wall organization

**Table 3.3** Example of small sizes clusters with significantly enriched GO terms. (Continued)

Clus.	Expression	Regulation	Regulator	Ensembl gene ID	GO term enrichments (Biological process) P-value < 0.05 ; p-value > 0.05
48			<p>Ace2</p> <p>Fkh1</p> <p>Swi5</p>	<p>YGR041W</p> <p>YBR158W</p>	<p>Cell division, Cellular budding</p>
57			<p>Rcs1</p>	<p>YOR153W</p> <p>YML116W</p>	<p>Drug transport, response to drug</p>

**Table 3.3** Example of small sizes clusters with significantly enriched GO terms. (Continued)

Clus.	Expression	Regulation	Regulator	Ensembl gene ID	GO term enrichments (Biological process) P-value < 0.05 ; p-value > 0.05
59	<p>Cluster 59</p>	<p>Cluster 59</p>	<p>Ash1</p> <p>Mcm1</p> <p>Swi5</p>	<p>YKL163W</p> <p>YJL159W</p>	<p>Cell wall organization</p>

**Table 3.4** Example of clusters which are related in expression and regulation and with significantly enriched GO terms.

Clus.	Expression	Regulation	Regulator	Ensembl gene ID	GO term enrichments (Biological process) P-value < 0.05 ; p-value > 0.05
18			Swi4 Swi6 Mbp1	YMR179W YBR071W	Regulation of transcription
29			Swi4 Mbp1	YPL267W YLR103C YPL124W	Cell cycle process, organelle organization



**Table 3.4** Example of clusters which are related in expression and regulation and with significantly enriched GO terms. (Continued)

Clus.	Expression	Regulation	Regulator	Ensembl gene ID	GO term enrichments (Biological process) P-value < 0.05 ; p-value > 0.05
30			<p>Swi4</p> <p>Swi6</p> <p>Mbp1</p>	<p>YGR189C      YKL103C</p> <p>YNL262W      YMR305C</p> <p>YGR221C      YPL256C</p> <p>YKR013W      YGR238C</p>	<p>Conjugation with cellular fusion, sexual reproduction</p>
49			<p>Swi4</p> <p>Swi6</p> <p>Mbp1</p>	<p>YIL140W      YER001W</p> <p>YML027W      YER111C</p> <p>YPR120C      YHR149C</p> <p>YKL045W      YGR152C</p> <p>YMR199W</p>	<p>G1/S transition of mitotic cell cycle, cellular budding, cell division</p>

**Table 3.4** Example of clusters which are related in expression and regulation and with significantly enriched GO terms. (Continued)

Clus.	Expression	Regulation	Regulator	Ensembl gene ID	GO term enrichments (Biological process) P-value < 0.05 ; p-value > 0.05
67	<p>Cluster 67</p>	<p>Cluster 67</p>	<p>Swi4</p> <p>Swi6</p> <p>Mbp1</p>	<p>YOR074C    YER070W</p> <p>YJL187C    YDR507C</p> <p>YCR065W    YGL038C</p> <p>YBR070C    YNL231C</p>	<p>Deoxy-ribonucleotide biosynthetic process, cell cycle check point, glycosylation</p>

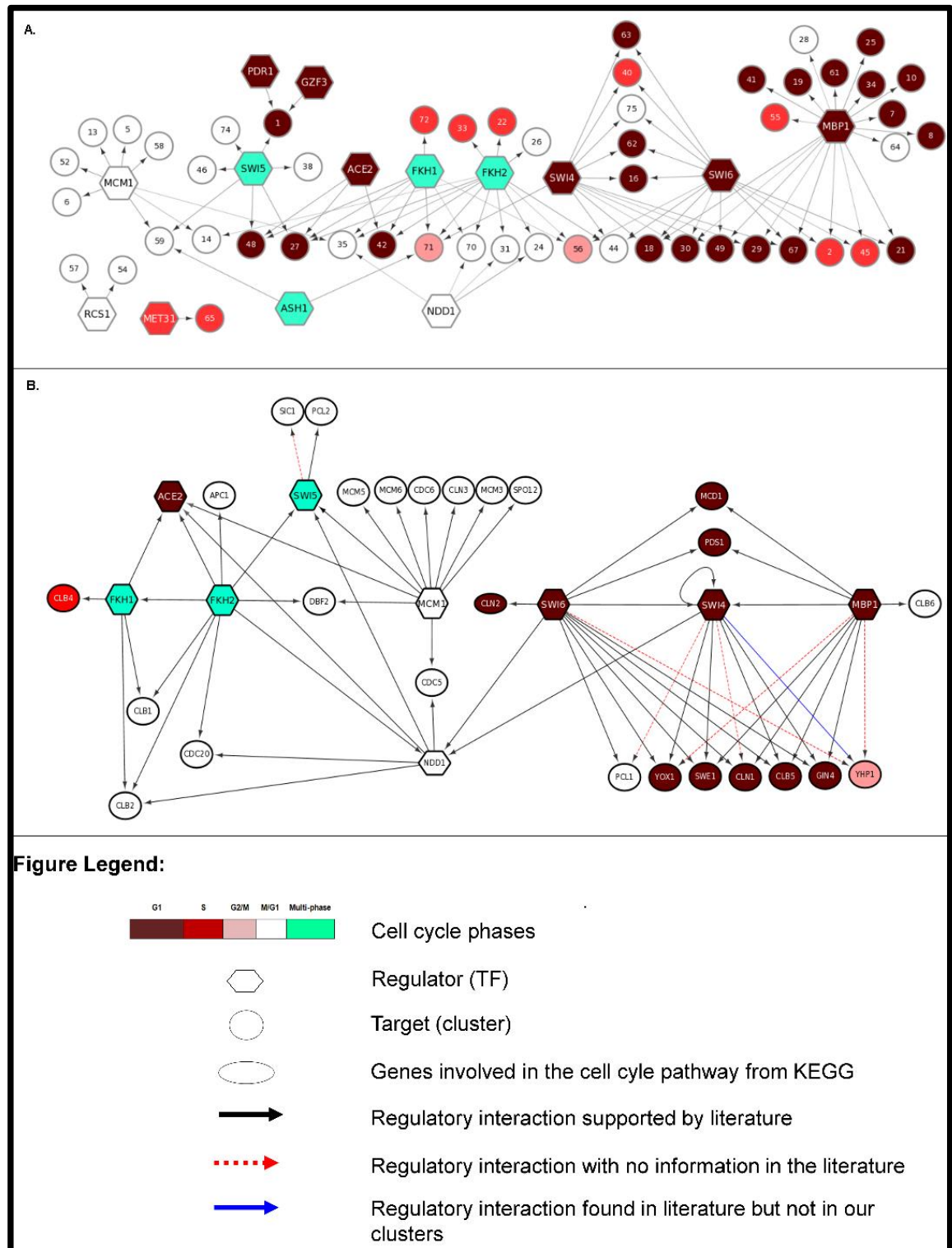
### 3.3.4. Yeast cell cycle TRN and regulators interactions

For the purpose of cluster validation, guided by the findings in Table 3.2, we have built a transcriptional regulatory network (TRN) as shown in Figure 3.6 where nodes are the relevant TFs and edges are the relationship between relevant TF and clear clusters where the binding probability is greater than 0.5 as explained in Section 3.3.1. The summary statistics of the regulatory interactions in Figure 3.6-Panel B are in Table 3.5 below.

i.	Total number of regulatory relationships in TRN	61
ii.	The number of i. that are known in the literature (true positive)	54
iii.	The number of i. that are not known in the literature (false positive)	6
iv.	The number of i. that are in the literature but not in the TRN (false negative)	1

**Table 3.5** A summary statistics for the regulatory network of transcription factors and other regulators.

The TRN was found to have a main component and two smaller disconnected components regulated by MET31 and RCS1. The regulation of the yeast cell cycle has been extensively studied both experimentally and in the context of algorithms aimed at reconstruction of the network from different sources of data (see [103] for a recent review). The regulatory relationships in Figure 3.6 are largely known, and most of the regulatory relationships in the lower panel are supported, as shown, by evidence from the literature (54 interactions out of 61 interactions are true positive). Some examples of known activation pathways of cell cycle related regulators that we have found are the regulation of Pcl2 through: Mbp1+Swi6 > Swi4 > Ndd1 > Swi5 > Pcl2; regulation of Ace2 through: Mbp1+Swi6 > Swi4 > Ndd1 > Swi5 > Ace2, and regulation of important B-type cyclins which control cell cycle progression (i.e. Clb1 and Clb2) through: Fkh1+Fkh2 for both cyclins and through: Swi4+Swi6+Mbp1 > Ndd1 + Fkh2 > Clb2 for Clb2. There are, however, a few regulatory interactions that we could not find evidence from the literature, for example, regulation of Yox1, Cln1 and Yhp1 by Mbp1, Swi4 and Mbp1/Swi6 respectively. Although there is not enough evidence or experiments have not been done on these interactions, it is possible that these regulatory interactions exist in vivo due to the fact that Mbp1, Swi4 and Swi6 are often found in a heterodimeric complex and act cooperatively.



**Figure 3.6** Transcriptional regulatory networks and regulatory interaction of our ‘clear’ clusters.

Panel **A**: The transcriptional regulatory network obtained from clusters with clear regulation and expression using the AIC objective function. The hexagonal nodes represent transcription factors and circular nodes regulated clusters (labelled 1-72, only clear clusters shown): colours represent cell cycle phases (peak expression phase for clusters, and the main phase of the regulated clusters for each transcription factor). Panel **B**: The regulatory network of transcription factors and other regulators extracted from the above network. Transcription factors shown are those associated by our algorithm to the regulation of clear clusters, and other cell cycle regulators were identified in our gene set and overlapped with cell-cycle pathway map in KEGG[104].

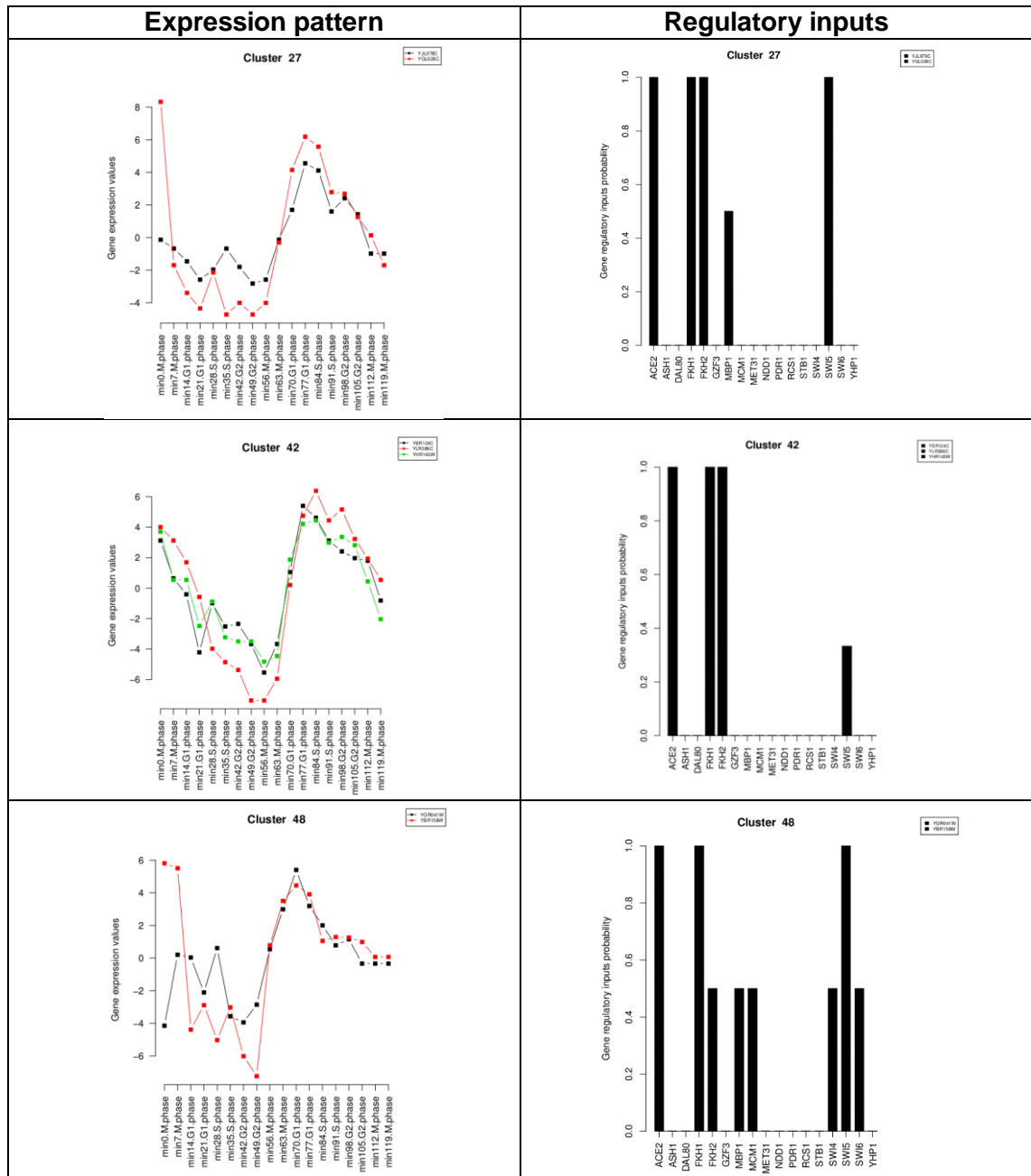
### 3.3.5. Biological analysis of the clusters

This section explains the biological relevance of the clusters found in Figure 3.6. To dissect this in an effective way, we described the TRNs and their regulation according to the cell cycle phases. All of the details of the clear clusters, such as expression, TF binding patterns and GO term enrichment can be found in Appendix C.

#### G1 phase

From Figure 3.6, we can clearly see that Mbp1, Swi4, Swi6 and Ace2 are the main regulators of G1 phase clusters and a few clusters in the S (clusters 2, 40, 45 and 55) and M/G1 (clusters 28, 44, 64, 75) phases. Mbp1/Swi6 and Swi4/Swi6 form heterodimeric complexes known as MBF and SBF, respectively [105, 106]. The G1 cyclins Cln1 and Cln2 are expressed in late G1 phase when they associate with Cdc28, which is a cyclin dependent kinase (CDK), to activate its kinase activity and complete progression through START. G1 cyclin expression depends on the transcription factor complexes, MBF and SBF [88, 103, 105, 107]. This then leads to initiation of early cell cycle events. Furthermore, S phase cyclins, Clb5 and Clb6 will usually increase during this phase to eliminate the suppression of S phase CDK (Cdk1) activity in G1 phase and drive entry into S phase [108, 109]. We have found in our extended network in Figure 3.6 (bottom) that Cln1 is regulated by SBF/MBF complexes and Cln2 by Swi6. In addition, SBF and Mbp1 but not Swi6 also promote the expression of S phase cyclins, Clb5 and Clb6, respectively. The enrichment of GO biological processes involved in these clusters include DNA replication, cellular response to stress, DNA replication, cell wall organization, regulation of kinase activity, re-entry to mitotic cell cycle, cytokinesis, cellular component disassembly and telomere maintenance. All these GO terms are involved in processes associated with cell cycle progression from G1 to S phase. Swe1 is a protein kinase that regulates the G2/M transition and as a negative regulator of Cdc28 kinase. Swe1 transcription is controlled by SBF [110]. Here, we have found that SBF, together with MBF regulates Swe1 expression. Our cluster 67 genes, to which Swe1 together with Gin4 map, are enriched in cell morphogenesis checkpoint, cytokinesis checkpoint and cell wall organization and almost all of these genes are regulated by SBF and MBF, thus there is a high chance that these complexes regulate Swe1 as well as Gin4. Another G1 TF, Ace2 also regulates three G1 clusters, namely clusters 27, 42 and 48. However, these clusters have genes that are bound by a different set of TFs along with Ace2 (see Figure 3.7) and with exception of cluster 27, cluster 42 and 48 are enriched in cell division, cell wall organization and cytokinesis GO terms. A separate G1 cluster (clus. 1) with three genes in it is regulated by Pdr1 and Gzf3 and all

are genes of uncharacterized protein function. There is no experimental evidence in the literature that, Pdr1 and Gzf3 are physically and/or genetically interacting.



**Figure 3.7** Cluster 27, 42, and 48 expression and binding patterns. Genes in this cluster are regulated by Ace2 and these clusters are maximally expressed in G1 phase.

## S phase

In S phase, DNA in the cell must be replicated for it to produce two daughter cells. DNA replication occurs during this S (synthesis) phase. This is also a crucial point where cells have to decide whether to continue with the cell cycle or to arrest in G0 phase- a non-dividing state. We would expect genes expressed in this phase are largely involved in biological processes related to this event. Regulation of S phase genes also by SBF and

MBF [111], particularly histones and genes associated with chromatin organization, is evident in clusters 2, 22, 40 and 55. Other S phase clusters (clus. 33 and 72) are regulated by FKH1 and FKH2 and are enriched in cell wall organization (clus. 33), chromosome partitioning (clus. 72) as cell is preparing to enter into M phase where cell division occurs. We note also the interesting disconnected component in Figure 3.5, cluster 65 being regulated by MET31, comprising genes associated with S-adenosylmethionine metabolism which has been linked to cell cycle control [112, 113]. Sulphur metabolism is involved in budding yeast and regulated by a variety of environmental and intracellular factors such as methionine.

### **G2/M phase**

Moving to G2/M and M phase, while SBF/MBF still participate in regulation, it becomes dominated by MCM1, NDD1, FKH1 and FKH2. G2/M phase is a period of rapid cell growth and protein synthesis during which the cell becomes ready for mitosis. G2/M phase clusters are 56 and 71 which are regulated by Fkh2/Fkh1/Swi6 and Ash1/Fkh1/Fkh2/Swi4 respectively. Only module 56 has an obvious functional enrichment in cell wall organization.

### **M/G1 phase**

In mitosis, a budding yeast cell separates the chromosomes in its nucleus into two identical sets in two daughter nuclei. During the process of mitosis, condensation of chromosome occurs and then spindle fibres pull the sister chromatids to opposite poles. This then proceeds into cytokinesis which divides the cell into two daughter cells. Clb1 and Clb2 are both M phase B-cyclins and both of them are expressed in M/G1 phase and regulate exit from mitosis [114]. The role of SBF in the regulation of Clb1 and Clb2 [115] as well as the role of Mcm1/Fkh2/Ndd1 in the regulation of Clb2 [116] are already known. However, our algorithm finds regulation of the key cyclins Clb1 and Clb2 by FKH1/2 and NDD1, but does not discover known links to SBF or MCM1. Other important M phase genes in the same cluster (Swi5 and Cdc20) and Ace2 in cluster 35 are regulated by Fkh1/Fkh2/Ndd1 and Mcm1/Fkh1/Fkh2/Ndd1, respectively. Zhu and co-workers (2000) [116] have found these genes lose their cell cycle regulation role in a mutant that lacks Fkh1 and Fkh2. In addition, the Fkh1/ Fkh2 mutant also displays aberrant regulation of the 'Sic1' cluster and genes in this cluster that was discovered by Zhu and Co-workers (200) are involved in mitotic exit [116]. This could be explained by our extended regulatory network (bottom-in Figure 3.6), where Sic1 is regulated by Swi5, and knock-out of Fkh1/Fkh2 will effect Swi5 transcription thus effecting the Sic1 cluster in a cascade manner. Apart from Fkh1/2 and Ndd1, few clusters in this phase are



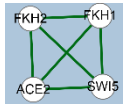
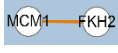

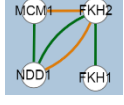
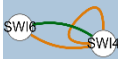


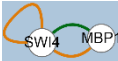
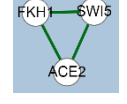

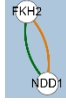
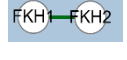

regulated by RCS1 (clusters 57 and 54), MCM1 (clusters 52, 6, 13, 5, 58, and 59), MCM1/ASH1 (clus. 59) and SWI5 (clusters 74, 46, and 38). The RCS1 cluster mainly function in multidrug transporter activity, whereas, genes regulated by SWI5 are more enriched in sexual reproduction. Most of the genes in the clusters regulated by MCM1 are not functionally enriched except for cluster 6 and cluster 59 which are involved in protein-DNA complex assembly and cell wall organization, respectively. These functions are important for preparation for M phase of the cell cycle.



Overall, we were able to obtain high confidence models/clusters of yeast cell cycle genes, thus utilize it in building our TRMs networks. We have been able to recapitulate the known cell cycle regulations and some unknown/novel regulation.

### **3.3.6. TF-TF interactions inferred from the clustering output**

Combinatorial regulation of genes by multiple TFs is known to be important and several of our clusters exhibited a high probability binding by more than one TF (e.g. regulation by SWI4, SWI6 and MBP1 in Figure 3.6). Such multiple regulation implies possible interaction between the factors concerned and in Figure 3.8 below, we summarize genetic and physical interaction evidence supporting combinatorial interactions in our clusters using interaction data recorded in the Biological General Repository for Interaction Database (BioGRID) [98]. *Saccharomyces* Genome Database (SGD) and BioGRID define physical interactions as direct physical binding of two proteins or co-existence in a stable complex and genetic interactions are indirectly inferred interactions between two or more mutants [96, 98]. Here, all but one (Gzf3-Pdr1-Swi5) identified combinatorial interaction is supported by evidence (i.e. physical and/or genetic interaction) from BioGRID [98] and most have extensive support.

