A combinatorial approach to scientific exploration of gene expression data: An integrative method using Formal Concept Analysis for the comparative analysis of microarray data.

Dustin P. Potter

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

Doctor of Philosophy
in
Mathematics

Reinhard Laubenbacher, Chair
Peter Haskell, Co-chair
Karen Duca
Abdul Salam Jarrah

August 3, 2005
Blacksburg, Virginia

Keywords: bioinformatics, Formal Concept Analysis, microarray analysis, integrative methods, combinatorics

A combinatorial approach to scientific exploration of gene expression data: An integrative method using Formal Concept Analysis for the comparative analysis of microarray data.

Dustin P. Potter

Abstract

This work focuses on a method that integrates gene expression values obtained from microarray experiments with biological functional information in order to make global comparisons of multiple experiments. The integrated data is represented as a partially ordered set using methods from Formal Concept Analysis. Appropriate measures defined on such sets are employed to compare a reference experiment to samples from a database of similar experiments, and a ranking of similarity is returned. In this work, the mathematical foundations of the method are presented as well as a fast algorithm for the construction of the representative lattices. The validity of our method is supported by its application to data sets of both simulated and reported microarray experiments.

Dedication

To my favorite alliterations:

EPP,

for the many roads we've walked;

BSS,

for the many roads to come.

Acknowledgments

This finished work is much more than the result of my few years in grad school, but is founded upon almost 32 years of living. In this light I must first give thanks to Michelle and Don without whom there would be no life. Mom, your love has always been a strength and support; it has provided me with the ability to continue even when my body and soul were ready to give up. Dad, the lessons I have learned from you have shaped me in more ways than I will ever know. My Aunt and Grandmother have been two of the most influential women in my life: Gretchen, you taught me to dream and Lois, you taught me that I am worthy of my dreams. My three siblings Danielle, Heather, and Shawn have given me the gift of humility as only those so very close to us can. There is little doubt that the man I am now would not exist if not for greatest gift I have ever been give: Elyjiah my son–thank you for putting up with me through the good and the bad. Though our time together has been relatively short, the support my wife Brandy has provided–emotionally, intellectually, and spiritually–is immeasurable. Brandy, my love for you is stronger now than ever and may it continue to grow in depth and breadth. To my two academic fathers, I cannot express my gratitude enough. Peter Haskell, you have been an incredible role model as a mathematician. Reinhard Laubenbacher, your exuberance for the work you are doing is contagious and I thank you for sharing the bug with me. Karen Duca, thank you for opening up the world of biology. Hélène Barcelo, you have been a fountain of ideas and another great role model as a mathematician. Abdul Salam Jarrah, you are a great friend and mathematician, thank you for your kind ear and great advice (the cigars weren't half bad either). Thank you to all the scientists at VBI; in particular Pedro Mendes, Vladimir Shulaev, and Brett Tyler. The staff at VBI must also be commended for keeping such a behemoth afloat. Nicholas Polys, Satya Root, Hussein Vastani–the three computer wise men. The Math faculty and staff at VT are fantastic and I can't thank you enough for the intellectual support you have provided. Hannah Swiger, without you there would be a few less PhDs in the world. Eileen Shugart, your love for the work you do is apparent and shapes the atmosphere and dynamics of the entire graduate program. The graduate students at VT have been an extended family to me. In particular, I give my love and gratitude to Laurence Calzone, Brian Camp, Omar Colon-Reyes, Edgar Delgado-Eckert, Elena Dimitrova, Brett Enge, Greg Hartman, Micah Leamer, John McGee, José María Menédez, Brendan Meuse, Chris Newman, Vinh Nguyen, Olgamary Rivera-Marrero, Ivan Rothstein, Jesse Seihler, and Paola Vera-Licona. I first set foot on this academic road at the Evergreen State College and so many kudos go to you, my

Alma Mater–go Geoducks!

Contents

List of Figures

List of Tables

Chapter 1

Inroduction

1.1 Extended abstract

Functional genetics is the study of the genes present in a genome of an organism, the complex interplay of all genes and their environment being the primary focus of study. The motivation for such studies is the premise that gene expression patterns in a cell are characteristic of its current state. The availability of the entire genome for many organisms now allows scientists unparalleled opportunities to characterize, classify, and manipulate genes or gene networks involved in metabolism, cellular differentiation, development, and disease.

System-wide studies of biological systems have been made possible by the advent of high-throughput and large-scale tools such as microarrays which are capable of measuring the mRNA levels of all genes in a genome. Tools and methods for the integration, visualization, and modeling of the large-scale data obtained in typical systems biology experiments are indispensable. Our work focuses on a method that integrates gene expression values obtained from microarray experiments with biological functional information related to the genes measured in order to make global comparisons of multiple experiments.