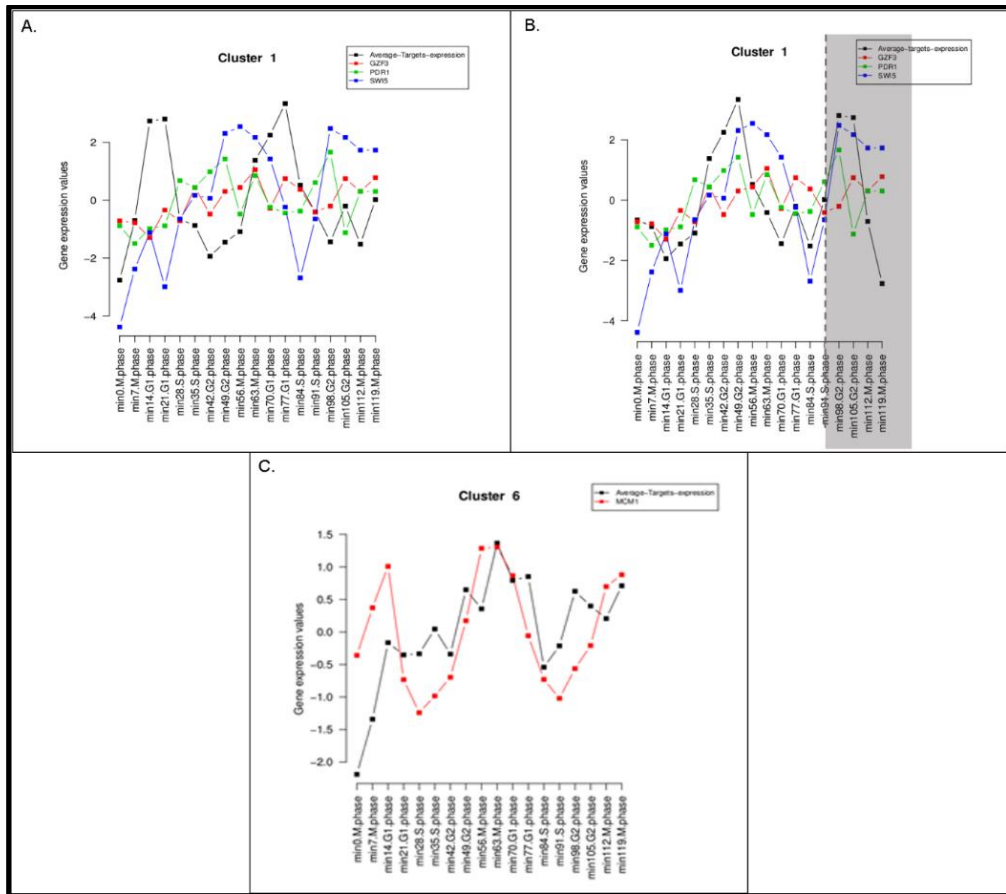


Double	Triple	Quadruple
Mbp1-Swi6 	Gzf3-Pdr1-Swi5 Mbp1-Swi4-Swi6 	Ace2-Fkh1-Fkh2-Swi5 
Fkh2-Mcm1 	Ace2-Fkh1-Fkh2 	Fkh1-Fkh2-Mcm1-Ndd1 
Swi4-Swi6 	Fkh2-Swi4-Swi6 	Fkh1-Fkh2-Ndd1 
Mbp1-Swi4 	Fkh1-Swi5-Ace2 	Ash1-Swi4-Fkh1-Fkh2 
Fkh2-Ndd1 	Fkh1-Fkh2-Swi6 	
	Ash1-Mcm1-Swi5 	

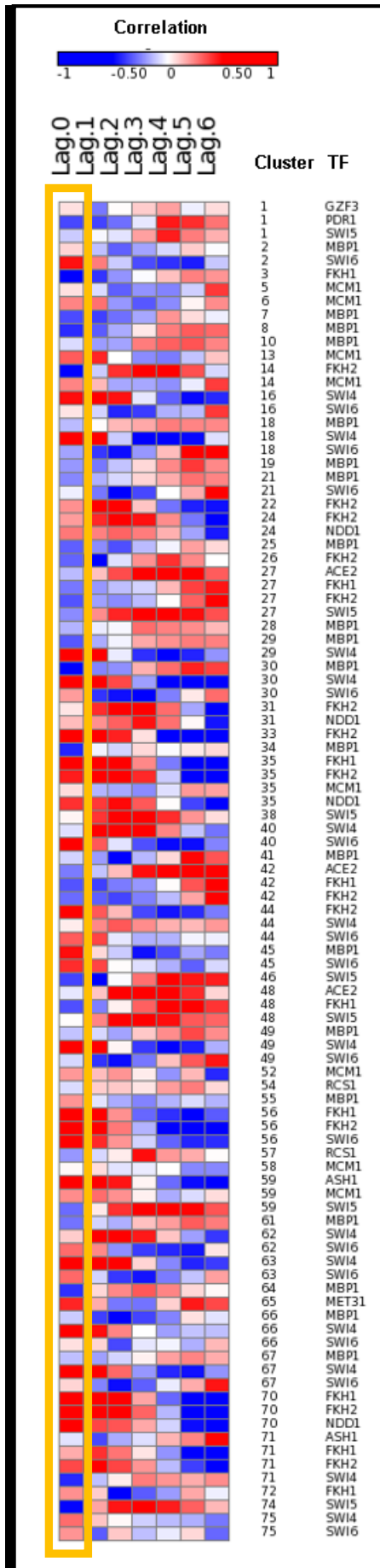
**Figure 3.8** Combinatorial regulatory interactions found in AIC clear clusters. Combinatorial interactions derive from clusters regulated by more than one TF (for instance cluster 67 in Figure 3.6) is regulated by SWI4, SWI6 and MBP1), and are listed in each column. Below each combinatorial interaction is a Figure showing the extent of support of the combination in physical and genetic interaction data in yeast from the BioGRID data base.  denotes a genetic interaction and  is a physical interaction. TFs in red indicate combinatorial interactions from our algorithm that are not supported in genetic or physical interaction data [99].

### 3.3.7. Time-lagged correlation analysis

Previously, TF-gene interactions during the yeast cell cycle were inferred using co-expression of a TF gene with its target genes across cell cycle time-points [77]. However, it is possible for a TF gene to be expressed with its target genes in a time-lag manner. We would like to investigate how many gene cluster transcriptional regulatory interactions (i.e. edges in Figure 3.6-above) are supported by TF gene-target gene with and without time-lag co-expressions. Figure 3.9 below shows an example of time-lag correlation where cluster 1 TFs are apparently more positively correlated at 4th lag (panel B) than at the real-time without time-lag (Panel A). In other words, TFs genes are expressed before their target genes expression. As we are interested in the activation of genes at specific cell cycle phase, when naturally interpreted, a lag means the time needed from the time a transcription factor is expressed until it acts on a gene.

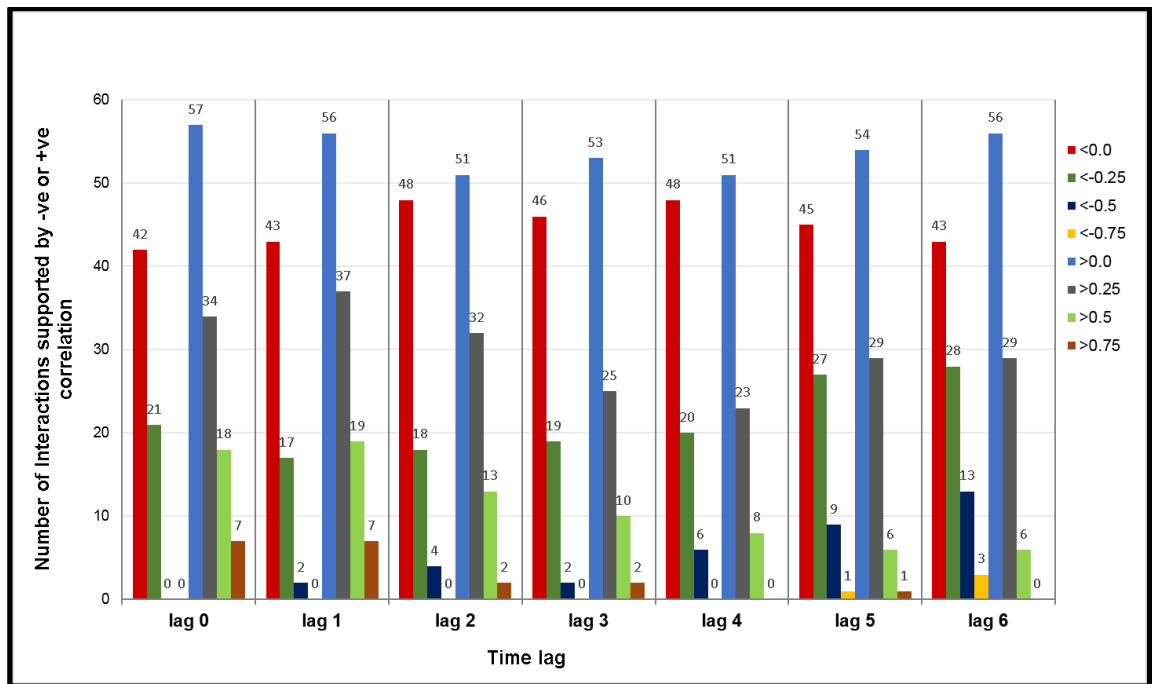


**Figure 3.9** Examples of time-lag correlation of average target genes expression and its corresponding TF(s) gene expression in cluster 1 and 6. Panel A shows the expression patterns of cluster 1 genes (TFs and average of target genes) without lag. Panel B shows the expression patterns of cluster 1 genes (TFs and average of target genes) with 4 lags. Panel C shows the expression patterns of cluster 6 genes (TFs and average of target genes) without time-lag.



**Figure 3.10** Time-lagged correlation starting from 0 time-point lag until 6 time-points lag for TFs co-operation.

TFs co-operation prediction based on normalized AIC transcriptional regulatory networks (52 ‘clear’ clusters). Darker red and blue colours represent high positive ( $r > 0.5$ ) and negative ( $r < -0.5$ ) correlations respectively.

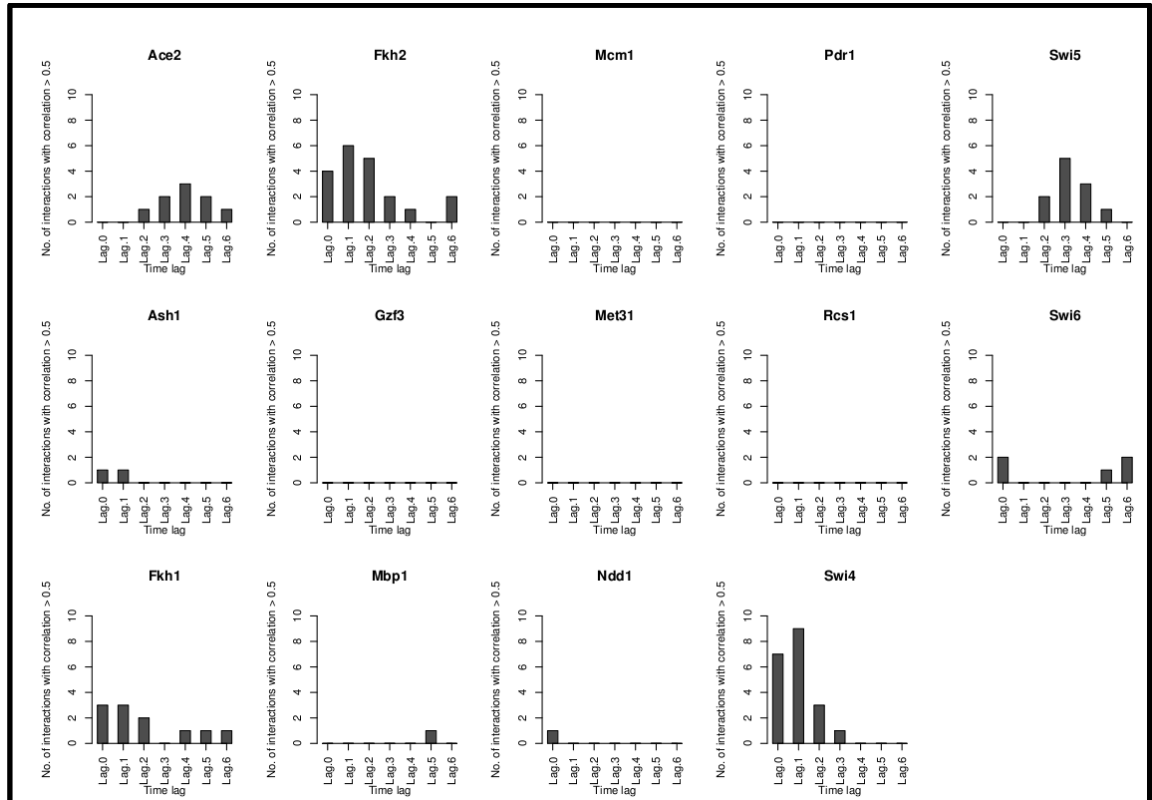


**Figure 3.11** Number of TF-Gene cluster interactions supported by time-lag correlation.

The TF-Gene cluster interactions were between TF gene expression and average target genes expression in a cluster and its correlation (positively or negatively correlated) was calculated at time lag,  $t = 0$  to  $t = 6$ .

If we were to build a transcriptional regulatory network of cell cycle genes based on the assumption of co-expression of TF gene with their target gene(s), only 18 interactions between TF and cluster are supported by a positive correlation (rows within yellow box in Figure 3.10) at  $t=0$  or no time-lag/shift. This could basically inform us that utilizing co-expression of TF with target genes, which is a commonly used method, is not enough to infer transcriptional regulatory interaction. However, as we shifted the expression patterns from 1 time-point lag to 5 time-points lag, one time-point at a time, more regulatory interactions become apparent (see Figure 3.10). We did not use time-lagged information to dictate our gene regulations as it could suffer from the fact that the mRNA transcript could be regulated post-transcriptionally and its protein could be regulated post-translationally. However, it is noteworthy to see some combinatorial regulations found from our regulatory networks are also co-expressed with positive correlation in at least at one of the 5 time-point lags with their target genes expression. Interestingly, novel TF-TF interaction with no protein-protein interaction support, for example, GZF3/PDR1/SWI5 in cluster 1 (see Figure 3.10) is moderately supported by the time-lagged correlation at the 4th time-lagged point. Another example is from a quadruplet combinatorial regulation such as cluster 27, ACE2/SWI5/FKH1/FKH2 co-regulates the genes and their genes are expressed in time-lagged manner with cluster 27 genes. This hypothesis is supported by genetic interaction but not from the known protein-protein

physical interaction (refer to Figure 3.6). This is novel where none of the existing regulation prediction algorithms have found this quadruplet co-regulation. We also discovered that some TFs are substantially time-lag correlated with their targets at a specific lag. For example, Fkh1 Fkh2 and Swi4 for quite a number of times, tend to lag at the first 3 time-lags, while Ace2 and Swi5 at the latter time-lags as shown in Figure 3.12 below.



**Figure 3.12** Distributions of number of interactions with positive time-lag correlation ( $r > 0.5$ ) for all 14 TFs.

Finally we restate that of the regulatory interactions predicted between TFs and genes within our clusters, only 18% are supported by significant correlation between those genes' expression patterns and the expression patterns of the regulating factors. Although this percentage increases if correlations off-set in time are considered, it shows that simple correlation of expression is not a good way of predicting regulation.

### **3.4. Discussion**

By applying our algorithm to genetic regulation in yeast, we note that the method produces results that to a large extent recapitulate existing knowledge. We chose to compare to LeTICE as a recent method based on a similar underlying premise but otherwise the methodology is very distinct. In this study, we have inferred the transcriptional regulatory networks (TRNs) of genes based on meta-heuristic joint probabilistic model of gene regulatory input and gene transcriptional output. The advantages of our method and in comparison to LeTICE are, (1) It provides maximum automation of the clustering procedure without the need for user-defined minimal cluster size, (2) our method has the ability to refine the clusters found by SA further using EM. As a result, we could also identify genes that share more than one cluster. In this study, genes could share clusters of the same phase or adjacent phase. However, this requires the model parameters to be near the global optimal as the EM refinement step is local in nature and could not be performed beyond the maximum likelihood SA itself, (3) our method has the capability to infer a high degree of combinatorial regulation. We have shown earlier that we can find triplet and quadruplet combinatorial TF binding. Some examples are supported by evidence from the literature (i.e. Mbp1/Swi4/Swi6) and PPIs database and some are not (i.e. Pdr1/Gzf3). In addition, some combinatorial interactions also appeared to be supported by time-lag correlation between them and the expression profiles of genes that they regulate, (4) our method could fit the model parameters according to the information criteria of choice, where stringent IC will result in fewer clusters with larger cluster size. On the other hand, the less stringent IC will result in more clusters with smaller cluster sizes.

Last, our method produced arguably better results than LeTICE and our method performs at least as well in terms of biological relevance of the clusters found and in recovering known relevant cell cycle TFs. In this application we suggest that limitations to some degree are associated with the limited nature of the data. The transcription factor binding data is not resolved by time or cell cycle phase, and this limits how well any method could perform.

### **3.5. Conclusion**

In this yeast test case, our method was able to generate clusters with clear biological meaning and suggest transcriptional regulatory networks for yeast cell cycle using the discovered clusters.

## Chapter 4. Application of model-based joint clustering to cancer data

### 4.1. Introduction

Cancer is a class of different diseases that affect the population and is often characterized by abnormal cells that divide in an uncontrolled way and invade healthy cells in the body. It is driven by many different and complex molecular mechanisms that still need to be discovered. The commonly studied molecular data types include but are not limited to gene and protein expression, copy number variations, DNA methylation, and gene mutations. These multiple data types are usually evaluated independently and this leads to an increase in the number of independent features that need to be computationally analysed. Each independent molecular feature represents an incomplete view of biological processes. Thus, in this work, we proposed an approach that can integrate multiple data types of binary and continuous nature in order to model the cancer gene drivers and pathways they might be affecting.

In tumour cells, there are genes that can potentially cause cancer and they are known as oncogenes. Oncogenes are often mutated and abnormally expressed in tumour cells. An example of oncogene group is the tumour suppressors. Tumour suppressor genes encode proteins that protect a cell from becoming cancerous by rapidly responding to diverse cellular stresses to regulate expression of target genes (e.g. cell cycle arrest, apoptosis, senescence, DNA repair, metabolic changes). Somatic mutations in driver genes functioning in important events of gene expression (i.e. signalling molecules/pathways, transcription factors, epigenetic modifiers) can cause changes in gene expression and hence, altered cell phenotypes. Changes in gene expression patterns can be driven by all the processes above and therefore it is interesting biologically to find cancer subgroups defined by characterised patterns of mutations and gene expression. Furthermore, it is important also to be able to distinguish driver gene mutations from passenger gene mutations. This problem is usually tackled by finding the recurrently mutated genes across samples. We are motivated to see if our method could delineate the driver gene mutations from passenger gene mutations. In addition, we want to investigate whether gene mutations together with aberrant gene expressions might or might not be able to group patients into prognostically relevant cancer subtypes.

To demonstrate the applicability of our method on cancer data, we have chosen Acute Myeloid Leukaemia (AML) as our test case. The main classification systems of AML has always been via French-American-British (FAB) [117-119], which largely relies on cell histopathology, and from the World Health Organization (WHO) [120] classification

system which mainly deals with cytogenetic aberrations in order to group AML into subtypes. Recently, more efforts have been placed on finding molecular markers of AML to further characterize and refine AML subtypes. This includes using gene expression information alone [121-123], gene mutations [124] and linking gene mutations to expression [125].

Patients with AML can be divided into subclasses (M0–M7) on the basis of morphology, and quantification of myeloblasts and erythroblasts. FAB consortium differentiated different groups of leukaemia based on the number of healthy blood cells, the size and number of leukaemia cells, the changes that appear in the chromosomes of the leukaemia cells and, the degree of cellular differentiation [119]. However, it was uncertain of whether FAB subtypes added prognostic information. The FAB classes include:

- M0 : Undifferentiated acute myeloblastic leukaemia
- M1 : Acute myeloblastic leukaemia with minimal maturation
- M2 : Acute myeloblastic leukaemia with maturation
- M3 : Acute promyelocytic leukaemia
- M4 : Acute myelomonocytic leukaemia
- M5: Acute monocytic leukaemia
- M6: Acute erythroid leukaemia
- M7: Acute megakaryocytic leukaemia

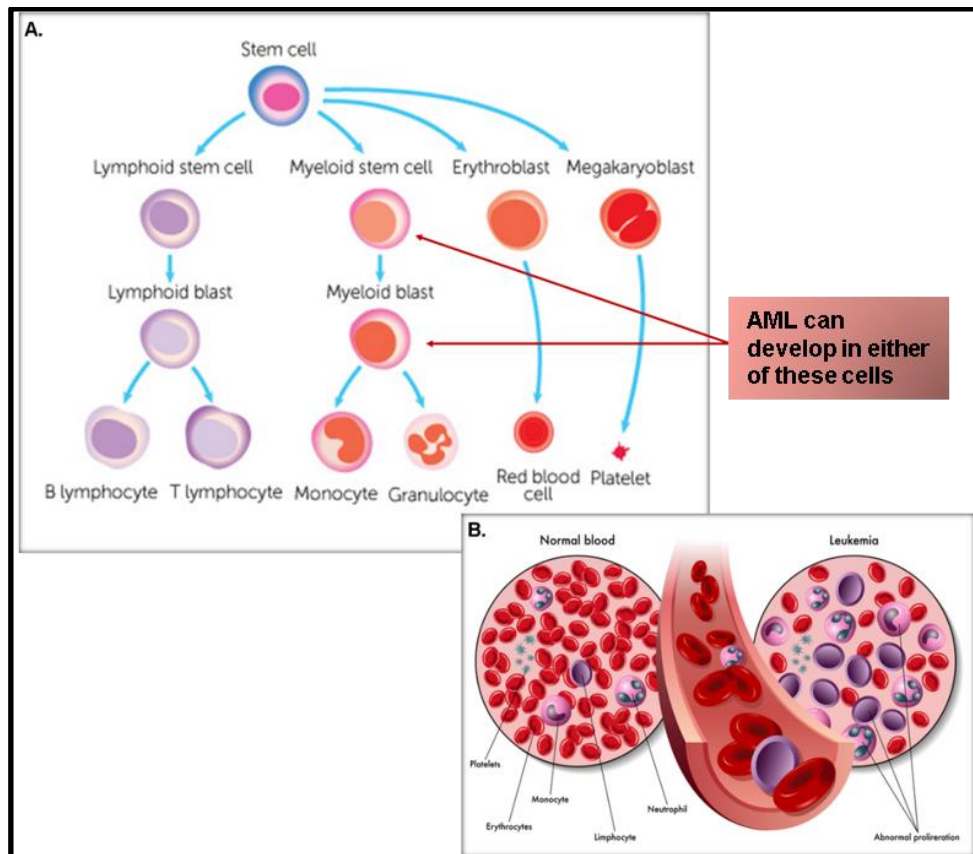


After decades of using FAB classifications, clinicians have now started to switch to the WHO classification as this is deemed to be more prognostically relevant with the recognition of cytogenetic diversity together with other molecular abnormalities. The WHO classification integrates genetic, immunophenotypic, biological, and clinical features in classifying patients/samples. However, for the not otherwise specified (NOS) cases which includes cases that do not belong to any of the classes, FAB is still used. However, Walter and co-workers (2013) have found that this is a flaw where FAB sub classification of NOS cases does not provide prognostic information and suggest to further use mutation related information (i.e. NPM1 and CEBPA) to improve the classification [126]. As a result, additional entities have been added to the WHO classifications:

- AML with recurrent genetic abnormalities:
  - AML with translocation of RUNX1/RUNX1T1
  - AML with translocation of CBEB/MYH11
  - Acute promyelocytic leukaemia (APL) with translocation of PML/RARA
  - AML with translocation of MLLT3/MLL
  - AML with translocation of DEK/NUP214
  - AML with translocation of RPN1/EVI1
  - AML (megakaryoblastic) with translocation of RBM15/MKL1
  - Provisional entity: AML with mutated NPM1
  - Provisional entity: AML with mutated CEBPA
- AML with myelodysplasia-related change
- Therapy-related neoplasm
- AML, not otherwise specified (NOS):
  - M0: Undifferentiated acute myeloblastic leukaemia
  - M1: Acute myeloblastic leukaemia with minimal maturation
  - M2: Acute myeloblastic leukaemia with maturation
  - M3: Acute promyelocytic leukaemia
  - M4: Acute myelomonocytic leukaemia
  - M5: Acute monocytic leukaemia
  - M6: Acute erythroid leukaemia
  - M7: Acute megakaryocytic leukaemia
  - Acute basophilic leukaemia
  - Acute panmyelosis with myelofibrosis

### 4.1.1. How AML develops

AML is a blood cancer that results from faulty haematopoiesis where bone marrow produces subnormal white blood cells. Acute myeloid leukaemia cells develop and proliferate very quickly and on entry into the bloodstream are circulated around the body. In AML, the haemopoietic stem cells of the bone marrow fail to become fully differentiated into white blood cells, red blood cells or platelets, and instead lead to clonal expansion of undifferentiated myeloid (immature white blood cells) also called myeloid blasts, as shown in Figure 4.1A above. The body has no use for these cells so they continue to build up. They will accumulate in the blood and leave little space for normal blood cells to grow and develop as illustrated in Figure 4.1B. The low count of blood cells may lead to anaemia, infection or bleeding. Given the importance of timely regulation of white blood cell differentiation, studying the aberrations at the molecular level which affect this is important for improving patient survival/prognosis and disease prevention.



**Figure 4.1** **A.** Basic normal blood cell development in bone marrow and **B.** abnormal blood cell production which leads to AML.

#### 4.1.2. Molecular aberrations of AML

AML usually develops from cells with somatically acquired driver mutations (e.g. in FLT3, NPM1, CEBPA, KIT, N-RAS, MLL, WT1, IDH1/2, TET2, DNMT3A, and ASXL1) as well as other cytogenetic aberrations (i.e. translocations, inversions) to drive prognosis. Patient prioritization for ideal treatment after disease diagnosis can be improved by using integrated analyses of the co-occurrence of mutated genes and chromosomal aberrations [127].

It has been suggested that AML arises from three complementary classes of mutations: class I (tyrosine kinases-FLT3, KIT, JAK1, JAK3, RAS pathway- NRAS, KRAS; Protein phosphatases-PTPTN11; Ubiquitin pathway-CBL), class II which contains transcription factors genes (RUNX1, PML/RARA, CBFβ/MYH11, GATA2, DEK/NUP214, CEBPA, PU1, MLL fusion, NPM1), a newly formed class III with genes associated with DNA methylation (TET2, IDH1, IDH2, DNMT3A, ASXL1, EZH2) and other class associated with tumour suppressor (WT1 and TP53) mutations [127, 128]. It has been shown that genetic heterogeneity in AML is not random. There is a risk stratification according to genetic heterogeneity [129].

The good risk group includes recurrent translocations such as PML/RARA, MYH11/CBFβ and RUNX1/RUNX1T1. The high-risk group includes patients with the DEK/NUP214 and RPN1-EVI1 translocations. Often, AML patients with complex karyotypes (more than 3 chromosomal aberrations) are also in the high risk group and AML with normal cytogenetics are in the intermediate risk group [129]. In terms of somatic mutations, the poor prognosis group is also associated with DNMT3A, MLL and all the class III mutations. On the contrary, patients with class II mutations usually have better outcomes than class I and III [127].

Molecular classification using gene expression profiling was able to sub-classify patients into different groups and resulted in sub classes with clear favourable and unfavourable prognosis [121-123]. These discoveries of driver mutations and gene expressions provide insight into the biological details of AML, but how they both contributed cohesively on patient's prognosis are unclear. More data has emerged indicating that gene mutations and gene expressions may be useful in patient prioritization and/or selection for optimal AML therapy [130, 131].

## 4.2. Methodology

To test the application of our methodology to data from cancer samples, we applied it to the Acute Myeloid Leukaemia (AML) mutation and gene expression data generated by The Cancer Genome Atlas (TCGA) Research Network [42, 132]. We downloaded RNA-seq and mutations data from TCGA AML cohort; data were available for 200 patients with 314 gene mutations. However, these data were subjected to pre-processing which is explained in this section.

### 4.2.1. Selecting mutated genes and variably expressed genes

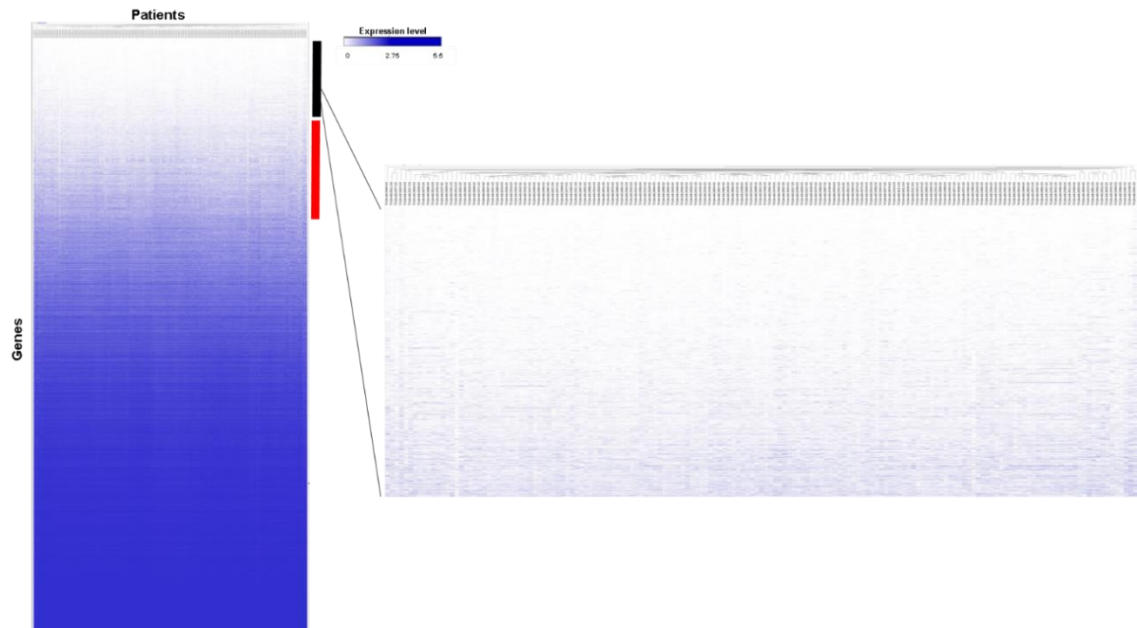
Datasets from TCGA were retrieved through using cBioPortal for Cancer Genomics tool [133, 134]. In accordance with the TCGA data usage guidelines and policy, samples were selected based on the availability of mutation and RNA-seq gene expression data. For gene mutation, genes which were mutated in at least two patients were chosen and samples with no mutation were removed, resulting in 170 samples and 154 gene mutations.

For gene expression, we first normalized the data set to remove systematic variation between different experiments or batch effects using quantile normalization. Quantile normalization is used widely to make the distributions the same across samples. We then chose to do an exploratory approach of data pre-selection by selecting the most variably expressed genes similar to what has been done in this field previously [135, 136]. The less variable genes are less informative in our study where we required genes that would contribute to the different classes of AML patients.

We used coefficient of variation,  $cv$  and standard deviation to represent variation between samples for each gene. Coefficient of variation was calculated using:

$$cv = \frac{\sigma}{\mu} \cdot 100$$

Here,  $\sigma$ ,  $\mu$  are the standard deviation and mean and of gene expression across samples/patients, respectively. The higher the  $cv$ , the greater the level of dispersion/inconsistency around the mean. Following calculation of  $cv$  for all genes, genes were then ranked from highest to lowest  $cv$ . However, genes with higher  $cv$  in the black region (see Figure 4.2) have lower expression values compared to the intermediate region. Genes with higher variable with subsequent high expression are preferable. In our case, we use rank order by sorting on rank of standard deviations (largest to smallest) for the top 10,000 genes previously sorted using  $cv$ . By doing this, genes with higher expression values and variably expressed will be at the top of the list.



**Figure 4.2** Gene expression patterns for AML patients.

Columns are the 170 AML patients and rows are 20,502 genes ranked from highest to lowest coefficient of variation across patients. The region in black shows genes with higher standard deviation and lower mean or in other words, higher  $cv$ . The region in red shows genes with intermediate  $cv$  and favorable to be used with our clustering where the expression values are slightly higher.

We chose the top 500 genes with highest ranked-based coefficients of variation and standard deviation across these samples (details of samples, mutations and chosen genes are given in Table 4.1 below).

**Table 4.1** Input data points and variables for AML dataset.

Input data consist of 170 samples, 154 mutated genes and 500 most variably expressed genes.

\* Genes which are mutated in at least 2 samples. \*\* Genes which are variably expressed based on ranked coefficient of variation and standard deviation.

Sample (AML patients)	Mutated genes*	Variably expressed genes**
TCGA-AB-2948-03, TCGA-AB-2853-03, TCGA-AB-2942-03	FRYL BMPER OR13H1 PHACTR1	UBE2V1 AHSP SH3BP4 GLT1D1 MYCT1 FAM127A ZFY NPR3 CSMD1 RNF217 GABRE
TCGA-AB-2909-03, TCGA-AB-2970-03, TCGA-AB-2825-03	PRPF8 GLTSCR1L CACNA2D3 PDCD2L	CYorf15B JPH1 BAALC C2orf54 DOCK1 EPCAM ELN LIFR MYOF LUM EPB42
TCGA-AB-3000-03, TCGA-AB-2985-03, TCGA-AB-2987-03	SEMA4A ARAP2 TUBA3C GJB3	UTY CT45A5 ASS1 CCNA1 GPR12 HOXB8 KIAA0087 HMGA2 DUSP27 PTPN20B AREG
TCGA-AB-2890-03, TCGA-AB-2980-03, TCGA-AB-2857-03	PPP1R3A MED12 RYR3 FLG	VENTX ZNF711 FERMT1 LHX6 TM4SF1 CYP1B1 NPTX2 PDK4 UNC13B NAPS B GPR126
TCGA-AB-3008-03, TCGA-AB-2862-03, TCGA-AB-2943-03	CACNA1E KMT2C PLEKHH1 PSME4	SNCAIP CD200 TOM1L1 VCAM1 SLC28A3 SLC05A1 NLRP2 FBLN1 C5orf23 DTNA RFP1L1S
TCGA-AB-2866-03, TCGA-AB-2904-03, TCGA-AB-2982-03	MTMR8 CROCC ZC3H18 TNC	DEFA4 MECOM CLEC7A HTR1F THNSL2 RETN LOC100101938 NRP1 PRLR HBM
TCGA-AB-2835-03, TCGA-AB-2863-03, TCGA-AB-2859-03	LRRC37B P2RY2 NUP98 APOB	SEMA3C PTGER3 ITGB3 VGLL3 PPARGC1A EPX UGGT2 IRX5 LTBP1 FGD5 MEIS1
TCGA-AB-2933-03, TCGA-AB-2879-03, TCGA-AB-3007-03	ATP1B4 CADPS ZNF687 TTBK1	HBB HDC IRX1 FBN2 ANO7 HOXB3 LDLRAD3 MYCL1 IRX3 SLC24A3 PHACTR3
TCGA-AB-2837-03, TCGA-AB-2963-03, TCGA-AB-2860-03	FCGBP NMUR2 GATA2 DIS3	TRPM4 NKX2-3 TMSB4Y SLITRK4
TCGA-AB-2995-03, TCGA-AB-2869-03, TCGA-AB-2988-03	DDX41 SAXO2 NRXN3 CALR	CLC EPHB2 S100A12 PF4 PPP1R9A TPSAB1 NUDT10 FCN1 MS4A2 DEFA1B MEG3
TCGA-AB-2944-03, TCGA-AB-2818-03, TCGA-AB-2916-03	MEFV E2F8 GRM3 PRAMEF2	CA1 COL5A1 PLCB4 MS4A3 SERPINA1 COL23A1 DKK2 CNNM1 PHKA1 S100A16 CPNE8
TCGA-AB-2847-03, TCGA-AB-2931-03, TCGA-AB-2981-03	SCARB1 SCN1A DNMT3B PICALM	GYPA ANK1 CD109 CLEC9A MRC1 TMIGD2 GYPB TPSD1 HOXA7 IGLL1 LGSN
TCGA-AB-2956-03, TCGA-AB-2881-03, TCGA-AB-2816-03	FOXP1 GSTK1 DOCK2 SMG1	C5orf20 CXCL12 LILRA6 HOXA6 LAMC1 DNNT CPNE7 PRRG1 LILRA5 HOXA5 PAX8
TCGA-AB-2977-03, TCGA-AB-2813-03, TCGA-AB-2929-03	DCLK1 CUL1 SETBP1 MUC16	RXFP1 LOC441666 LILRA3 HOXA4 SCUBE1 SNORD116-4 SIGLEC9 PRR16 LIN7A HOXA3
TCGA-AB-2992-03, TCGA-AB-2927-03, TCGA-AB-2964-03	PKHD1L1 RNF213 THRAP3 SPEN	FLJ22536 NDST3 OPALIN GSTM1 SYCP2L COBL C-Yorf15A DSC2 BEX1 APOC2 TUSC1
TCGA-AB-2954-03, TCGA-AB-2884-03, TCGA-AB-2901-03	C17ORF97 ASXL1 EPPK1 NF1	HOXA9 PROK2 NTRK1 MYCN LOC284551 C5AR1 PTK2 ADAMTS2 NAV3 POU4F1 ITGA9
TCGA-AB-2848-03, TCGA-AB-2998-03, TCGA-AB-2811-03	UNC5B GBP4 ADGRG4 NSD1	PKP2 STAB1 HPGDS MPEG1 AR CD1D CHI3L1 BPI GPR173 DTRD9 GPC4
TCGA-AB-2815-03, TCGA-AB-2843-03, TCGA-AB-2959-03	ZBTB33 CACNA1B ATP10B COL12A1	CD1E FBLN2 SCN9A VAT1L DDX3Y GPC6 CD1C CYP4F2 CLGN C17orf55 AOX2P
TCGA-AB-2880-03, TCGA-AB-2971-03, TCGA-AB-2855-03	TRPM3 BCR DCHS2 GRID1	DACH1 EVC MS4A4A COL3A1 MDFI SPAG6 CLTCL1 MYO18B ADAMTS1 PRKY PTGFR
TCGA-AB-2973-03, TCGA-AB-2928-03, TCGA-AB-2967-03	OR11H12 PTPRT CDK11B TFG	HK3 SHANK1 PTGDS ADAMTS3 VSTM1 H2AFY2 ANXA8L2 CD163 WDE DHR59 LPHN3
TCGA-AB-2836-03, TCGA-AB-2821-03, TCGA-AB-2870-03	FAM57B EZH2 LRBA TMEM255B	EIF1AY NEGR1 PTPRG VPREB1 SAGE1 SCARA3 CYP2S1 HTR7 PTPRD TNNT1 GNG11
TCGA-AB-2845-03, TCGA-AB-2840-03, TCGA-AB-2972-03	ETV6 SUZ12 DDR2 GAS6	MTMR11 ADAMTS18 UMODL1 ZFP57 LILRB4 ADRA2C ZNF727 BCORL2 TSPAN7 PTRF LILRB2
TCGA-AB-2832-03, TCGA-AB-2949-03, TCGA-AB-2937-03	DNAH9 ADGRG7 HECW1 PLCE1	CDH9 HOXA11AS KRT8 CACNA2D3 ALOX15B NXF3 CDH2 SCN2A FAM38B SERPINB2
TCGA-AB-2856-03, TCGA-AB-2810-03, TCGA-AB-2849-03	HNRNP K ABL1 TOP3B ELL	PTPRM SERPINB10 CDH4 DNAJC12 HOXA11 CALN1 MYO7A CCL23 DPP10 PACSIN1
TCGA-AB-2841-03, TCGA-AB-2965-03, TCGA-AB-2865-03	CD74 RAD21 GIGYF2 BRINP3	HOXA10 BMP3 SIGLEC1 HNMT IL131RA XIST PPBP ROBO1 C8orf79 SDK2 RAMP1
TCGA-AB-2911-03, TCGA-AB-2861-03, TCGA-AB-2806-03	CSMD3 TTN CSMD1 STAG2	C3orf50 STOX2 ROBO4 C2orf200 HOXB9 PTH2R MS4A6A DSG2 HBG1 TEX15
TCGA-AB-2842-03, TCGA-AB-2930-03, TCGA-AB-2925-03	MUC5B PHF6 KCNA4 MLLT10	HOXB2 CYR11 HBA1 C2 MEFV KCNK17 LPO HBA2 SLC8A3 TMEM189- AADAT
TCGA-AB-2823-03, TCGA-AB-3002-03, TCGA-AB-2887-03	STRIP2 KRAS MECOM KIT	HOXB6 PROM1 PBX1 C7 L3MBTL4 UGT2B11 HOXB7 WIT1 CYP7B1 GPR85 KDM5D
TCGA-AB-3011-03, TCGA-AB-2914-03, TCGA-AB-2918-03	KDM3B SMC1A BSN SMC3	ELANE HOXB4 APP CLEC4E EPB41L3 DLK1 TTY15 HOXB5 C1QC CLEC4D MYEF2
TCGA-AB-2991-03, TCGA-AB-2969-03, TCGA-AB-3006-03	KDM6A KMT2A PHIP U2AF1	TLR8 KIAA1462 PPARG C1QB CLEC4C NDN SCN3A PRSS21 COL4A5 C1QA ASGR2
TCGA-AB-2896-03, TCGA-AB-2913-03, TCGA-AB-2826-03	DLC1 PTPN11 RIMS1 RUNX1T1	DDIT4L PXDN SLC44A5 TSIX AZU1 C2orf103 LGALS2 CT45A1 FGF13 C10orf114 KIF17
TCGA-AB-2838-03, TCGA-AB-2897-03, TCGA-AB-2814-03	KDR MYH11 ILDR1 WT1	MOSC2 PAWR CT45A3 APBA1 TCN1 WNT7B MOSC1 FAM171A1 MAMDC2 CCDC48
TCGA-AB-2941-03, TCGA-AB-2891-03, TCGA-AB-2934-03	TET1 CBF1 WAC NRAS	LOC654433 VLDLR MMP8 SLPI HBG2 PTPN14 KRT17 TBC1D3G MMP9 AIF1L
TCGA-AB-2986-03, TCGA-AB-2828-03, TCGA-AB-2803-03	TCEAL6 TP53 SCAF8 CEBPA	DPPA4 ZNHIT2 CD14 UGT3A2 TGFB1 LRP1 TIFAB LOC644172 SORT1 SPON1
TCGA-AB-2990-03, TCGA-AB-2844-03, TCGA-AB-2819-03	SEMA3A TET2 CADM2 RARA	MMP2 LRP6 IL1R2 KIAA1598 FOXC1 SCHIP1 ANXA8 CDA ST18 TMEM136 KRT18
TCGA-AB-2820-03, TCGA-AB-2978-03, TCGA-AB-2808-03	GRIK2 PML GRIK4 IDH1	GTSF1 CRLF2 PRRT4 PPNAN-P2RY11 SECTM1 SECTM1 SECTM1 SECTM1
TCGA-AB-2921-03, TCGA-AB-2858-03, TCGA-AB-2983-03	GALNT18 IDH2 SI RUNX1	PI15 NCRNA00185 ARHGEF10L CD34 COL1A2 FAT1 HPGD MKRN3 KIAA1324L
TCGA-AB-2867-03, TCGA-AB-2999-03, TCGA-AB-2888-03	CNTNAP4 DNMT3A CMYA5 FLT3	BGN EREG IGSF10 LTK SLC4A1 MARCO MYL4 COL1A1 THBS1 ZNF334
TCGA-AB-2955-03, TCGA-AB-2919-03, TCGA-AB-2996-03	MAP2 NPM1	SHROOM4 CDC42BPA CLEC14A KIAA1217 PCBP3 ADCY2 COL2A1 CEACAM6 C7orf58 CDH11 TRH
TCGA-AB-2898-03, TCGA-AB-2846-03, TCGA-AB-2854-03		SPINK2 HTRA3 TACSD2 RANBP17 TRIM71 TRO KCNE1L CLEC5A CEACAM8 IL1RL1 KIRREL
TCGA-AB-2935-03, TCGA-AB-2976-03, TCGA-AB-3001-03		ENPEP LTF RPS4Y1 BMX DCN NLRP12 MN1 CLEC10A PRAME VNN3 SELENBP1
TCGA-AB-2979-03, TCGA-AB-2966-03, TCGA-AB-2908-03		S100P THSD7A TMEM105 PTX4 VNN1 ZNF521 FCGR3B EVPL ACY3 ALAS2 IGFBP2
TCGA-AB-2932-03, TCGA-AB-2885-03, TCGA-AB-2872-03		DLGAP2 LOXHD1 C10orf140 DEFB1 S100A9 SHD FAM110B USP9Y PRDM16 MXRA5 S100A8
TCGA-AB-2824-03, TCGA-AB-2868-03, TCGA-AB-2917-03		TCGA-AB-2924-03, TCGA-AB-2912-03, TCGA-AB-2874-03
TCGA-AB-2939-03, TCGA-AB-2952-03, TCGA-AB-2875-03		TCGA-AB-2830-03, TCGA-AB-2993-03, TCGA-AB-2915-03
TCGA-AB-2924-03, TCGA-AB-2912-03, TCGA-AB-2874-03		TCGA-AB-2936-03, TCGA-AB-2817-03, TCGA-AB-3009-03
TCGA-AB-2830-03, TCGA-AB-2993-03, TCGA-AB-2915-03		TCGA-AB-2834-03, TCGA-AB-2807-03, TCGA-AB-2920-03
TCGA-AB-2936-03, TCGA-AB-2817-03, TCGA-AB-3009-03		TCGA-AB-3005-03, TCGA-AB-2900-03, TCGA-AB-2871-03
TCGA-AB-2834-03, TCGA-AB-2807-03, TCGA-AB-2920-03		TCGA-AB-2984-03, TCGA-AB-2822-03
TCGA-AB-3005-03, TCGA-AB-2900-03, TCGA-AB-2871-03		
TCGA-AB-2984-03, TCGA-AB-2822-03		

## 4.2.2. Clusters analysis

As the classes are unknown beforehand, using our un-supervised clustering, we were able to discover clusters which we analysed separately in terms of biological and statistical significances.