In our method, the integrated data is represented as a lattice and, using appropriate measures, a reference experiment can be compared to samples from a database of similar experiments, and a ranking of similarity is returned. In this work, support for the validity of our method is demonstrated both theoretically and empirically: a mathematical description of the lattice structure with respect to the integrated information is developed and the method is applied to data sets of both simulated and reported microarray experiments. A fast algorithm for constructing the lattice representation is also developed.

1.2 Setting the scene

1.2.1 The historical setting

The biologist seeks to understand both the molecular components that make up all living things as well as their complex interactive networks [94] with the ultimate goal of completely understanding the function of biological organisms [71]. It took Mendel [85] to plant the seeds of our present day understanding of inheritance, Morgan [90] and to point us in the direction of the chromosome for the location of the theoretical gene, and Watson, Crick [125] and Franklin [83] to give us the double-helix blueprint for all living organisms before such noble aspirations could take root and become a possibility. The flow of cellular regulatory information has been established, DNA is transcribed into RNA which is then translated into protein which in turn interacts with the cellular environment (see figure 1.1) [60].

It is the unique underlying structure of deoxyribonucleic acid (DNA) that enables genetic biology. The base pairs – adenine (A), cytosine (C), guanine (G), and thymine (T) – can be

arranged in almost any sequence in the DNA molecule and hence any digital information can be stored in this unique structure. Within ten years of Watson and Crick's initial discovery, Meselson and colleagues showed that the information encoded in DNA is transferred to the rest of the cell by the transcription of each gene into a complementary molecule, called messenger RNA (mRNA) [18]. mRNA is made up of A, C, G, and uracil (U) which replaces T. The 20 amino acids, the building blocks of all proteins, are each comprised of a string of three bases of DNA and RNA: each triplet of nucleic acids (codons) in a gene encodes one amino acid [34]. For example, AGT encode the amino acid serine [61]. Combinatorially, there are $4^3 = 64$ possible codons, 61 of which encode for the 20 amino acids, with obvious redundancy, and 3 of which act as markers that signal the termination of the growing protein chain (see Table 1.1 for the amino acid dictionary).

Table 1.1: The genetic code for amino acids [61]

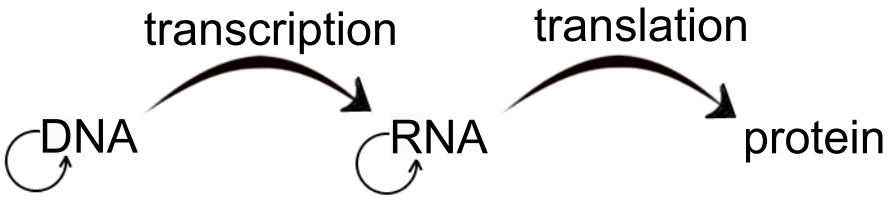
		Second Position					
		U	C	A	G		
First Position	U	phenyl-alanine	serine	tyrosine	cysteine	U	**Third Position**
						C	
		leucine		stop	stop	A	
				stop	tryptophan	G	
	C	leucine	proline	histidine	arginine	U	
						C	
				glutamine		A	
						G	
	A	isoleucine	threonine	asparagine	serine	U	
						C	
				lysine	arginine	A	
		methionine				G	
	G	valine	alanine	aspartic acid	glycine	U	
						C	
				glutamic acid		A	
						G	

With the findings of Crick and colleagues in 1961 [34], the normal pathway of information flow was established: DNA is transcribed into RNA which is then translated into protein (see Figure 1.1). This flow of information suports Crick's central dogma of molecular biology as enunciated in 1958 [119] and published in 1970:

> The central dogma of molecular biology deals with the detailed residue-by-residue transfer of sequential information. It states that such information cannot be transferred from protein to either protein or nucleic acid [35].

Though it is now known that the flow of information is not as linear nor as closed a system as depicted in Figure 1.1 [60], it will suffice for our purposes.

Figure 1.1: The normal pathway of information flow



With the central dogma and the pathway of information flow as a starting point, molecular biologists have branched out in many directions with the goal of understanding the interactive dynamic network of the different molecules that make up the cell. Presently we have a broad and well-founded basic understanding of elementary processes behind heredity, evolution, development, and disease [71]. In the half century since Crick et al.'s description of DNA, thousands of genes have been identified and the function of their transcription determined.