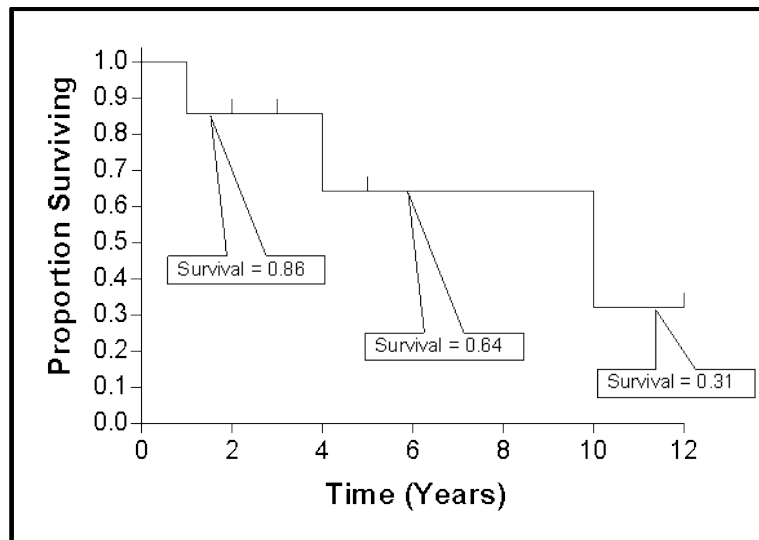
### 4.2.2.1. Kaplan-Meier survival estimate

One way of evaluating the clusters found is by comparing treatment response, or the survival probability (i.e. that an individual survives since the beginning of diagnosis time), between clusters. It is sometimes the case that patients in different subtypes have distinct survival probability patterns and this can prove to be clinically relevant. To be able to segregate patients into subtypes with distinct survivals is the main goal in a patient's treatment, even to the extent of implementing personalized cancer therapy.

In cancer study, an important measure or event of interest is the time from cancer diagnosis and/or surviving a treatment until recurrence or relapse-free/disease-free. In the TCGA AML case, the patients' recorded clinical information is the time period between first diagnosis and the months of surviving the cancer. There are different clinical observations of patients: (1) still alive and disease free (2) still alive but with recurred/progressed of disease (3) dead after having recurred/progressed disease (4) disease free but dead probably due to old age. However, with the status of survival –living or dead, and overall survival- months after diagnosis, we could use these information to predict the survival of patient from the time of diagnosis. The survival probability  $S(t)$  is the probability that an individual survives from the time of diagnosis to a specified future time  $t$  [137].

$$s(t_j) = s(t_{j-1}) \left(1 - \frac{d_j}{n_j}\right)$$

Here,  $s(t_j)$  is the probability of being alive at time  $t_j$  and this can be calculated from the probability of being alive at  $t_{j-1}$ .  $n_j$  and  $d_j$  are the number of patients alive at  $t_j$  and the number of death of event at  $t_j$  respectively. The term event here refers to death and the probability of surviving from one interval to the next,  $j = 1, \dots, k$  can be multiplied together and cumulatively build up the survival probability. Starting at  $t_0 = 0$  and with  $s(0) = 1$ , the probability of survival for patients would simply reduce to the ratio  $\left(1 - \frac{d_j}{n_j}\right)$ . The probabilities of survival,  $s(t_j)$  across time intervals,  $t_j$  are usually represented using the Kaplan-Meier survival plot which contains survival curves of probability/proportion surviving versus time (i.e. months or years).



**Figure 4.3** An example of the Kaplan-Meier survival plot.

At the beginning (Time =0) all patients (i.e. seven patients) were alive and still in the curve. During the interval (i.e. 1-4 years), a patient was dead so that at the end of this interval, 6 patients were still at risk (i.e. proportion surviving this interval is 6/7 or 0.86). For the following intervals, the proportion/probability of surviving is calculated cumulatively (i.e. proportion surviving at interval 4-10 multiply by the proportion surviving 1-4).

Comparing between survival curves of two or more groups can be done using log-rank test,  $\chi^2$  with the corresponding null hypothesis, that there is no difference between population/clusters survival curves or the probability of an event occurring at any time point is the same for each population/cluster,  $i$ :

$$\chi^2 = \sum_{i=1}^c \frac{(O_i - E_i)^2}{E_i}$$

Here,  $C$  is the number of cluster,  $O_i$  is the – counts of number of observed events in cluster  $i$ , and  $E_i$  is the counts of number of expected events. This value then is compared to a  $\chi^2$  distribution with  $(C - 1)$  degree of freedom and p-value may be calculated following this [137, 138].

#### 4.2.2.2. Comparative marker selection tool

Finding a set of features/markers which can discriminate between distinct clusters of patients is an intuitive approach where genes which are differentially up and down regulated in each cluster might dictate the characteristics and prognosis of patient's cancer. A comparative marker selection (ComparativeMarkerSelection(v10) ) tool in the GenePattern module provided by the Broad Institute is useful for this exercise [139]. This tool uses a test statistic,



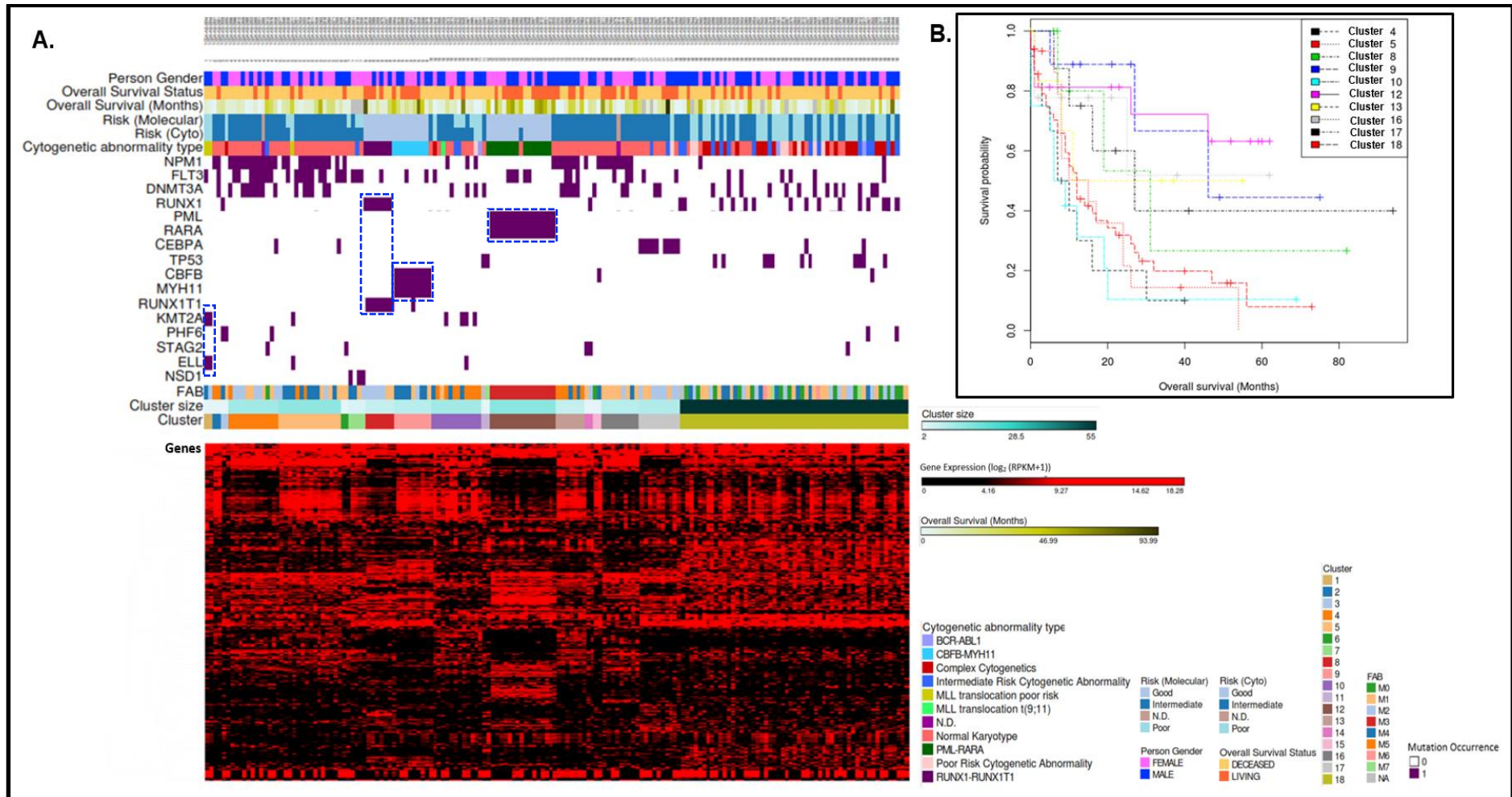
t-test to evaluate the differential expression between two classes of samples. However, in this case where we have multiple phenotypes/clusters, one-versus-all comparisons were done for all 500 genes in each cluster. Up and down regulated genes which are significantly different between clusters (i.e. genes with FDR (Benjamini-Hochberg corrected) less than 0.005 and with  $-10.0 < \text{score} > 10.0$ ) were selected.

#### **4.2.2.3. DAVID functional analysis**

As briefly introduced in the Chapter 1, clustering of genes is usually subjected to a systematic evaluation to see if genes in the same group are biologically related. We would like to explore the biological meaning of the markers in each group in term of its molecular/biological processes and pathways. Furthermore, if they are to some degree correlated with prognosis of patient's survival, this could explain why patients carrying this marker tend to survive for longer or shorter times. There are a lot of tools available for the discovery of enriched Gene Ontology (GO) terms, for example, Database for Annotation, Visualization, and Integrated Discovery (DAVID) [140] , GOrilla [141] and Gene Set Enrichment Analysis (GSEA) [142]. These tools statistically emphasize on the most enriched GO terms including but not limited to biological processes, molecular functions and cellular locations given lists of genes. Here, in our study, we used the most frequently used tool, DAVID which accepts as input lists of genes correspond to our clusters.

### **4.3. Results and discussion**

Based on our findings with simulated data, we investigated clustering of this mutation and expression data using the AIC related criteria with  $\lambda = 2.0$  and  $2.5$ . Again clustering with this real data set showed greater variability in results between these two penalty functions than was evident in simulations, with clusters predominantly very small (two samples per cluster) from  $\lambda = 2.0$  and would be problematic for cluster validation later (e.g. survival probability analysis). Accordingly, we chose  $\lambda = 2.5$  in this case on biological and statistical grounds. Figure 4.4 below shows a graphical representation of 18 clusters found by clustering patients according to their molecular signatures (i.e. gene mutational status and gene expression) and the overall survival curves for clusters. Following the 18 clusters found by using AIC ( $\lambda = 2.5$ ), the most differentially expressed genes in each cluster in comparison to the rest of the clusters were selected using the one-versus-all phenotype test in the Gene Pattern tool from [139] and the result is presented in Table 4.2 below.



**Figure 4.4** 18 clusters found from using AIC ( $\lambda = 2.5$ ).

Panel **A.** shows Clustering of AML samples shown in columns of 170 samples using AIC ( $\lambda=2.5$ ) across most variably expressed genes (lower) with 500 genes and the mutated genes (above) with 18 genes coloured in dark purple. Genes with fusion mutation are marked within the blue-dotted boxes. Panel **B.** is Kaplan-Meier estimators for the 10 clusters with more than 2 samples with survival information available in each cluster. The 10 Kaplan-Meier estimators perform differently with a significant p-value in the Log-Rank Test,  $p=0.00133$ .

**Table 4.2** Genes that are significantly differentially expressed between clusters determined using the one-versus-all phenotype test in the Gene Pattern tool from [139].

Genes with FDR (Benjamini-Hochberg corrected) less than 0.005 and with  $-10.0 < \text{score} < 10.0$  were selected. Genes in red are the highly expressed genes whereas genes in blue are less expressed genes. Cluster numbers highlighted in yellow are clusters with relevant mutations (i.e. genes in dark green at the bottom of the list)

1	2		3	4	5	6	7	8		9		
SLITRK5	C1QC	DLK1	DSG2	CYP4F2	PRDM16	HOXA3	COL4A5	CCNA1	RUNX1T1	ZNF711	MSLN	DDIT4L
DLK1	DKK2	ZNHIT2	KIRREL	JPH1	C10orf140	HK3	DNAJC12	LOC399959	TSPAN7	SEMA3C	NRP1	PRDM16
CDH2	MSR1	SNCAIP	ENPEP	CLEC4C	NKX2-3	HOXA4	IGSF10	THSD7A	MPO	NEGR1	CYP2S1	SLITRK4
FLJ42875	LGALS2	ARPP21	MXRA5	COBL	FLJ42875	MAFB	CLEC4C		RFPL1S	DPP10	CLEC10A	FAM38B
DDIT4L	CD300E	COL2A1	VGLL3	ADRA2C	HOXB3	HOXA7	CD1E	NSD1	EVC	KIAA0087	LRP6	HOXA6
ZNF727	C5AR1	NUDT10	FAT1	NKX2-3	HOXB6	HOXA5	SPAG6	(0.75)	HPGDS	DOCK1	RXFP1	CYP7B1
GPR173	C1QB	NCRNA00185	CDH11	COL4A5	COL4A5	HOXA6	TACSTD2		SLCO5A1	ABO	MTMR11	RANBP17
BEND4	LILRB2	BCORL2	CHRDL1	HOXB4	HOXB4	CD300E	ASS1	FLT3	PRRT4	C3orf50	GPR12	FLJ42875
PAWR	PTGFR	TMSB4Y	COL3A1	HOXA5	HOXA5	LILRA6	IL31RA	(0.75)	POU4F1	HOXB4	ST18	COL4A5
FAM127A	HTR7	TTY15	PXDN	HOXA6	HOXA6	CDA	NPTX2		SHANK1	HOXB5	KIF17	
	FPR1	EIF1AY	COL1A2	HOXB2	HOXB2	SERPINA1	DDIT4L		SLC24A3	HTR1F	FBLN2	MYH11
ELL	HNMT	CYorf15A	NDN	HOXB5	HOXB5	HOXA10	DLGAP2		LPO	PHACTR3	PTPRM	(1.0)
(1.0)	LILRA5	THSD7A	DCN	HOXA3	HOXA3	CCL23	PRR16		PTX4	HOXA5	TRIM71	
	SIGLEC9	CYorf15B	BGN	HOXA7	HOXA7	SIGLEC9	TSIX		DNMT	IL31RA	AR	CBFB
KMT2A	MS4A4A	USP9Y	APP	IGSF10	IGSF10	TMEM105	LGALS2		C20orf54	HOXB3	VSTM1	(1.0)
(1.0)	MEFV	CYYR1	COL1A1	MEIS1	MEIS1	CHRDL1	CYP4F2		TRIM71	NKX2-3	TGFBI	
	HK3	KDM5D	HBB	HOXA4	HOXA4	APP	CEACAM6			HOXB6	MARCO	
	LDLRAD3	DDX3Y		HOXA9	HOXA9	KIRREL	S100A8			DTNA	CD1E	
	ARHGEF10L	DLGAP2	NPM1	TOM1L1	TOM1L1		S100A9			MEIS1	CLEC5A	
	SERPINA1		(1.0)	CYP7B1	CYP7B1	FLT3	PRLR			ZNF334	PLBD1	
	MPEG1			TRH	TRH	(0.6)	MYOF			HOXA3	DUSP27	
	MS4A6A		PHF6				BPI			FLJ42875	CD14	
	SIGLEC1		(1.0)		DNMT3A	NPM1		FLT3		CPNE8	GLT1D1	
	KYNU				(0.66)	(0.73)		(1.0)		PPARGC1A	CD1C	
	CPNE8				FLT3					HOXA7		
					(0.66)					HOXA6		
					NPM1							
					(1.0)					RUNX1T1		
										(1.0)		
										RUNX1		
										(1.0)		

**Table 4.2** Genes that are significantly differentially expressed between clusters determined using the one-versus-all phenotype test in the Gene Pattern tool from [139].

Genes with FDR (Benjamini-Hochberg corrected) less than 0.005 and with  $-10.0 < \text{score} < 10.0$  were selected. Genes in red are the highly expressed genes whereas genes in blue are less expressed genes. Cluster numbers highlighted in yellow are clusters with relevant mutations (i.e. genes in dark green at the bottom of the list) **(Continued)**

10	11	12		13		14	15		16		17	18	
ARHGEF10L MS4A6A UGGT2 FAM127A C17orf55 NPR3 DNNT ARPP21 SHD NTRK1 C5orf23 OPALIN HDC UMODL1 UGT2B11	ANK1 MYL4 VWDE PLSCR4 CA1 PTK2 ADAMTS18 ZNHIT2 NKX2-3 GPR12 SORCS1 LPO ACY3 SHANK1  <b>TP53 (1.0)</b>	IL17RE SIX3 FGF13 PCBP3 PTPRG NUDT10 CLTCL1 LPO PRRT4 IGFBP2 S100B ASS1 IRX5 PPARG LTK ELANE GABRE PTGDS LOC399959 MPO STAB1 COL23A1 KCNE1L KRT17 SLC24A3 ANO7 CPA3 SLPI PRRG1 UGT3A2 MOSC2	LIN7A RETN CCNA1 ZNF711 LAMC1 PRODH ANXA8 TDRD9 AZU1 CTSG STOX2 MS4A3 KRT18 EVPL NDST3 BEND6 VSTM1 MEG3 FBN2 PTGER3 SHROOM4 IRX1 TEX15 AR DDIT4L AADAT PROM1 VNN3 C20orf200	SLC8A3 DHRS9 KIAA1598 HOXA11AS LDLRAD3 HOXA4 HOXB2 MN1 HOXB5 PRDM16 HOXB6 HOXA11 CD109 EREG CPNE8 TMEM105 KIAA0087 HOXB3 FLJ42875 NKX2-3 SLITRK5 HOXB4 HTR1F HOXA10 CNNM1  <b>RARA (1.0)</b>  <b>PML (1.0)</b>	HOXB6 PHACTR3 HOXB5 APOC2 CT45A1 SCUBE1 WNT7B NKX2-3 H2AFY2 HOXA11 HOXB3 RPS4Y1 C2 HOXB2 LOC728606 CCL23 HOXB7 HOXA7 HNMT ZFY CPNE8 SORT1	HOXA6 HOXA11AS DDX3Y HOXA3 UTY PRKY KDM5D GTSF1 TLR8 MYO7A OPALIN GPR12 ZNF521 CLEC9A CD34 TMIGD2 TRH  <b>DNMT3A (0.71)</b>  <b>NPM1 (1.0)</b>	MYL4 AHSP COL4A5 DLK1 JPH1 CLEC4C GLI2 OPALIN ALOX15B BCORL2 SCN2A TTY15 EIF1AY CYP7B1 USP9Y RPS4Y1 UTY DDX3Y ZFY DLGAP2 ANXA8 TNNT1 MARCO LPHN3 LGALS2 CLEC10A MEFV  <b>STAG2 (1.0)</b>	MSR1 TMEM176A CD300E MTMR11 TMEM176B PDK4 TLR8 MS4A6A CLEC7A KYNLU DPP10 HOXB9 FGF13 GLI2 VAT1L SNCAIP CDH2 LOC399959 FLJ42875 CDH4 COL2A1 NUDT10 NKX2-3	GABRE FERMT1 GPR85 PTPN20B C20orf200 PTGFR NDST3 DSG2 CSMD1 PAX8 PPARGC1A APOC2 MPO MOSC2 KRT8 PLCB4 LOC654433 BEND6 CYP7B1	NKX2-3 HOXA5 HOXA3 SCARA3 HOXA6 HOXA4 FAM38B LIN7A HOXB3 ZNF521 LTBP1 SLC24A3 HOXA9 HOXB6 HOXA7 HOXB4 CPA3 MMP2 FAM110B MEIS1 HOXA10 HOXB5 MDFI DSC2 WNT7B	TRIM71 SIGLEC1 MPEG1 PTX4 PRLR PTK2 KIF17 OLFML2A MSR1 RETN DUSP27 CACNA2D3  <b>NPM1 (1.0)</b>	UGT2B11 APBA1 HPGDS SHD C8orf79 CYP7B1 PROM1 SLITRK5 TRO GPR173 HOXB6 KIAA0087 CES1 HOXB3 C20orf200 MEIS1 ELN HOXB4 NKX2-3 HOXB5 KRT17 HOXA10 HOXA9 HOXA7 HOXA6  <b>CEBPA (0.9)</b>	APP CD34 MN1 PROM1 FAM171B BAALC GPR173 SDK2 HMGA2 LOC728606 APOC2

It is well known that many genes are recurrently altered in AML including single to multiple genetic mutations and cytogenetic abnormalities such as translocation-mediated fusion events (i.e. PML/RARA, KMT2A/ELL, CBFB/MYH11, and RUNX1/RUNX1T1). Genetic fusion is considered as a type of mutation by Kihara and co-workers (2014) [143] and this is evident in several of the clusters that our method produces (see blue dotted-line boxes in Figure 4.4A). Here, cluster 18 in Figure 4.4A was found to be the largest cluster consisting of 55 patients having distinct patterns of expression but is not associated with any mutation at high probability but associated with TP53-RUNX1-DNMT3A mutations at much lower probabilities (less than 0.5). This cluster is similar to cluster 10 and 15 where both have clear gene expression patterns but do not associate with any mutation with a high probability. This illustrates that this method is sufficiently robust to discover gene expression based clusters without an associated mutational pattern in the dataset. These observations are potentially related to different oncogenic mechanisms other than the known mutations and cluster 15 and 18 show statistically significant differences in survival. The rest of the clusters, with the exception of clusters 10, 15 and 18, mostly have a distinct mutation and expression patterns (see Figure 4.4A above, Table 4.3 and Table 4.4 below). Discussions on the clusters found are presented based on the prognostic properties of clusters as follows:

**Unfavourable clusters (clusters 1, 3, 4, 5, 7, 10, 11, 13, 14, 17, and 18 with overall survival probability < 0.5)**

From Figure 4.4A, we can see that not all unfavourable/poor prognosis clusters classified based on survival probabilities from Kaplan-Meier curves in Figure 4.3B are associated with high confidence (more than half of the patients annotated with certain risk) to the 'poor' or 'intermediate' molecular/cytogenetic risk annotation (i.e. Clusters 1, 7, 10, 11 and cluster 4, respectively). Here, we have found that clusters 3, 5, 13, 14 and 17 all have poor prognosis although they are mostly noted as with 'intermediate' risk. For details of the genes and mutational statuses related to these clusters, please see Table 4.2.

Cluster 1 is a unique group for AML as both patients in this cluster are of poor prognosis. Notable genes in this cluster are KMT2A, ELL, and SLITRK5. MLL or its alias KMT2A- a homologue of *Drosophila trithorax* protein which has methyltransferase activity lies on chromosome 11, and is frequently involved in translocations with other genes including, but not limited to ELL thus, producing fusion proteins that lost methyltransferase activity and promotes transition of hematopoietic cell into becoming leukaemia stem cells [144, 145]. The most variably expressed gene in this cluster is SLITRK5. SLITRKs are expressed predominantly in neural tissues and have neurite-modulating activity [146]. Milde and co-workers (2007) have found that SLITRKs could be involved in normal as well as malignant

haematopoiesis and as novel marker of hematopoietic stem cells [147]. We also found that genes that are down-regulated involved in negative regulation of cell communication/signal transduction.

Similarly, cluster 3 was made up of just two patients and both have recurrent PHF6 mutation. Amino acid mutation (i.e. arginine and lysine) of PHF6 protein domain causes it to lose its DNA-binding capacity. Todd and co-workers have come to the conclusion that PHF6 is a tumour suppressor based on coherent findings with another four cohorts of studies where mutation of PHF6 results in loss of function and correlates with enhanced tumour progression [148]. There is no clear link between up-regulation of the only significantly expressed gene, CYP4F2 (with GO annotations related iron ion binding and oxidoreductase) in cluster 3 (refer to Table 4.2) and poor prognosis of AML. The down-regulated genes are significantly enriched in GO terms for positive regulation of the innate immune response (see Table 4.3 below). This is clear evidence that loss of function of PHF6 together with down-regulation of innate immune response contributes to the overall poor prognosis of this cluster.

Cluster 4, 5, and 13 were found with co-mutations between different classes of mutations (i.e. mutation class I-II-III, FLT3-NPM1-DNMT3A). Although cytogenetic/molecular risk for all three clusters was of 'intermediate' risk, Kaplan-Meier curves for cluster 4 and 5 show that both clusters were associated with low survival probability (~0.1 at 40 months) and cluster 13 has better survival probability. According to GeneCardV3 [149], FLT3 is a class III receptor tyrosine kinase that regulates haematopoiesis through pathways associated with apoptosis, proliferation, and hematopoietic cell differentiation. NPM1 encodes a phosphoprotein which travels between the nucleus and cytoplasm and its gene product is involved in the regulation of the ARF/p53 pathway [149]. DNMT3A gene encodes a DNA methyltransferase that functions in de novo methylation, rather than maintenance methylation [149]. Gene expression profiling for cluster 4 (refer to Table 4.2) strongly implicates that the HOX gene cluster together with MEIS1 (the HOX regulator) as the critical downstream target genes of cluster 4 genes co-mutations. This gene family encodes DNA-binding transcription factors that may regulate gene expression, function in fertility, embryo viability, and the regulation of hematopoietic stem cell expansion and lineage commitment [149]. Genes that are up-regulated in all three clusters are mostly HOX gene cluster and functionally enriched in myeloid cell differentiation regulation and embryonic development.

Cluster 7 with FLT3 and NSD1 co-mutations is also associated with a poor outcome. NSD1 is gene involved in epigenetic regulation, similar to DNMT3A, and could be classified similarly in class III. CCLA1 (Cyclin A1)- a cell cycle protein was overexpressed in cluster 7 and it plays a role in the growth and suppression of apoptosis in these leukemic cells [150]. TP53 mutation,

in cluster 11 is associated with the shortest survival and poor risk. This can be explained by its function as a tumour suppressor gene in keeping cells from dividing in an uncontrolled way. Genes responsible for cell communication are overexpressed in this cluster (see Table 4.3 below). Cell-cell and cell-surrounding stroma/extracellular matrix communication plays an important role in promoting tumour/haematopoietic progenitor cell survival, expansion and differentiation [151].

Cluster 14 and 17 are both associated with a single gene mutation, STAG2 and CEPBA, respectively. STAG2 protein is a subunit of the cohesion complex which is responsible for the separation of sister chromatids during cell division and any defects would result in aneuploidy in human cancer [149]. Cluster 17 is associated with intermediate molecular/cytogenetic risk and similar to the intermediate survival prognosis we have found from the survival curve. The CEPBA gene encodes a transcription factor that recognizes the CCAAT motif in the promoters of target genes to form homo- and heterodimers with CCAT and enhancer binding proteins [149]. CEPBA modulates the expression of cell cycle genes as well as homeostasis. It is known that mutation of CEPBA is associated with a favourable prognosis of AML. We have not found a strong association between STAG2 and CEPBA mutations and the genes that are aberrantly expressed in each of these clusters.

The final cluster associated with unfavourable prognosis of AML is cluster 18. There are three genes co-mutated with low confidence (i.e. TP53, RUNX1, and DNMT3A) in this cluster. Co-mutations of genes from three different classes of mutations associated with AML, Class II (TF- RUNX1), III (DNA methylation-DNMT3A), and tumour suppressor class (TP53) have resulted in the patient's poor outcome with 20 percent chance of surviving by the age of 40. Genes that are over-expressed in this cluster are associated with negative regulation of neurogenesis and cell differentiation which could block the myeloid cell differentiation.

Clus.	Mutation	Biological process enrichment (GO terms)
C1	KMT2A / ELL	Negative regulation of cell communication/signal transduction
C3	PHF6	Positive regulation of innate immune response
C4	NPM1 - FLT3 - DNMT3A	Embryonic skeletal development; regulation of transcription; regulation of primary metabolic processes; hematopoietic/lymphoid organ development; immune system process/response;
C5	NPM1 - FLT3 - DNMT3A*	Regulation of signal transduction; response to stimulus; Cell differentiation Embryonic development; negative regulation of immune system process/response; negative regulation of myeloid cell differentiation; blood vessel development
C11	TP53	Response to stimulus; cell communication
C13	NPM1 - DNMT3A	Positive regulation of cytokine production; positive regulation of angiogenesis Embryonic development; hematopoietic/lymphoid organ development; immune system development; blood vessel morphogenesis; negative regulation of myeloid cell differentiation
C18**	TP53 - RUNX1 - DNMT3A	Negative regulation of neurogenesis; negative regulation of cell differentiation

**Table 4.3** Poor prognosis clusters enriched with at least one statistically significant biological process GO term (P-value < 0.05).

GO terms in blue and red are associated with the down- and upregulated genes.

\* Lower confidence mutation.

\*\* Cluster 18 with less confident mutations (probabilities <0.5).



### **Favourable clusters (clusters 2, 6, 8, 9, 12, 15, and 16 with overall survival probability > 0.5)**

From Figure 4.4A, we can see that clusters 8, 9, and 12 are annotated with ‘good’ molecular/cytogenetic risk from the clinical data provided and the rest are associated with ‘intermediate’ risk. For details of the genes and mutational statuses related to these clusters, refer to Table 4.2. Cluster 2 and 16 are associated with NPM1 mutation, whereas cluster 6 is associated with FLT3 mutation. Although both genes have been implicated in the poor prognosis clusters previously, we do not have enough evidence to infer cluster 2 and 6 as mutations that play a role in the overall survival status (i.e. dead or alive) as there are only 2 patients per cluster. However, for cluster 16, NPM1 mutation alone affects the patient’s survival probability less than when NPM1 is in combination with FLT3 (Cluster 5) or DMNT3A-FLT3 (Cluster 4). Interestingly, although genes functional in the negative regulation of myeloid cell differentiation and the immune system and still over-expressed, the effects of these genes might be overcome by the down-regulation of genes responsible for positive regulation of cell proliferation and the negative regulation of apoptosis (see Table 4.4). This might support our hypothesis as to why co-mutations associated with NPM1 have worse outcomes than NPM1 alone, suggesting that NPM1 co-mutations might cooperatively participate in the development of AML.

The fusion mutations in cluster 8 (RUNX1/RUNX1T1), cluster 9 (CBFB/MYH11) and cluster 12 (PML/RARA) are clustered well with their respective expression patterns [152]. We note a single sample in the middle of cluster 12 that has PML/RARA mutations and a distinct gene expression pattern but is not annotated with the accurate cytogenetic abnormality; this appears to be an annotation error. Clusters 8, 9 and 12 are also associated with survival differences (refer to Figure 4.4B). These fusion mutations are all associated with a favourable prognosis of AML based on Kaplan-Meier analysis in Figure 4.3B as well as established molecular and cytogenetic risks [143] in Figure 4.4A. Cluster 8, 9 and 12 mutations mostly fall into class II mutations (see section 4.1.2), where genetic alterations of these genes impair hematopoietic differentiation and might be responsible for the cell’s survival advantage by interfering with terminal differentiation and apoptosis [127, 153, 154].

In the case of PML/RARA, transcript aberration can interfere with the signalling pathway of both PML and RARA. Moreover, with lower level of retinoic acids, PML/RARA can recruit co-repressors and HDACs to its target genes and promotes cells growth by blocking the cell apoptosis and inhibit haematopoietic cell differentiation [155, 156]. As for CBFB- a core-binding factor, fusion with MYH1- a gene coding for the myosin heavy chain, can interfere with

the core binding factor (CBF) and therefore block cell differentiation to promote cell proliferation [157-159].

The last gene fusion observed is RUNX1/RUNX1 in cluster 8. RUNX1/RUNX1T1 (also known as AML1/ETO) is the most common cytogenetic abnormality in AML. RUNX1 is a transcription factor from the class II of AML mutations and it forms a complex with the cofactor CBF $\beta$  that binds to the core element of many enhancers and promoters of genes which are involved in the hematopoietic stem cell differentiation into myeloid and lymphoid cell lineage [149, 160]. RUNX1/RUNX1T1 adversely affects the HOX genes family by down-regulating these genes (see Table 4.4) in comparison to the unfavourable clusters. It is possible that the favourable outcome of this fusion may be facilitated by the down regulation of genes that function in blocking the myeloid cells differentiation.

Clus.	Mutation	Biological process enrichment (GO terms)
C8	RUNX1 / RUNX1T1	Embryonic development; myeloid cell differentiation; angiogenesis; inflammatory response Response to oxidative stress; detoxification
C9	MYH11/ CBF $\beta$	Negative regulation of signal transduction; negative regulation of cell communication Innate immune response; negative regulation of programmed cell death
C12	PML / RARA	Embryonic development; negative regulation of myeloid cell differentiation, hematopoietic progenitor cell differentiation Cell proliferation; inflammatory response; myeloid mediated immune response
C16	NPM1	Positive regulation of cell proliferation; negative regulation of apoptosis; regulation of cell signalling Embryonic development, hematopoietic/lymphoid organ development; negative regulation of myeloid cell differentiation; negative regulation of immune system process

**Table 4.4** Intermediate (C13 and C16) and good prognosis clusters enriched with at least one statistically significant biological process GO term (P-value < 0.05).

GO terms in blue and red are associated with the down- and upregulated genes.

\* Lower confidence mutation.

\*\* Cluster 18 with less confident mutations (probabilities <0.5).

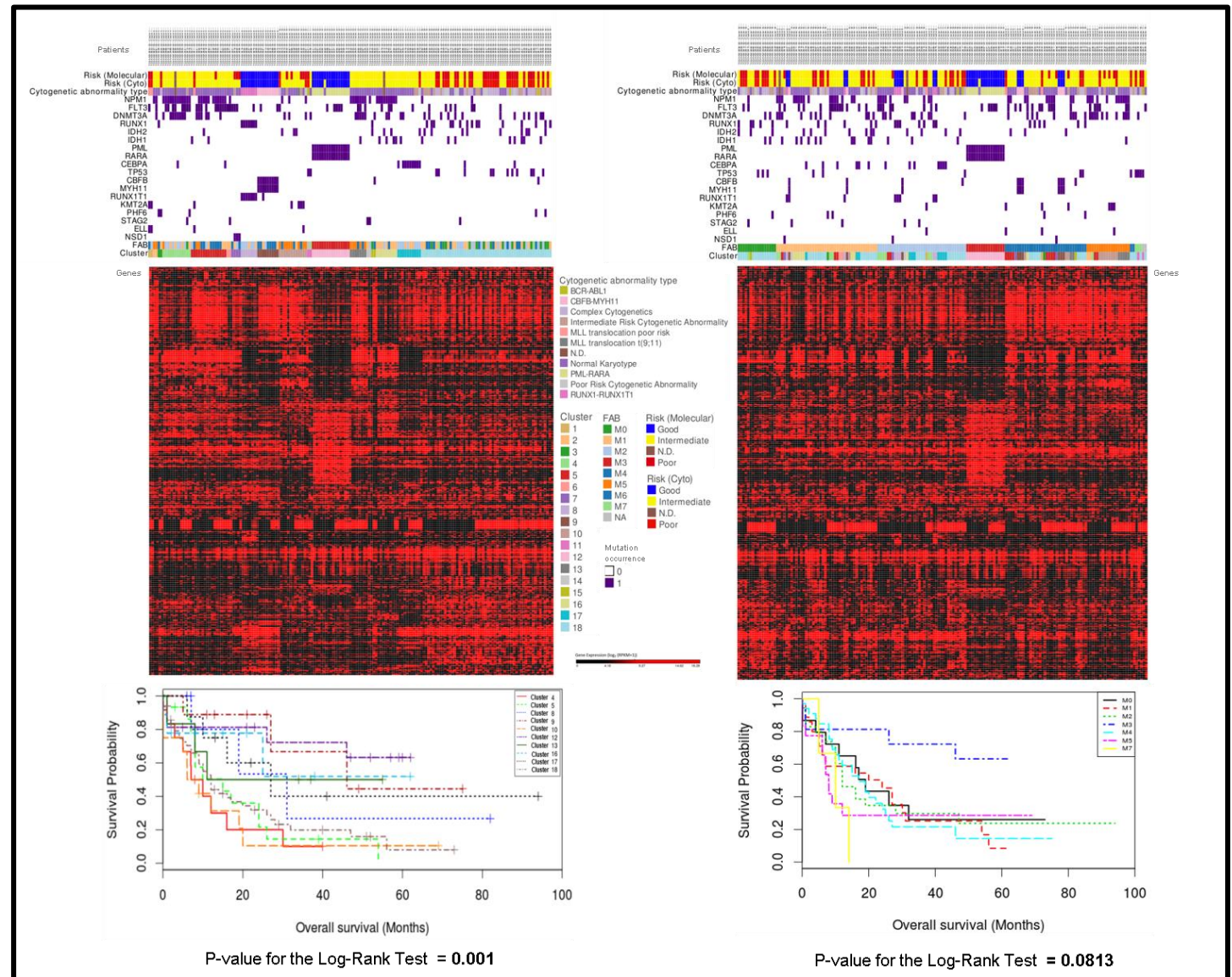
### **4.3.1. Comparison between our clusters and FAB classifications**

As mentioned before, the FAB classifications are purely based on cell histopathology or morphology, thus the grouping of patients might not be well correlated with survival, mutations and/or expression. We would like to compare our method with the FAB classification method in term of the prognosis of AML patients in each group. Using the same data and sorting patients based on FAB classifications has resulted in patients been grouped into 7 clusters corresponding to M0-7 FAB classes (see Figure 4.5-Right below). Our clusters are correlate with the cytogenetic/molecular risk or prognosis where good, intermediate and poor risks are clustered separately. In comparison with mixture of risks from using FAB classification (see top two rows in Figure 4.5), with the exception of cluster M3 from FAB classification (equivalent to our cluster 6), the rest of FAB clusters consist of mixture of mutations and expression patterns within cluster. We have explained earlier that mutational status is highly correlated with the survival of patients for AML, the Kaplan-Meier survival curves for FAB-based clusters are indistinct compared to our clusters survival curves. The Kaplan- Meier estimators for FAB-based clusters are not significantly different with a high p-value from the Log-Rank Test,  $p = 0.0813$ . There is not enough separation in the survival curves to be considered as good classifications of patients. On the contrary, our clusters are with the Kaplan- Meier estimators that are significantly different from each other and with lower p-value for the Log-Rank Test,  $p = 0.00$ . Hence, our method could separate the patients into groups with corresponding distinct survival probability and aberration markers better than the FAB classification.

**Figure 4.5** Comparison between our clustering method on AML patients and from using FAB classifications.

**Left:** Heatmap and Kaplan-Meier survival plot of clusters from our clustering algorithm.

**Right:** Heatmap and Kaplan-Meier survival plot of clusters from using FAB grouping



#### **4.4. Conclusion**

In this chapter, we have demonstrated that combination of different types of molecular aberrations including but not limited to gene mutations and expression are associated well with prognosis and have produced clusters which are biologically significant. This would need to be extended to a more complex study case where more data types and patients are becoming available before this could be implemented into the clinic. With just ~200 patients from AML, the variety of driver mutations and gene expressions we discovered would be less likely to be accepted as the model of AML patient classifiers as this requires a more thorough investigation once more patients, molecular and clinical data available. Until recently, FAB and WHO classifier are the main references for clinicians, but in the near future, combinatorial approach would be likely to be integrated into these existing classifications as we have demonstrated that combination of mutational status with gene expression contributes to a more prognostically distinct sub-classification of patients.

## Chapter 5. Discussion and future work

The work described in this thesis covers a single topic with wide range of applications, from automated clusters finding by integrating binary and continuous inputs to its application in reconstruction of transcriptional regulatory networks in yeast and cancer subtypes discovery. In this chapter, I will discuss the overall performance and limitation of this method and the potential future application of this work.

### 5.1. Method development

Our research began with the development of a method for clustering objects/data-points consisting of multiple data types simultaneously. The work carried out at this stage concentrate mostly on optimizing the proposed mixture model of maximum likelihood probabilities (i.e. Bernoulli's and Gaussian's distributions) in terms of the penalty criteria and runtime parameters. Initially, when we tested the algorithm with a range of penalty/information criteria (IC) on a lower-noise simulated dataset, it showed that most of them were able to find the correct solution. However, when we tested the algorithm on the yeast dataset, each IC produced different results, where stronger IC objective function produced fewer clusters than the less stringent penalty. Following this, we came to realized that our simulated data might not represent the real data well, hence, we introduced more noise into the simulated data to reflect the fact that variance exists in the real data. With more realistic simulated data, we were able to capture the different effects of ICs have on the results. An empirically derived AIC ( $\lambda = 2.5$ ) was found to be the optimum ICs to be used with the method. However, there is evidence that using the well-known AIC ( $\lambda = 2$ ) can sometimes produce much more biologically relevant clusters than AIC ( $\lambda = 2.5$ ) such as in the yeast case. As the multitude of features greater than the sample size or vice versa, the limitation of this method comes into sight where optimization has failed due to a large computational power required to solve the simulated annealing optimization procedure. To address this problem, we suggest that dimensional reduction strategies should be applied prior to our program, as our program is not suitable for primary explorative approach in discovery of novel molecular features, but rather locally grouping objects using features of interest to imply any interesting relationship that could be observed between them.