1.2.2 From a reductionist to a systems science

Until very recently all approaches were reductionist in nature [59]: experiments and findings were restricted to local interactions at the molecular level (i.e., a gene or a protein). Discoveries about individual components are stitched together to construct a system-level view of the system whether it be a tissue, an organ, or an organism. Great strides in the understanding of complex organisms as well as advances in medicine have been made with reductionist approach. The original human genome project (which some herald as the hallmark of the systems-biology paradigm shift) was first proposed within reductionism: determine all genes; find out their respective functions; draw biological conclusions; apply them to medical problems [52]. However, we have come to realize that even though the nucleotide sequence of the human genome is known, the original framework of the human genome project was too limited. There is not always a mapping from one gene to one protein with a single function [48]; non-coding sequences play an integral part in gene regulation as do signals from the extra-cellular environment [20]; and auxiliary molecules are central in the transcription of many genes [64].

Though necessary for understanding, reductionist approaches are too cumbersome and limited for many of the biological issues that confront scientists today and hence are not sufficient. For example, Dr. von Eschenbach of the National Cancer Institute stated that because of the large size of cancer networks as well as the lack of knowledge concerning the kinetic

interaction of enzymes in the network, in order to meet the goal of the National Cancer Act – to conquer cancer by 2015 – the reductionist approach must be replaced by an integrative methodology [124]. Motivated by the human genome project and encouraged by multiple areas of biological research, programs such as "transcriptomics", the measurement of different mRNA levels [120]; "proteomics", the large scale study of protein expression structure and function [98]; and "metabolomics", the study of the metabolic profile of a cell [93], have been developed in order to better understand the interaction of all relevant molecular species and the regulation of their synthesis [52]. These new approaches can be characterized as a systems approach to biological understanding as described by Ideker et al.:

> ... it [systems biology] investigates the behavior and relationships of all of the elements in a particular biological system while it is functioning. These data can then be integrated, graphically displayed, and ultimately modeled computationally [63].

The central goal of reductionist biology is to describe the mechanisms of life in terms of simple deterministic principles while the goal of systems biology is to understand the organization, function, and evolution of an organism and its environment [6]. As evidenced by the incredible growth of companies, academic departments, scientific journals, and conferences with the words "systems biology" in the title, there is little doubt that the systems approach to biological research will be a driving mode. In this time of philosophical transition, however, "the baby must not be thrown out with the bath water": the biological discoveries made via traditionally reductionist means are essential to our understanding of life and will be integrated into a more inclusive picture through systems biology. In other words, the understanding of individual or small groups of genes, proteins and metabolites will continue to be important to the advancement of biological understanding; the focus of biological research, however, is shifting to an understanding of the structure and dynamics of an entire system [72].

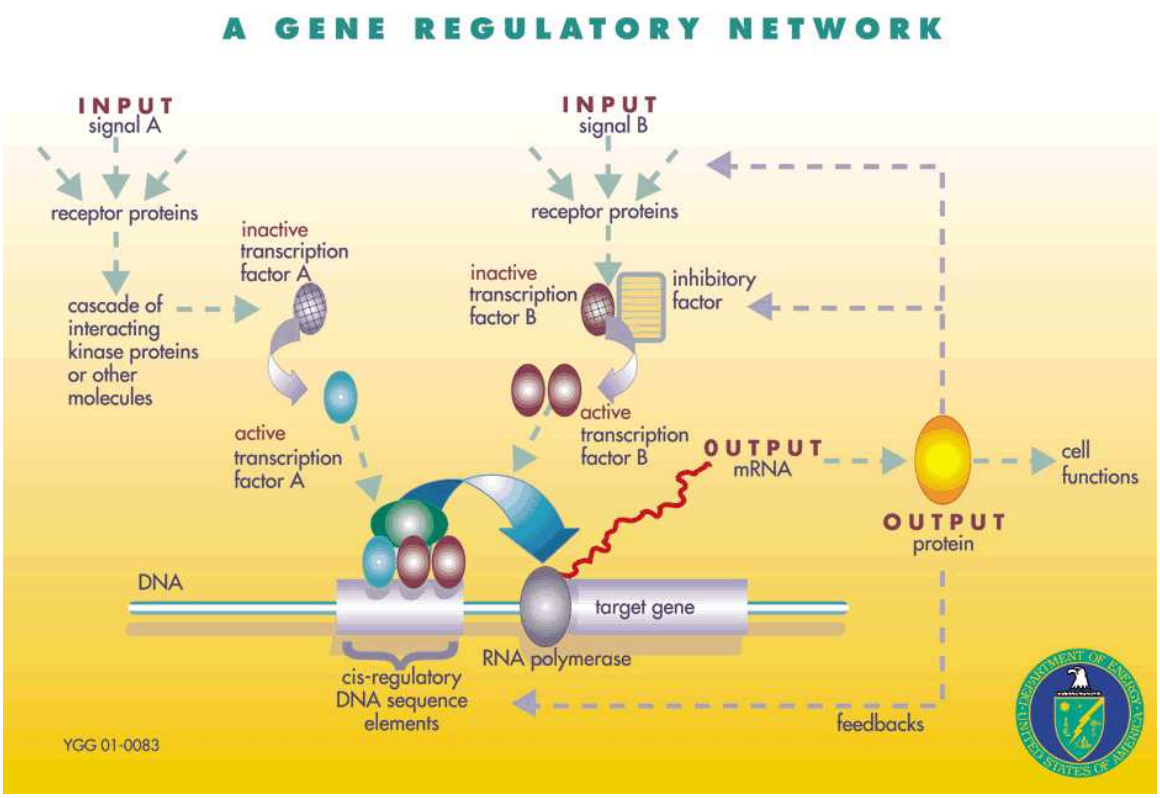
The systems approach to biology, and in general the sciences, is by no means a new methodology. In the late forties, Norbert Wiener's systems-level approach led to biological cybernetics [126]. Ahead of his time, Ludwig von Bertalanffy, in the late sixties, proposed a systems-theory of everything in which biology sat near the center of his hierarchical structure of the sciences with respect to complexity [123]. However, it was not until recently that methods for high-throughput measurement/observation of system-wide levels of mRNA, proteins, and metabolites were developed, allowing for the approaches expounded by Wiener, Bertalanffy, and their colleagues to be realized.