## 5.2. Transcriptional regulatory networks reconstruction

In this work, we choose to apply our method to a simple model organism which is yeast and model its cell cycle transcriptional regulatory networks. Cell cycle is an important process in sustaining an organism's life and knowing the complex mechanisms of this process would be useful in predicting similar behavior we would expect in much complex organism such as man. The availability of large scale TF ChIP-chip datasets and cell-cycle genes expression across 18 time-points for yeast means that our method can explore the transcriptional regulatory circuit and predictions made of modulation of genes regulatory interaction could potentially be added to the existing yeast cell cycle knowledge-base. For example, from the clusters found, we have been able to recapitulate the important key regulators of cell cycle and 3 more TFs (i.e. Ash1, Gzf3, and Met31) as potentially having a cell cycle regulation role. As for the 'unclear' clusters, we have given less consideration throughout the discussion in Chapter 3 although they are mostly cell cycle genes, no confident regulatory information could be used to infer the regulation of these clusters. It might be a case of regulation by other TFs than the ones that are currently available in the database or by other mechanisms (i.e. post-transcriptional and/or post-translational modifications). In addition, TF binding datasets at each time-point are not available at the time of writing and this incompleteness is limiting our ability to infer time-specific gene cluster regulation. Hence, we just inferred the regulatory hypothesis that a TF potentially regulates genes maximally expressed at specific stage of cell cycle.

Another advantage of our method is that co-regulation of genes can be inferred easily by looking at the co-binding of TFs to the gene promoter. This is important as we have noted in the chapter 1 that TF most of the time regulates target gene cooperatively, be it forming a complex with other co-activator/repressor directly onto the gene promoter or through a mediator. Apart from single and double TF-TF interactions, we have found triplet and quadruplet TF-TFs interactions, many of which are supported by genetic and/or physical interaction database (i.e. Gzf3-Pdr1-Swi5 and Ace2-Fkh1-Fkh2-Swi5).

### 5.3. Identification of cancer subtypes

The work carried out in Chapter 4 was done using the clustering method we have developed and it represents the importance of the integrative approach for the identification of cancer subtypes in AML. As explained before, cancer subtypes are usually strongly related to the cancer patient's prognosis as a result of different genetic aberrations. It is also important to note that proteomic and metabolomics aberrations might also contribute the prognostic properties observed. However, less data related to these omics are available and much attention has been given to molecular aberrations such as gene mutation (genomic data), methylation (epigenetic data), and expression (transcriptomic data) in this research area. Here, we have shown that our clustering method is capable of recapitulating the known fusion gene mutation into clusters which turned out to correlate well with gene expression and good prognosis or survival probability of patients. For example, PML/RARA, MYH11/CEPB, and RUNX1/RUNX1T1 clusters are correlated with gene expression aberrations responsible for distinct biological functions (i.e. embryonic development and myeloid cell differentiation, innate immune response and regulation of apoptosis). Similar to the yeast test case, some double, triple and quadruplet co-mutations (i.e. NPM1 - FLT3 - DNMT3A, NPM1 - DNMT3A, and TP53 - RUNX1 - DNMT3A) have been found to be responsible for the clustering of poor prognosis patients. Interestingly, genes that are responsible for poor patient survival (i.e. embryonic development, hematopoietic/lymphoid organ development, immune system development, blood vessel morphogenesis, negative regulation of myeloid cell differentiation) are all over-expressed as opposed to the fusion-related mutations. These validate our cluster-based AML subtypes as being functional subtypes thus suggesting that with the usage of appropriate dimensional reduction (here we used unsupervised dimensional reduction), our method can be applied to wide range of problems with different type of features from cancer related data.



#### 5.4. Future work

In this work, we demonstrated a few valuable applications of this method and the whole research community working on gene regulations and cancer subtypes/classifications could be benefited from using our method. In relation to the first problem, this novel method could be expanded from the study of yeast to data generated from much more complex eukaryotes (i.e. mouse) to build transcriptional regulatory network (i.e. Transcriptional regulation of the hematopoietic stem cell differentiation by TFs at the promoter region or enhancer). Equally, another aspect of future work would be to try it on further cancer data. Application of this method in prognostically relevant grouping of cancer patients based on cancer driver-gene aberrations would interest the cancer related research groups (for example, the research consortium in the University of Leeds which is currently works on Diffused large B-Cell Lymphoma (DLBCL) genes expression and mutations).

The prospective usage of this tool is broad and not restricted to just data related to molecular genetics. It can be applied to data generated from other fields such as, geology, economics, social sciences, and much more due to the nature of our method which is flexible in accepting any input that can be easily represented as binary and continuous. In addition, this method could be expanded to include other data types as well such as (ordinal or nominal data types) although these data types can easily be translated into binary form by renaming the features into orders or categories. In addition, this tool could contribute further to delineate relevant features from irrelevant features using over-represented features in each clusters found, be it biology related features of features from outside our biology-related field. Last but not least, if there is a way in the future to improve the speed of our method in optimizing the solution in order to accommodate larger dataset, an improvised version of this method could be used as the primary tool in discovering novel molecular features exploratively rather than locally such as what we are doing now.

## References

1. Regenmortel, M.H.V.V., *Reductionism and complexity in molecular biology*. EMBO Reports, 2004. **5**(11): p. 1016-1020.
2. Lodish, H.F., J.E. Darnell, and D. Baltimore, *Molecular cell biology*. 3. rev. ed. 1995, New York: Scientific American Books : Distributed by W.H. Freeman and Co. xlvii, 1344, 55 s.
3. Weinhold, B., *Epigenetics: The Science of Change*. Environmental Health Perspectives, 2006. **114**(3): p. A160-A167.
4. Ghazalpour, A., et al., *Comparative Analysis of Proteome and Transcriptome Variation in Mouse*. PLoS Genet, 2011. **7**(6): p. e1001393.
5. Schwanhauser, B., et al., *Global quantification of mammalian gene expression control*. Nature, 2011. **473**(7347): p. 337-342.
6. Cooper G, M., *The Cell: A Molecular Approach. 2nd edition*. The Complexity of Eukaryotic Genomes. 2000, Sunderland (MA): Sinauer Associates.
7. National Human Genome Research Institute, N. *The Human Genome Project Completion: Frequently Asked Questions*. 2003; Available from: <https://www.genome.gov/11006929/2003-release-international-consortium-completes-hgp/>.
8. Collins, F.S., et al., *Finishing the euchromatic sequence of the human genome*. Nature, 2004. **431**(7011): p. 931-945.
9. Drysdale, R., *FlyBase : a database for the Drosophila research community*. Methods Mol Biol, 2008. **420**: p. 45-59.
10. Stein, L., et al., *WormBase: network access to the genome and biology of Caenorhabditis elegans*. Nucleic Acids Research, 2001. **29**(1): p. 82-86.
11. O'Connor, C.M.A., J. U, *Essentials of Cell Biology*. 2010, Cambridge, MA: NPG Education.
12. Marsman, J. and J.A. Horsfield, *Long distance relationships: Enhancer–promoter communication and dynamic gene transcription*. Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms, 2012. **1819**(11–12): p. 1217-1227.
13. Mora, A., et al., *In the loop: promoter–enhancer interactions and bioinformatics*. Briefings in Bioinformatics, 2015.
14. Bhattacharjee, S., et al., *Combinatorial Control of Gene Expression*. BioMed Research International, 2013. **2013**: p. 407263.
15. Kato, M., et al., *Identifying combinatorial regulation of transcription factors and binding motifs*. Genome Biology, 2004. **5**(8): p. R56.

16. Teng, L., et al., *Discover context-specific combinatorial transcription factor interactions by integrating diverse ChIP-Seq data sets*. Nucleic Acids Research, 2014. **42**(4): p. e24-e24.
17. He, Q., et al., *High conservation of transcription factor binding and evidence for combinatorial regulation across six Drosophila species*. Nat Genet, 2011. **43**(5): p. 414-420.
18. Ravasi, T., et al., *An Atlas of Combinatorial Transcriptional Regulation in Mouse and Man*. Cell. **140**(5): p. 744-752.
19. Rivera, Chloe M. and B. Ren, *Mapping Human Epigenomes*. Cell. **155**(1): p. 39-55.
20. Jaenisch, R. and A. Bird, *Epigenetic regulation of gene expression: how the genome integrates intrinsic and environmental signals*. Nat Genet.
21. Bonasio, R., S. Tu, and D. Reinberg, *Molecular Signals of Epigenetic States*. Science (New York, N.Y.), 2010. **330**(6004): p. 612-616.
22. Egger, G., et al., *Epigenetics in human disease and prospects for epigenetic therapy*. Nature, 2004. **429**(6990): p. 457-463.
23. Jones, P.A. and S.B. Baylin, *The fundamental role of epigenetic events in cancer*. Nat Rev Genet, 2002. **3**(6): p. 415-428.
24. Meyer, C.A. and X.S. Liu, *Identifying and mitigating bias in next-generation sequencing methods for chromatin biology*. Nat Rev Genet, 2014. **15**(11): p. 709-721.
25. Hurd, P.J. and C.J. Nelson, *Advantages of next-generation sequencing versus the microarray in epigenetic research*. Briefings in Functional Genomics & Proteomics, 2009. **8**(3): p. 174-183.
26. Miller, M.B. and Y.-W. Tang, *Basic Concepts of Microarrays and Potential Applications in Clinical Microbiology*. Clinical Microbiology Reviews, 2009. **22**(4): p. 611-633.
27. Wang, Z., M. Gerstein, and M. Snyder, *RNA-Seq: a revolutionary tool for transcriptomics*. Nat Rev Genet, 2009. **10**(1): p. 57-63.
28. Trapnell, C., et al., *Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation*. Nat Biotech, 2010. **28**(5): p. 511-515.
29. Pillai, S. and S.P. Chellappan, *ChIP on Chip Assays: Genome-Wide Analysis of Transcription Factor Binding and Histone Modifications*, in *Chromatin Protocols: Second Edition*, S.P. Chellappan, Editor. 2009, Humana Press: Totowa, NJ. p. 341-366.
30. Park, P.J., *ChIP-seq: advantages and challenges of a maturing technology*. Nat Rev Genet, 2009. **10**(10): p. 669-680.

31. Li, H., J. Ruan, and R. Durbin, *Mapping short DNA sequencing reads and calling variants using mapping quality scores*. *Genome Research*, 2008. **18**(11): p. 1851-1858.
32. Zhang, Y., et al., *Model-based Analysis of ChIP-Seq (MACS)*. *Genome Biology*, 2008. **9**(9): p. R137.
33. Hanahan, D. and Robert A. Weinberg, *Hallmarks of Cancer: The Next Generation*. *Cell*. **144**(5): p. 646-674.
34. Pećina-Šlaus, N., *Tumor suppressor gene E-cadherin and its role in normal and malignant cells*. *Cancer Cell International*, 2003. **3**: p. 17-17.
35. Dang, Chi V., *MYC on the Path to Cancer*. *Cell*. **149**(1): p. 22-35.
36. The National Cancer Institute, N. *Types of Treatment*. 2014; Available from: <https://www.cancer.gov/about-cancer/treatment/types>.
37. Zack, T.I., et al., *Pan-cancer patterns of somatic copy number alteration*. *Nat Genet*, 2013. **45**(10): p. 1134-1140.
38. Daar, A.S., S.W. Scherer, and R.A. Hegele, *Implications of copy-number variation in the human genome: a time for questions*. *Nat Rev Genet*, 2006. **7**(6): p. 414-414.
39. Jin, B., Y. Li, and K.D. Robertson, *DNA Methylation: Superior or Subordinate in the Epigenetic Hierarchy?* *Genes & Cancer*, 2011. **2**(6): p. 607-617.
40. Kulis, M. and M. Esteller, *2 - DNA Methylation and Cancer*, in *Advances in Genetics*, H. Zdenko and U. Toshikazu, Editors. 2010, Academic Press. p. 27-56.
41. *The ENCODE (ENCyclopedia Of DNA Elements) Project*. *Science*, 2004. **306**(5696): p. 636.
42. Tomczak, K., P. Czerwińska, and M. Wiznerowicz, *The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge*. *Contemporary Oncology*, 2015. **19**(1A): p. A68-A77.
43. Mayr, E. and W.J. Bock, *Classifications and other ordering systems*. *Journal of Zoological Systematics and Evolutionary Research*, 2002. **40**(4): p. 169-194.
44. Sloutsky, R., et al., *Accounting for noise when clustering biological data*. *Briefings in Bioinformatics*, 2013. **14**(4): p. 423-436.
45. Wu, L.F., et al., *Large-scale prediction of *Saccharomyces cerevisiae* gene function using overlapping transcriptional clusters*. *Nat Genet*, 2002. **31**(3): p. 255-265.
46. Castro, R. and R. Nowak, *Likelihood Based Hierarchical Clustering and Network Topology Identification*, in *Energy Minimization Methods in Computer Vision and Pattern Recognition: 4th International Workshop, EMMCVPR 2003, Lisbon, Portugal, July 7-9, 2003. Proceedings*, A. Rangarajan, M. Figueiredo, and J.

- Zerubia, Editors. 2003, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 113-129.
47. D'Haeseleer, P., *How does gene expression clustering work?* Nat Biotech, 2005. **23**(12): p. 1499-1501.
  48. Vesanto, J. and E. Alhoniemi, *Clustering of the self-organizing map*. Trans. Neur. Netw., 2000. **11**(3): p. 586-600.
  49. Ringner, M., *What is principal component analysis?* Nat Biotech, 2008. **26**(3): p. 303-304.
  50. Bermingham, M.L., et al., *Application of high-dimensional feature selection: evaluation for genomic prediction in man*. Scientific Reports, 2015. **5**: p. 10312.
  51. Libbrecht, M.W. and W.S. Noble, *Machine learning applications in genetics and genomics*. Nat Rev Genet, 2015. **16**(6): p. 321-332.
  52. The Gene Ontology, C., et al., *Gene Ontology: tool for the unification of biology*. Nature genetics, 2000. **25**(1): p. 25-29.
  53. Huang, D.W., B.T. Sherman, and R.A. Lempicki, *Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources*. Nat. Protocols, 2008. **4**(1): p. 44-57.
  54. Subramanian, A., et al., *Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles*. Proceedings of the National Academy of Sciences, 2005. **102**(43): p. 15545-15550.
  55. Stephens, Z.D., et al., *Big Data: Astronomical or Genomical?* PLoS Biol, 2015. **13**(7): p. e1002195.
  56. Everitt B. S., S.A., *The Cambridge Dictionary of Statistics 4th edition*. 2010: Cambridge University Press.
  57. Edwards, A.W.F., *Likelihood*. 1992, Cambridge (expanded edition, 1992, Johns Hopkins University Press, Baltimore): Cambridge University Press.
  58. Moon, T.K., *The expectation-maximization algorithm*. IEEE Signal Processing Magazine, 1996. **13**(6): p. 47-60.
  59. Akaike, H., *New Look at Statistical-Model Identification*. IEEE Transactions on Automatic Control 1974. **19**(6): p. 716 - 723.
  60. Burnham, K.P. and D.R. Anderson, *Model selection and multimodel inference: a practical information theoretic approach*. 2nd ed. 2002, New York: Springer-Verlag.
  61. Schwarz, G., *Estimating Dimension of a Model*. The Annals of Statistics, 1978. **6**(2): p. 461-464.
  62. Dias, J., *Latent Class Analysis and Model Selection*, in *From Data and Information Analysis to Knowledge Engineering*, M. Spiliopoulou, et al., Editors. 2006, Springer Berlin Heidelberg. p. 95-102.

63. Bozdogan, H., *Model selection and Akaike's Information Criterion (AIC): The general theory and its analytical extensions*. Psychometrika, 1987. **52**(3): p. 345-370.
64. Hannan, E.J. and B.G. Quinn, *The Determination of the Order of an Autoregression*. Journal of the Royal Statistical Society. Series B (Methodological), 1979. **41**(2): p. 190-195.
65. Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi, *Optimization by Simulated Annealing*. Science, 1983. **220**(4598): p. 671.
66. Everitt, B.S., *A finite mixture model for the clustering of mixed-mode data*. Statistics & Probability Letters, 1988. **6**(5): p. 305-309.
67. Morlini, I., *A latent variables approach for clustering mixed binary and continuous variables within a Gaussian mixture model*. Advances in Data Analysis and Classification, 2012. **6**(1): p. 5-28.
68. McParland, D. and I.C. Gormley, *Model based clustering for mixed data: clustMD*. Advances in Data Analysis and Classification, 2016. **10**(2): p. 155-169.
69. Cai, J.-H., et al., *A mixture of generalized latent variable models for mixed mode and heterogeneous data*. Computational Statistics & Data Analysis, 2011. **55**(11): p. 2889-2907.
70. Shen, R., A.B. Olshen, and M. Ladanyi, *Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis*. Bioinformatics, 2009. **25**(22): p. 2906-2912.
71. Kim, S., et al., *Integrative phenotyping framework (iPF): integrative clustering of multiple omics data identifies novel lung disease subphenotypes*. BMC Genomics, 2015. **16**(1): p. 924.
72. Kirk, P., et al., *Bayesian correlated clustering to integrate multiple datasets*. Bioinformatics, 2012. **28**(24): p. 3290-3297.
73. Mason Samuel, A., et al., *MDI-GPU: accelerating integrative modelling for genomic-scale data using GP-GPU computing*, in *Statistical Applications in Genetics and Molecular Biology*. 2016. p. 83.
74. Bilmes, J.A., *A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models*. International Computer Science Institute, 1998. **4**(510): p. 126.
75. Corporation, O. *Designing a Swing GUI in NetBeans IDE*. Available from: <https://netbeans.org/kb/docs/java/quickstart-gui.html>.
76. Blais, A. and B.D. Dynlacht, *Constructing transcriptional regulatory networks*. Genes & Development, 2005. **19**(13): p. 1499-1511.

77. Eisen, M.B., et al., *Cluster analysis and display of genome-wide expression patterns*. Proceedings of the National Academy of Sciences, 1998. **95**(25): p. 14863-14868.
78. Segal, E., R. Yelensky, and D. Koller, *Genome-wide discovery of transcriptional modules from DNA sequence and gene expression*. Bioinformatics, 2003. **19**(suppl 1): p. i273-i282.
79. Tanay, A., et al., *Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data*. Proceedings of the National Academy of Sciences of the United States of America, 2004. **101**(9): p. 2981-2986.
80. Cho, R.J., et al., *A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle*. Molecular Cell, 1998. **2**(1): p. 65-73.
81. Gasch, A.P., et al., *Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes*. Molecular Biology of the Cell, 2000. **11**(12): p. 4241-4257.
82. Spitz, F. and E.E.M. Furlong, *Transcription factors: from enhancer binding to developmental control*. Nat Rev Genet, 2012. **13**(9): p. 613-626.
83. He, F., et al., *Dynamic cumulative activity of transcription factors as a mechanism of quantitative gene regulation*. Genome Biology, 2007. **8**(9): p. R181-R181.
84. Bar-Joseph, Z., et al., *Computational discovery of gene modules and regulatory networks*. Nat Biotech, 2003. **21**(11): p. 1337-1342.
85. Youn, A., D.J. Reiss, and W. Stuetzle, *Learning transcriptional networks from the integration of ChIP–chip and expression data in a non-parametric model*. Bioinformatics, 2010. **26**(15): p. 1879-1886.
86. Peña-Castillo, L. and T.R. Hughes, *Why Are There Still Over 1000 Uncharacterized Yeast Genes?* Genetics, 2007. **176**(1): p. 7-14.
87. Wittenberg, C. and S.I. Reed, *Cell cycle-dependent transcription in yeast: promoters, transcription factors, and transcriptomes*. 0000. **24**(17): p. 2746-2755.
88. Simon, I., et al., *Serial Regulation of Transcriptional Regulators in the Yeast Cell Cycle*. Cell, 2001. **106**(6): p. 697-708.
89. Crosby, M.E., *Cell Cycle: Principles of Control*. The Yale Journal of Biology and Medicine, 2007. **80**(3): p. 141-142.
90. Cokus, S., et al., *Modelling the network of cell cycle transcription factors in the yeast *Saccharomyces cerevisiae**. BMC Bioinformatics, 2006. **7**: p. 381-381.
91. Wu, W.-S. and W.-H. Li, *Systematic identification of yeast cell cycle transcription factors using multiple data sources*. BMC Bioinformatics, 2008. **9**(1): p. 522.
92. Haynes, B.C., et al., *Mapping Functional Transcription Factor Networks from Gene Expression Data*. Genome Research, 2013.

93. Harbison, C.T., et al., *Transcriptional regulatory code of a eukaryotic genome*. Nature, 2004. **431**(7004): p. 99-104.
94. Lee, T.I., et al., *Transcriptional Regulatory Networks in Saccharomyces cerevisiae*. Science, 2002. **298**(5594): p. 799-804.
95. Tsai, H.-K., H.H.-S. Lu, and W.-H. Li, *Statistical methods for identifying yeast cell cycle transcription factors*. Proceedings of the National Academy of Sciences of the United States of America, 2005. **102**(38): p. 13532-13537.
96. Cherry, J.M., et al., *Saccharomyces Genome Database: the genomics resource of budding yeast*. Nucleic Acids Research, 2012. **40**(Database issue): p. D700-D705.
97. Yu, G., et al., *GOSemSim: an R package for measuring semantic similarity among GO terms and gene products*. Bioinformatics, 2010. **26**(7): p. 976-978.
98. Chatr-aryamontri, A., et al., *The BioGRID interaction database: 2015 update*. Nucleic Acids Research, 2015. **43**(D1): p. D470-D478.
99. Kanehisa, M., et al., *KEGG as a reference resource for gene and protein annotation*. Nucleic Acids Research, 2016. **44**(D1): p. D457-D462.
100. Schmitt, W.A., R.M. Raab, and G. Stephanopoulos, *Elucidation of Gene Interaction Networks Through Time-Lagged Correlation Analysis of Transcriptional Data*. Genome Research, 2004. **14**(8): p. 1654-1663.
101. Wang, K., et al., *K-Profiles: A Nonlinear Clustering Method for Pattern Detection in High Dimensional Data*. BioMed Research International, 2015. **2015**: p. 918954.
102. Youn, A., D.J. Reiss, and W. Stuetzle, *Learning transcriptional networks from the integration of ChIP-chip and expression data in a non-parametric model*. Bioinformatics, 2010. **26**(15): p. 1879-86.
103. Haase, S.B. and C. Wittenberg, *Topology and Control of the Cell-Cycle-Regulated Transcriptional Circuitry*. Genetics, 2014. **196**(1): p. 65.
104. Kanehisa, M. and S. Goto, *KEGG: Kyoto Encyclopedia of Genes and Genomes*. Nucleic Acids Research, 2000. **28**(1): p. 27-30.
105. Breeden, L., *Start-Specific Transcription in Yeast*, in *Transcriptional Control of Cell Growth: The E2F Gene Family*, P.J. Farnham, Editor. 1996, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 95-127.
106. Koch, C., et al., *A role for the transcription factors Mbp1 and Swi4 in progression from G1 to S phase*. Science, 1993. **261**(5128): p. 1551-1557.
107. Ho, Y., et al., *Regulation of Transcription at the Saccharomyces cerevisiae Start Transition by Stb1, a Swi6-Binding Protein*. Molecular and Cellular Biology, 1999. **19**(8): p. 5267-5278.



108. Enserink, J.M. and R.D. Kolodner, *An overview of Cdk1-controlled targets and processes*. Cell Division, 2010. **5**: p. 11-11.
109. Bertoli, C., J.M. Skotheim, and R.A.M. de Bruin, *Control of cell cycle transcription during G1 and S phases*. Nature reviews. Molecular cell biology, 2013. **14**(8): p. 518-528.
110. Sia, R.A., E.S. Bardes, and D.J. Lew, *Control of Swe1p degradation by the morphogenesis checkpoint*. The EMBO Journal, 1998. **17**(22): p. 6678-6688.
111. Eriksson, P.R., et al., *Regulation of Histone Gene Expression in Budding Yeast*. Genetics, 2012. **191**(1): p. 7-20.
112. Su, N.-Y., et al., *A Dominant Suppressor Mutation of the met30 Cell Cycle Defect Suggests Regulation of the Saccharomyces cerevisiae Met4-Cbf1 Transcription Complex by Met32*. Journal of Biological Chemistry, 2008. **283**(17): p. 11615-11624.
113. Mizunuma, M., et al., *Involvement of S-adenosylmethionine in G(1) cell-cycle regulation in Saccharomyces cerevisiae*. Proceedings of the National Academy of Sciences of the United States of America, 2004. **101**(16): p. 6086-6091.
114. Jacobson, M.D., et al., *Testing Cyclin Specificity in the Exit from Mitosis*. Molecular and Cellular Biology, 2000. **20**(13): p. 4483-4493.
115. Iyer, V.R., et al., *Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF*. Nature, 2001. **409**.
116. Zhu, G., et al., *Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth*. Nature, 2000. **406**(6791): p. 90-4.
117. Bennett, J.M., et al., *Proposal for the recognition of minimally differentiated acute myeloid leukaemia (AML-M0)*. British Journal of Haematology, 1991. **78**(3): p. 325-329.
118. Bennett, J.M., et al., *Criteria for the diagnosis of acute leukemia of megakaryocyte lineage (m7): A report of the french-american-british cooperative group*. Annals of Internal Medicine, 1985. **103**(3): p. 460-462.
119. Bennett, J.M., et al., *Proposals for the Classification of the Acute Leukaemias French-American-British (FAB) Co-operative Group*. British Journal of Haematology, 1976. **33**(4): p. 451-458.
120. Vardiman, J.W., et al., *The 2008 revision of the World Health Organization (WHO) classification of myeloid neoplasms and acute leukemia: rationale and important changes*. Blood, 2009. **114**(5): p. 937-951.
121. Golub, T.R., et al., *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring*. Science, 1999. **286**(5439): p. 531-7.
122. Ross, M.E., et al., *Gene expression profiling of pediatric acute myelogenous leukemia*. Blood, 2004. **104**(12): p. 3679-87.

123. Valk, P.J., et al., *Prognostically useful gene-expression profiles in acute myeloid leukemia*. N Engl J Med, 2004. **350**(16): p. 1617-28.
124. Renneville, A., et al., *Cooperating gene mutations in acute myeloid leukemia: a review of the literature*. Leukemia, 2008. **22**(5): p. 915-31.
125. Becker, H., et al., *Prognostic gene mutations and distinct gene- and microRNA-expression signatures in acute myeloid leukemia with a sole trisomy 8*. Leukemia, 2014. **28**(8): p. 1754-8.
126. Walter, R.B., et al., *Significance of FAB subclassification of "acute myeloid leukemia, NOS" in the 2008 WHO classification: analysis of 5848 newly diagnosed patients*. Blood, 2013. **121**(13): p. 2424-2431.
127. Dombret, H., *Gene mutation and AML pathogenesis*. Blood, 2011. **118**(20): p. 5366.
128. Naoe, T. and H. Kiyoi, *Gene mutations of acute myeloid leukemia in the genome era*. International Journal of Hematology, 2013. **97**(2): p. 165-174.
129. Kansal, R., *Acute myeloid leukemia in the era of precision medicine: recent advances in diagnostic classification and risk stratification*. Cancer Biology & Medicine, 2016. **13**(1): p. 41-54.
130. Kadia, T.M., et al., *Toward individualized therapy in acute myeloid leukemia: A contemporary review*. JAMA Oncology, 2015. **1**(6): p. 820-828.
131. Gulley, M.L., T.C. Shea, and Y. Fedoriw, *Genetic Tests To Evaluate Prognosis and Predict Therapeutic Response in Acute Myeloid Leukemia*. The Journal of Molecular Diagnostics : JMD, 2010. **12**(1): p. 3-16.
132. *Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia*. N Engl J Med, 2013. **368**(22): p. 2059-74.
133. Cerami, E., et al., *The cBio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data*. Cancer Discov, 2012. **2**(5): p. 401-4.
134. Gao, J., et al., *Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal*. Sci Signal, 2013. **6**(269): p. p11.
135. Zhao, Q., et al., *Combining multidimensional genomic measurements for predicting cancer prognosis: observations from TCGA*. Briefings in Bioinformatics, 2015. **16**(2): p. 291-303.
136. Harvard., B.I.o.M. *FIREHOSE Broad GDAC*. Available from: <https://gdac.broadinstitute.org/>.
137. Clark, T.G., et al., *Survival Analysis Part I: Basic concepts and first analyses*. British Journal of Cancer, 2003. **89**(2): p. 232-238.
138. Bewick, V., L. Cheek, and J. Ball, *Statistics review 12: Survival analysis*. Critical Care, 2004. **8**(5): p. 389-394.

139. Reich, M., et al., *GenePattern 2.0*. Nat Genet, 2006. **38**(5): p. 500-501.
140. Huang, D.W., et al., *The DAVID Gene Functional Classification Tool: a novel biological module-centric algorithm to functionally analyze large gene lists*. Genome Biology, 2007. **8**(9): p. R183-R183.
141. Eden, E., et al., *GORilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists*. BMC Bioinformatics, 2009. **10**: p. 48-48.
142. Subramanian, A., et al., *Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles*. Proceedings of the National Academy of Sciences of the United States of America, 2005. **102**(43): p. 15545-15550.
143. Kihara, R., et al., *Comprehensive analysis of genetic alterations and their prognostic impacts in adult acute myeloid leukemia patients*. Leukemia, 2014. **28**(8): p. 1586-1595.
144. Krivtsov, A.V. and S.A. Armstrong, *MLL translocations, histone modifications and leukaemia stem-cell development*. Nat Rev Cancer, 2007. **7**(11): p. 823-833.
145. Marc R. Mansour , A.T.L., *Chromosomal Translocations and Genome Rearrangements in Cancer*, in *Chromosomal Translocations and Genome Rearrangements in Cancer*, M.M.L.B. Janet D. Rowley, Terence H. Rabbitts, Editor. 2016, Springer International Publishing: Switzerland. p. 189-222.
146. Aruga, J., N. Yokota, and K. Mikoshiba, *Human SLITRK family genes: genomic organization and expression profiling in normal brain and brain tumor tissue*. Gene, 2003. **315**: p. 87-94.
147. Milde, T., et al., *A novel family of slitrk genes is expressed on hematopoietic stem cells and leukemias*. Leukemia, 2007. **21**(4): p. 824-827.
148. Todd, M.A.M., D. Ivanochko, and D.J. Picketts, *PHF6 Degrees of Separation: The Multifaceted Roles of a Chromatin Adaptor Protein*. Genes, 2015. **6**(2): p. 325-352.
149. Safran, M., et al., *GeneCards Version 3: the human gene integrator*. Database: The Journal of Biological Databases and Curation, 2010. **2010**: p. baq020.
150. Pandey, S., et al., *Abstract 773: WT1 regulation of Cyclin A1 in leukemia*. Cancer Research, 2014. **73**(8 Supplement): p. 773.
151. Gillette, J.M. and J. Lippincott-Schwartz, *Hematopoietic progenitor cells regulate their niche microenvironment through a novel mechanism of cell-cell communication*. Communicative & Integrative Biology, 2009. **2**(4): p. 305-307.
152. *Genomic and Epigenomic Landscapes of Adult De Novo Acute Myeloid Leukemia*. New England Journal of Medicine, 2013. **369**(1): p. 98-98.
153. Takahashi, S., *Current findings for recurring mutations in acute myeloid leukemia*. Journal of Hematology & Oncology, 2011. **4**: p. 36-36.

154. Speck, N.A. and D.G. Gilliland, *Core-binding factors in haematopoiesis and leukaemia*. Nat Rev Cancer, 2002. **2**(7): p. 502-513.
155. Park, D.J., et al., *Comparative analysis of genes regulated by PML/RAR alpha and PLZF/RAR alpha in response to retinoic acid using oligonucleotide arrays*. Blood, 2003. **102**(10): p. 3727-36.
156. Puccetti, E. and M. Ruthardt, *Acute promyelocytic leukemia: PML//RAR[alpha] and the leukemic stem cell*. Leukemia, 2004. **18**(7): p. 1169-1175.
157. Castilla, L.H., et al., *The fusion gene Cbfb-MYH11 blocks myeloid differentiation and predisposes mice to acute myelomonocytic leukaemia*. Nat Genet, 1999. **23**(2): p. 144-146.
158. Haferlach, C., et al., *AML with CFBF-MYH11 rearrangement demonstrate RAS pathway alterations in 92% of all cases including a high frequency of NF1 deletions*. Leukemia, 2010. **24**(5): p. 1065-1069.
159. Schwind, S., et al., *inv(16)/t(16;16) acute myeloid leukemia with non-type A <em></em>CBFB-MYH11</em> fusions associate with distinct clinical and genetic features and lack <em></em>KIT</em> mutations*. Blood, 2013. **121**(2): p. 385.
160. Tonks, A., et al., *Transcriptional dysregulation mediated by RUNX1-RUNX1T1 in normal human progenitor cells and in acute myeloid leukaemia*. Leukemia, 2007. **21**(12): p. 2495-2505.

## Appendix A

### FlexiCoClustering User Manual

---



**Bioinformatics Group**

Leeds University and St. James' Hospital

Flexible model-based co-clustering – FlexiCoClustering manual

Joint clustering of binary and continuous data

Authors: Fatin Zainul Abidin\*, David Westhead\*\*

Contacts:

\*bs12fnza@leeds.ac.uk

\*\*D.R.Westhead@leeds.ac.uk

Section 1: Introduction to algorithm

---

The method is designed to cluster multiple data of different types where each entity for clustering is described by a sets of binary and continuous variables. It is generically applicable to a range of different problems in biology. It uses a simple model based framework based on a joint probability distribution over binary and continuous variables that is a mixture over a variable number of clusters. It uses penalized maximum likelihood (ML) estimation of mixture model parameters using information criteria and meta-heuristic searching for optimum clusters by Monte-Carlo simulated annealing (SA). The program takes as input a mixture of binary data (e.g. presence/absence of mutations, motifs, regulatory input, epigenetic marks etc.) and continuous data (e.g. gene expression, protein abundance, metabolite levels) for a list of samples (e.g. genes, patients). This program works best with smaller and concise datasets, thus pre-filtered data to only important features is preferable. To date, this program works well with ~1000 rows in the input file (number of entities to cluster) and longer run time might be needed for larger datasets to converge to a good solution. An example of pre-filtering of a dataset would be reducing the number of genes to only highly variable genes. Upon taking the input files required, the program will run until either the termination or convergence criterion are met. The clustering solution is then refined using expectation maximization, taking the simulated annealing solution as the starting point.

### Platform Dependencies

Task Type: Clustering of data points

CPU Type: any

Operating System: any

Language: Java - JDK 1.8

Section 2: How to run the Program

---

? How to run the FlexiCoClustering using command-line interface (FlexiCoClustering-CLI)

1. Download all the files from <https://github.com/BioToolsLeeds/FlexiCoClusteringPackage/>
2. To run the demo , use the following command:

```
java -jar <path to the FlexiCoClustering.jar/FlexiCoClustering.jar> <Input.txt>
<Output.txt>
```

Press enter

3. To re-submit the same job with restart after program termination:

Change Nrun: '0' to Nrun: '1'

Increase the MaxTemps to a higher value than the previous run if the MaxTemps iteration was

Completed or else just use default MaxTemps parameter. Then, use the following command:

```
java -jar <path to the FlexiCoClustering.jar/FlexiCoClustering.jar> <Input.txt>
<Output.txt>
```

Press enter

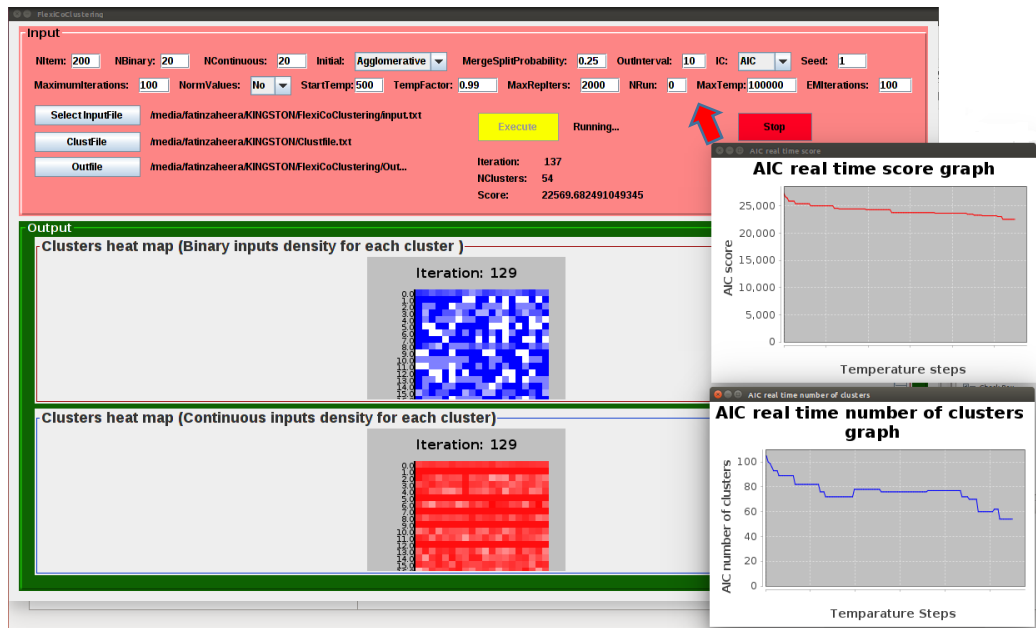
4. The program can be terminated at any time by pressing Ctrl+C.

A detailed description of the runtime parameters and input file format is given in section 3 below.

? How to run the FlexiCoClustering using GUI based (FlexiCoClustering-GUI)

1. Download all the files from <https://github.com/BioToolsLeeds/FlexiCoClusteringPackage/>
2. To run the demo, use the following command:  
*java -jar <path to the FlexiCoClustering.jar/FlexiCoClustering.jar> or double click the .jar file*

3. A graphical user interphase (GUI) window will be opened and looks like this snapshot below:



**Figure 1:** A snapshot of the FlexiCoClustering GUI upon submitting the `java -jar` command on the terminal/command prompt. Red arrow shows where user should change the NRun to '1' after the initial run (NRun=0) have finished if it is required at all.

4. On the GUI options (using demo example):

```
"ClustFile"      : Name the ClustFile as i.e. Clustfile.txt
"Outfile"        : Name an Outfile as i.e. Output.txt
"Select  inputFile" : Select <path to the
FlexiCoClustering.jar>\example\input.txt
```

For the real run using user own data, please change parameters accordingly. Parameters are described in Table 2 on the next page.

Press the "Execute" button. On default, this program will run for 100000 temperature steps (MaxTemps) and produce two real-time updated heat map image files (.png) in every 10 iterations interval for binary and continuous variables. The Output.txt and Clustfile.txt will be updated as the program progress and once the program terminated respectively.

An EM refining file (EMRefinement.txt) containing all the marginal densities for the clusters will be produced automatically in the same working directory.

If more than initially specified number of maximum temperature steps (MaxTemps) is required after step 2 had finished, re-run the program by first by replacing '0' with '1' in the Nrun. Increase the MaxTemps to a higher value than the previous run if the MaxTemps iteration was completed or else just use default MaxTemps parameter.

Then, press the "Execute" button again. User can also terminate the run at any time by pressing 'Stop' button.

Section 3: Package input and runtime parameters

Input file (e.g. input.txt)

A space separated formatted text file containing the binary and continuous input dataset and runtime parameters (command line interface only).

A	<pre> NItems: 100 NBinary: 20 NContinuous: 20 Agglomerative IC: AIC MergeSplitProbability: 0.25 MaximumIterations: 100 StartTemp: 500 TempFactor: 0.999 MaxTemps: 5000 MaxRepters: 2000 Seed: 1 EMIterations: 100 OutInterval: 10  ClustFile: clustfile.txt NormExp: 0 Nrun: 0 Gene 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 3.41 2.81 8.91 2.33 9.09 9.82 1.23 2.24 3.43 3.43 9.76 ... Gene 2 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 3.41 2.81 8.91 2.33 9.09 9.82 1.23 2.24 3.43 3.43 9.76 ... Gene 3 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 3.41 2.81 8.91 2.33 9.09 9.82 1.23 2.24 3.43 3.43 9.76 ... Gene 4 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 3.41 2.81 8.91 2.33 9.09 9.82 1.23 2.24 3.43 3.43 9.76 ... Gene 5 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 3.41 2.81 8.91 2.33 9.09 9.82 1.23 2.24 3.43 3.43 9.76 ... </pre>
B	<pre> BinaryInputs: TF1 TF2 TF3 TF4 TF5 TF6 TF7 TF8 TF9 TF10 TF11 TF12 TF13 TF14 TF15 TF16 TF17 TF18 TF19 TF20 ContinuousInputs: Exp1 Exp2 Exp3 Exp4 Exp5 Exp6 Exp7 Exp8 Exp9 Exp10 Exp11 Exp12 Exp13 Exp14 Exp15 Exp16 ... Gene 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 3.41 2.81 8.91 2.33 9.09 9.82 1.23 2.24 3.43 3.43 9.76 ... Gene 2 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 3.41 2.81 8.91 2.33 9.09 9.82 1.23 2.24 3.43 3.43 9.76 ... Gene 3 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 3.41 2.81 8.91 2.33 9.09 9.82 1.23 2.24 3.43 3.43 9.76 ... Gene 4 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 3.41 2.81 8.91 2.33 9.09 9.82 1.23 2.24 3.43 3.43 9.76 ... Gene 5 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 3.41 2.81 8.91 2.33 9.09 9.82 1.23 2.24 3.43 3.43 9.76 ... </pre>

**Table 1:** Example of an input file for A. command line interphase (CLI) and B. graphical user interphase (GUI) based package. From 3rd or 21st row onwards of GUI or CLN based package respectively, first column shows the data points (i.e. gene names, sample names) and the second column onwards are the binary inputs ("1" and "0") followed by continuous values.

Output file (e.g. Output.txt)

A text file containing all the runtime updates such as score, current best modules/clusters, etc.

EM refinement file (e.g. EMRefinement.txt)

A text file containing all the marginal densities of each cluster found from the simulated annealing procedure.



<b>Name</b>	<b>Functional Description</b>
Nitems	Number of data points to cluster. Must be equal to the number of data lines (rows) in the input file.
Nbinary	Number of binary variables per entity to be clustered. Must be equal to the number of 1/0s at the start of each data line in the input file.
NContinuous	Number of continuous variables in the input files. Must be equal to the number of floating point numbers at the end of each data line in the input file.
Agglomerative/Divisive	Starting point option for clustering, if agglomerative start with all data points in separate clusters, if divisive start with all in a single cluster.
IC ( <u>AIC</u> )	Objective function/Information criterion (see table below for options available)
MergeSplitProbability ( <u>0.25</u> )	Monte Carlo move: either an ordinary step (moving a data point between clusters) or a cluster merge/split according to this probability.
MaximumIterations ( <u>100</u> )	Number of Monte-Carlo moves at each temperature
StartTemp ( <u>500</u> )	Starting temperature of the simulated annealing. Higher temperature-more random solution will be accepted at the beginning of simulated annealing.
TempFactor ( <u>0.99</u> )	Temperature reduction factor at each iteration of the temperature loop.
MaxTemps ( <u>100000</u> )	Maximum number of temperature to be simulated (termination criterion). Higher value will make the SA runs longer.
MaxRepters ( <u>2000</u> )	Maximum number of simulated annealing best score repetitions. If the score does not change at up to this number of repetition, the SA will be terminated although the maximum temperature is not reached.
Seed ( <u>1</u> )	The seed of the random number generator.
EMIterations ( <u>100</u> )	Maximum number EM iterations
OutInterval ( <u>10</u> )	Intervals at which the solution is printed to the output file and at which the heat maps are updated on GUI
ClustFile	The name of the final clusters output file
NormExp ( <u>0</u> )	Normalizes continuous inputs to zero mean and a standard deviation for each data points- z-scores.
Nrun ( <u>0</u> )	Number of re-run of the program after initial run

**Table 2:** Runtime parameters of both GUI and CLN based package. The underlined and bold values are the default values of the runtime parameters.

Section 4: Mathematical representation of score calculation

Solution (clusters) score calculation:

$$p(r_{i1}, \dots, r_{in_r}, e_{i1}, \dots, e_{in_g}) = \sum_{m=1}^N \alpha_m \prod_{j=1}^{n_r} B(r_{ij}; p_{mj}) \prod_{l=1}^{n_g} N(e_{il}; \mu_{ml}, \sigma_{ml}) \dots\dots\dots (1)$$

$N$ : data points  $i$ , representing genes, tumour samples etc.

$r_{ij} \in \{0,1\}, j = 1, \dots, n_r$  binary variables

$e_{il}, l = 1, \dots, n_g$  continuous variables

$\alpha_m$  are mixing coefficients  $\sum \alpha_m = 1$ .

$B$  denotes the Bernoulli distribution with parameter  $p_{mj}$ , and  $N$  is a normal distribution with parameters  $\mu_{ml}$  and  $\sigma_{ml}$ .

We assume a probability distribution (1) which is a mixture of  $N_m$  components (clusters).

In the case of genetic regulation the mixture components represent the well-known concept of a cluster of co-regulated genes, with, for example, Bernoulli parameters  $p_{mj}$  representing the probability of binding for particular transcription factors in promoter/enhancer elements, and the  $\mu_{ml}$  representing a shared average pattern of gene expression, which could be a time or developmental series but is not required to be. In the case of tumour samples, clusters could be related samples where Bernoulli parameters associate mutation probabilities at particular loci with shared patterns of oncogenic gene expression.

Since the number of clusters is unknown and difficult to estimate, an initial heuristic search was adopted for an approximately optimal model, followed by refinement of the solution by expectation maximization. The heuristic search employed a Monte-Carlo simulated annealing algorithm (see Algorithm 1) to optimize objective functions of the form

$$O(L, k) = -2L + \lambda k(N) \dots\dots\dots (2)$$

where  $L$  is the (maximized) log-likelihood from the distribution above,  $\lambda$  is a function of the number of data points  $N$  and  $k$  is the number of parameters in the model.

Several different functions  $\lambda(N)$  can be used with our algorithm as shown in table 3 below:

	Criterion	$\lambda$	N	Equation	Reference
a.	AIC2	2	1	$-2L + 2k$	Akaike, 1973
b.	AIC2.5	2.5	1	$-2L + 2.5k$	-
c.	AIC3	3	1	$-2L + 3k$	Bozdogan, 1993
d.	HQC	2	$\ln \ln(N_g)$	$-2L + 2k(\ln(\ln(N_g)))$	Hannan and Quinn, 1979
e.	AIC4	4	1	$-2L + 4k$	-
f.	AIC5	5	1	$-2L + 5k$	-
g.	BIC	1	$\ln N_g$	$-2L + k(\ln(N_g))$	Swartz, 1978
h.	CAIC	1	$\ln N_g + 1$	$-2L + k(\ln(N_g) + 1)$	Bozdogan, 1987

**Table 3:** Different objective functions tested and can be chosen by user sorted

ascendingly (from a. to h.) based on its stringency in penalizing free parameters and number of data points in the model.

<b>Monte-Carlo simulated annealing (SA) for clusters optimization.</b>	
1:	Normalize expression (genes)*
2:	Clusters = Initialize clusters (agglomerative/divisive)
3:	Best clusters = [list]
4:	Score = 0.0
5:	Old score = 0.0
6:	Best score = 0.0
7:	Difference = 0.0
8:	Count temperature = 0
9:	Temperature = Start temperature
10:	While Count temperature < Maximum temperature:
11:	Temperature * Temperature decreasing factor
12:	Score = Calculate modules score (clusters)
13:	beta = 1.0/temp
14:	While Iteration < Maximum iterations: **
15:	If random > Merge and split probability:
16:	Change (clusters)
17:	Else:
18:	If random > 0.5:
19:	Merge (clusters)
20:	Else: Split (clusters)
21:	Old Score = Calculate clusters score (clusters)
22:	Difference = Difference + (Score-Old score)
23:	If Difference < 0.0 or random < exponent(-beta*Difference):
24:	'accept'
25:	if Score < Best score:
26:	Best score = Score
27:	Best clusters = clusters
28:	Else:
29:	'reject'
30:	Score = Old score
31:	Count temperature + 1
32:	If Best score = Old Score:
33:	Count score = Count score + 1
34:	If Count score == 2000:
35:	break
36:	return Best score, Best clusters

**Algorithm 1:** Pseudo-code for the Monte-Carlo Simulated annealing algorithm.

\*Optional

\*\*This step runs in parallel (minimum of 5 parallel threads).

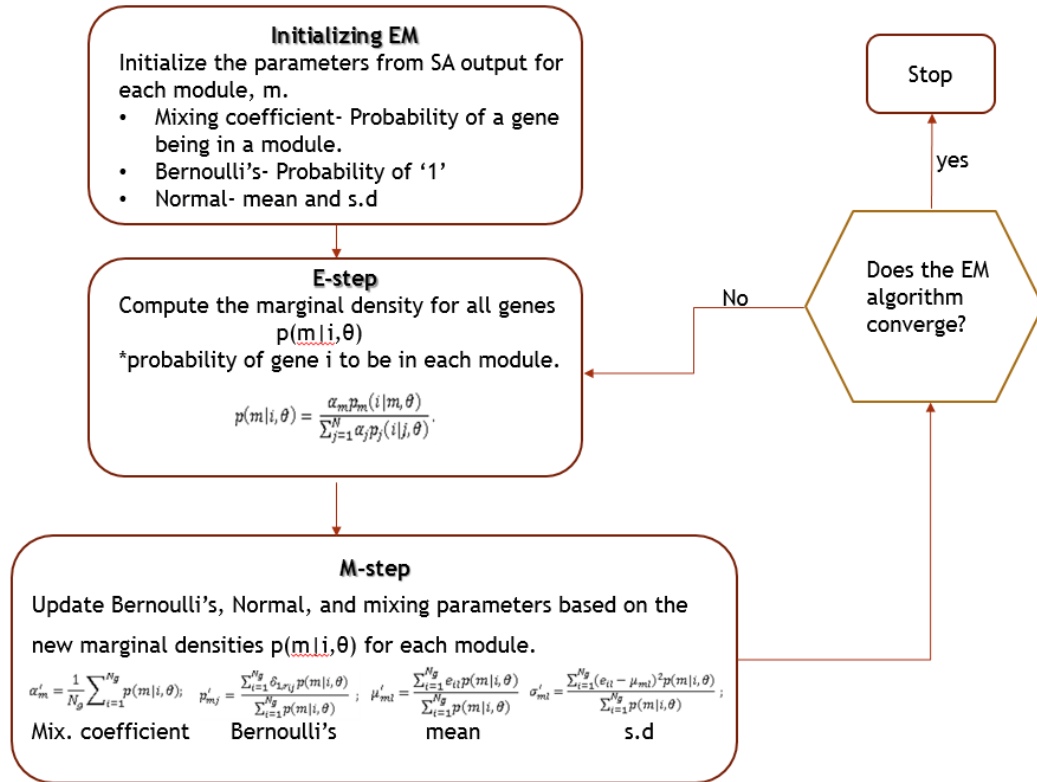
### Expectation-maximization

The parameters of model produced as the best solution from the heuristic search can be refined by EM, with the useful side effect of estimating the degree of mixing between modules through the probability density that data point  $i$  is generated from mixture component  $m$  (3).

$$p(m|i, \theta) = \frac{\alpha_m p_m(i|m, \theta)}{\sum_{j=1}^N \alpha_j p_j(i|j, \theta)} \dots \dots \dots (3)$$

This is derived from Bayes' rule:  $p_m$  is the probability density for mixture component  $m$  defined in 2.2,  $\theta$  denotes the (current) vector of parameters for all modules and the mixing coefficients  $\alpha_m$  can be interpreted as prior probabilities for membership of each module.

The steps of the EM algorithm are showed in the figure as followed:



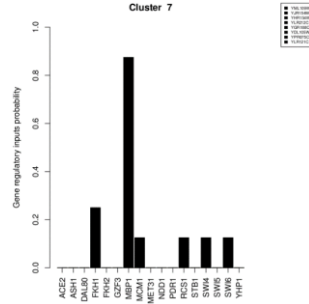
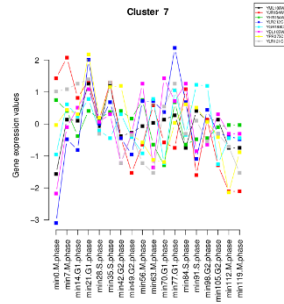
**Figure 1:** Refinement of model parameters using EM which starts with the prior probability densities from the SA output and refinement of its parameters until convergence. Convergence here means, until the parameters values do not changed for 2 consecutive EM iterations or until the maximum number of iterations has been reached.

## Appendix B

## GO terms enrichment and regulators for AIC 'clear' clusters

Clus	Expression	Regulation	Regulator	Gene	GO term enrichments (Biological process) p-value < 0.05 ; p-value > 0.05
1			PDR1 GZF3 SWI5	YDR085C YKR091W YGL032C YCL027W	Response to pheromone, sexual reproduction

7

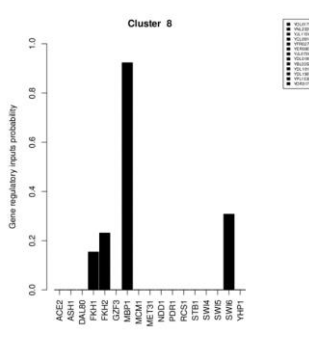
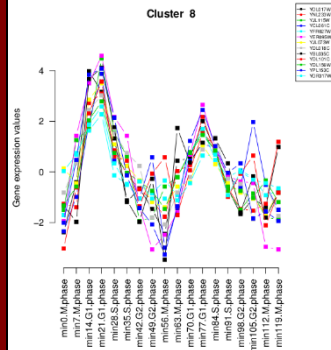


MBP1

YOL017W YNL233W YJL115W  
 YCL061C YFR027W  
 YER095W  
 YJL073W YDL018C YBL035C  
 YDL101C YDL156W YPL153C  
 YOR317W

DNA replication , cellular response to stress

8



MBP1

YOR066W YHR005C

Regulation of biological process

<p>10</p>	<p>Cluster 10</p>	<p>Cluster 10</p>	<p>MBP1</p>	<p>YLR383W YDR528W</p>	<p>Cellular process</p>
<p>16</p>	<p>Cluster 16</p>	<p>Cluster 16</p>	<p>SWI4 SWI6</p>	<p>YJR054W YDR501W</p>	<p>Cellular process</p>

<p>18</p>	<p>Cluster 18</p>	<p>Cluster 18</p>	<p>SWI4 SWI6 MBP1</p>	<p>YMR179W YBR071W</p>	<p>Regulation of transcription</p>
<p>19</p>	<p>Cluster 19</p>	<p>Cluster 19</p>	<p>MBP1</p>	<p>YNL102W YIL026C YJR030C YKR077W YHR153C YLL066C YBR073W</p>	<p>Chromosome segregation</p>



21			<p>SWI6 MBP1</p>	<p>YDL003W YKL113C</p>	<p>DNA repair, DNA replication, reproductive process</p>
25			<p>MBP1</p>	<p>YDR297W YKL101W YKL067W YFL008W YNL312W YPL241C YDR279W YKL165C YHR110W YMR076C</p>	<p>DNA repair, mitotic sister chromatid cohesion</p>

<p>27</p>	<p>Cluster 27</p>	<p>Cluster 27</p>	<p>ACE2</p> <p>FKH1</p> <p>FKH2</p> <p>SWI5</p>	<p>YJL078C YGL028C</p>	<p>-</p>
<p>29</p>	<p>Cluster 29</p>	<p>Cluster 29</p>	<p>SWI4</p> <p>MBP1</p>	<p>YPL267W YLR103C YPL124W</p>	<p>Cell cycle process, organelle organization</p>

30	<p style="text-align: center;"><b>Cluster 30</b></p>	<p style="text-align: center;"><b>Cluster 30</b></p>	<p>SWI4</p> <p>SWI6</p> <p>MBP1</p>	<p>YGR189C YKL103C YNL262W</p> <p>YMR305C YGR221C YPL256C</p> <p>YKR013W YGR238C</p>	<p>Conjugation with cellular fusion, sexual reproduction</p>
34	<p style="text-align: center;"><b>Cluster 34</b></p>	<p style="text-align: center;"><b>Cluster 34</b></p>	<p>MBP1</p>	<p>YJR043C YNL263C</p>	<p>-</p>

41	<p style="text-align: center;"><b>Cluster 41</b></p>	<p style="text-align: center;"><b>Cluster 41</b></p>	<p>MBP1</p>	<p>YBR007C YLR342W</p>	<p>Cell wall organization or biogenesis, carbohydrate biosynthetic process</p>
42	<p style="text-align: center;"><b>Cluster 42</b></p>	<p style="text-align: center;"><b>Cluster 42</b></p>	<p>ACE2 FKH1 FKH2</p>	<p>YER124C YLR286C YHR143W</p>	<p>Cell separation after cytokinesis, cell wall organization</p>

<p>48</p>	<p>Cluster 48</p>	<p>Cluster 48</p>	<p>ACE2</p> <p>FKH1</p> <p>SWI5</p>	<p>YGR041W YBR158W</p>	<p>Cell division</p>
<p>49</p>	<p>Cluster 49</p>	<p>Cluster 49</p>	<p>SWI4</p> <p>SWI6</p> <p>MBP1</p>	<p>YIL140W YER001W YML027W YER111C YPR120C YHR149C YKL045W YGR152C YMR199W</p>	<p>G1/S transition of mitotic cell cycle, cellular budding, cell division</p>

<p>61</p>	<p>Cluster 61</p>	<p>Cluster 61</p>	<p>MBP1</p>	<p>YPR135W YAR007C YOR033C          YOL007C YDR097C YJL074C          YPR174C YPR175W</p>	<p>DNA repair, cellular response to stimulus, chromosome organization, DNA replication</p>
<p>62</p>	<p>Cluster 62</p>	<p>Cluster 62</p>	<p>SWI4          SWI6</p>	<p>YGR014W YDL055C</p>	<p>Cell wall organization or biogenesis</p>

<p>63</p>			<p>SWI4 SWI6</p>	<p>YNL300W YPL163C</p>	<p>-</p>
<p>67</p>			<p>SWI4 SWI6 MBP1</p>	<p>YOR074C YER070W YJL187C YDR507C YCR065W YGL038C YBR070C YNL231C</p>	<p>Deoxy-ribonucleotide biosynthetic process, cell cycle check point, glycosylation</p>

<p>2</p>			<p>SWI6 MBP1</p>	<p>YNR009W YOR247W YMR307W YPL127C YNL126W YDR113C</p>	<p>Negative regulation of transcription, epigenetic</p>
<p>22</p>			<p>FKH2</p>	<p>YMR198W YHL028W</p>	<p>Establishment of localization in cell</p>



33	<p style="text-align: center;"><b>Cluster 33</b></p> <p style="text-align: center;"><b>Cluster 33</b></p>	FKH2	<p>YKL096W-A                      YPL116W</p> <p>YEL017W</p> <p>YPL075W   YKL096W   YML064C</p> <p>YOL030W   YIL131C   YNL176C</p>	<p>Cell wall organization, regulation of transcription</p>
40	<p style="text-align: center;"><b>Cluster 40</b></p> <p style="text-align: center;"><b>Cluster 40</b></p>	SWI4  SWI6	<p>YDR224C   YDR225W</p>	<p>Negative regulation of transcription,                      Chromatin assembly or disassembly</p>

<p>45</p>	<p>Cluster 45</p>	<p>Cluster 45</p>	<p>SWI6 MBP1</p>	<p>YER003C YLR437C YJL092W</p>	<p>Nucleobase, nucleoside, nucleotide and nucleic acid metabolic process</p>
<p>55</p>	<p>Cluster 55</p>	<p>Cluster 55</p>	<p>MBP1</p>	<p>YBR010W YNL031C YBL002W YBL003C YNL030W YBR009C</p>	<p>Chromatin assembly or disassembly</p>

<p>65</p>	<p>Cluster 65</p>	<p>Cluster 65</p>	<p>MET1</p>	<p>YOR152C YLR180W YLL061W YER091C YKL001C YGL184C</p>	<p>Sulphur amino acid biosynthetic process, methionine metabolic process</p>
<p>72</p>	<p>Cluster 72</p>	<p>Cluster 72</p>	<p>FKH1</p>	<p>YMR144W YLR210W</p>	<p>Spindle body separation, chromosome partitioning</p>

<p>56</p>	<p>Cluster 56</p>	<p>Cluster 56</p>	<p>FKH1</p> <p>FKH2</p> <p>SWI6</p>	<p>YFL037W                      YMR215W</p> <p>YER032W</p> <p>YIL123W    YLR056W    YJL158C</p> <p>YCL063W</p>	<p>Cell wall organization</p>
<p>71</p>	<p>Cluster 71</p>	<p>Cluster 71</p>	<p>ASH1</p> <p>FKH1</p> <p>FKH2</p> <p>SWI4</p>	<p>YPR013C    YOL114C</p>	<p>-</p>

<p>5</p>			<p>MCM1</p>	<p>YHR152W YMR253C</p>	<p>-</p>
<p>6</p>			<p>MCM1</p>	<p>YEL032W YAL040C YMR031C YLR274W YJL194W</p>	<p>Interphase, pre-replicative complex assembly, DNA replication</p>

<p>13</p>			<p>MCM1</p>	<p>YOR066W YHR005C</p>	<p>Regulation of biological process</p>
<p>14</p>			<p>FKH2 MCM1</p>	<p>YBR094W YDR191W YLR254C YFL026W YGR092W YDR190C YNL145W YGR143W YGL201C YAR018C YGR138C YBR139W YHR151C</p>	<p>Pheromone-dependent signal transduction involved in conjugation with cellular fusion</p>

<p>24</p>	<p>Cluster 24</p>	<p>Cluster 24</p>	<p>FKH2 NDD1</p>	<p>YDR033W YGL008C</p>	<p>Ion transport, establishment of localization</p>
<p>26</p>	<p>Cluster 26</p>	<p>Cluster 26</p>	<p>FKH2</p>	<p>YOR058C YBR138C</p>	<p>-</p>

<p>28</p>			<p>MBP1</p>	<p>YLR273C YOL011W YLR049C          YDL089W YFL011W YEL040W          YPR018W YNL225C YGR153W          YIL066C YOR230W YGR109C</p>	<p>DNA metabolic process, DNA replication</p>
<p>31</p>			<p>FKH2          NDD1</p>	<p>YPR149W YMR032W          YLR084C          YNL058C YGL116W YPL242C          YPR156C</p>	<p>Cell division, cytokinesis</p>



<p>35</p>	<p>Cluster 35</p>	<p>Cluster 35</p>	<p>FKH1 FKH2 MCM1 NDD1</p>	<p>YJR092W YLR131C YGL021W YMR001C YPL141C</p>	<p>Mitotic cell cycle, post translational modification, protein</p>
<p>38</p>	<p>Cluster 38</p>	<p>Cluster 38</p>	<p>SWI5</p>	<p>YKL164C YDL117W YKL185W YLR194C YNL078W YIL009W YLR079W YBR083W YJL157C</p>	<p>Reproduction, Filamentous growth</p>

<p>44</p>	<p><b>Cluster 44</b></p>	<p><b>Cluster 44</b></p>	<p>FKH2 SWI4 SWI6</p>	<p>YOR372C YOR313C</p>	<p>Cell cycle phase</p>
<p>46</p>	<p><b>Cluster 46</b></p>	<p><b>Cluster 46</b></p>	<p>SWI5</p>	<p>YML100W YKL043W</p>	<p>Cellular Process</p>

<p>52</p>	<p><b>Cluster 52</b></p> <p>Gene expression values</p> <p>YDR342C YML110C</p>	<p><b>Cluster 52</b></p> <p>Gene regulatory input probability</p> <p>YDR342C YML110C</p>	<p>MCM1</p>	<p>YDR342C YML110C</p>	<p>Cellular Process</p>
<p>54</p>	<p><b>Cluster 54</b></p> <p>Gene expression values</p> <p>YGL055W YDR309C</p>	<p><b>Cluster 54</b></p> <p>Gene regulatory input probability</p> <p>YGL055W YDR309C</p>	<p>RCS1</p>	<p>YGL055W YDR309C</p>	<p>Cellular organization, Cellular process compartment organization</p>

<p>57</p>	<p>Cluster 57</p>	<p>Cluster 57</p>	<p>RCS1</p>	<p>YOR153W YML116W</p>	<p>Drug transport, response to drug</p>
<p>58</p>	<p>Cluster 58</p>	<p>Cluster 58</p>	<p>MCM1</p>	<p>YDR461W YJL079C</p>	<p>Cellular process</p>

<p>59</p>	<p>Cluster 59</p>	<p>Cluster 59</p>	<p>ASH1 MCM1 SWI5</p>	<p>YKL163W YJL159W</p>	<p>Cell wall organization</p>
<p>64</p>	<p>Cluster 64</p>	<p>Cluster 64</p>	<p>MBP1</p>	<p>YOR342C YPL014W</p>	<p>-</p>

<p>70</p>			<p>FKH1 FKH2 NDD1</p>	<p>YOR315W YLR353W YLR190W YDR146C YMR183C YPL155C</p> <p>YGR108W YNL172W YJL051W YPR119W YIL158W</p>	<p>Regulation of microtubule cytoskeleton organization, M phase or mitotic cell cycle, nuclear division</p>
<p>74</p>			<p>SWI5</p>	<p>YDL127W YNL192W</p>	<p>Sexual reproduction, cell division</p>

<p>75</p>	<p>Cluster 75</p> <p>Gene expression values</p>	<p>Cluster 75</p> <p>Gene regulatory input probability</p>	<p>SWI4</p> <p>SWI6</p>	<p>YBL030C YNL289W</p>	<p>Cellular process</p>
-----------	---	--	-------------------------	------------------------	-------------------------