1.2.3 Becoming a quantitative science

The development of tools for system-wide high-throughput measurements has not only allowed for the realization of systems biology but has also encouraged the development of

molecular biology as a quantitative science, as opposed to being solely descriptive [1]. A typical genetic model has traditionally been comprised of either verbal descriptions, a cartoon model, or line diagrams (e.g., Figure 1.2) in which the vertices correspond to genes, proteins, or metabolites and an edge exists between two vertices if there is a biological relationship between the two molecules (e.g., one gene regulates the transcription of the other, a protein is the product of a gene, two proteins bond to form a complex molecule, a metabolite is converted into another through an enzyme, etc.).

Figure 1.2: A typical example of a genetic model. [92]



It should not be misconstrued however that all information at the gene/protein/metabolite level can now be represented quantitatively. Much of the known information with respect to genomics is descriptive or nominal in nature. Following are examples of descriptive characteristics of genes and their transcripts.

Protein Motif: A pattern of amino acids that is conserved across many proteins and confers a particular function. For example, the presence of one particular motif in a protein indicates that the protein probably binds ATP and may therefore require ATP for its action [5].

Gene Ontologies: Ontologies are 'specifications of a relational vocabulary'. In other words, they are sets of defined terms like the sort that one would find in a dictionary, but the terms are networked. The terms in a given vocabulary are likely to be restricted to those used in a particular field, and in the case of GO, the terms are all biological. The Gene Ontology (GO) project is a collaborative effort to address the need for consistent descriptions of gene products in different databases. The GO collaborators are developing three structured, controlled vocabularies (ontologies) that describe gene products in terms of their associated molecular functions, biological processes, and cellular components in a species-independent manner [33]:

> **Molecular functions**: Molecular function describes activities, such as catalytic or binding activities, at the molecular level. GO molecular function terms represent activities rather than the entities (molecules or complexes) that perform the actions, and do not specify where or when, or in what context, the action takes place. Molecular functions generally correspond to activities that can be performed by individual gene products, but some activities are performed by assembled complexes of gene products [32].
>
> **Biological process**: A biological process is a series of events accomplished by one or more ordered assemblies of molecular functions [32].
>
> **Cellular component**: A cellular component is a component of a cell but with the proviso that it is part of some larger object, which may be an anatomical structure (e.g., rough endoplasmic reticulum or nucleus) or a gene product group (e.g., ribosome, proteasome or a protein dimer) [32].

Enzyme Commission (EC) Numbers: EC numbers are a numerical classification scheme for enzymes, based on the chemical reactions they catalyze. As a system of enzyme nomenclature, every EC number is associated with a recommended name for the respective enzyme. Every enzyme code consists of the letters "EC" followed by four numbers separated by periods. Those numbers represent a progressively finer classification of the enzyme. For example, the enzyme tripeptide aminopeptidase has the code EC 3.4.11.4 which is constructed as follows: 3 stands for hydrolases (enzymes that use water to break up some other molecule), 3.4 for hydrolases that act on peptide bonds, 3.4.11 for those that cleave off the amino-terminal amino acid from a polypeptide, and 3.4.11.4 for those that cleave off the amino-terminal end from a tripeptide [127].

1.2.4 Microarray technologies

Genomics has been revolutionized by the advent of recent quantitative means of measuring the levels of mRNA, protein, and metabolites. Though the method proposed in this paper

can theoretically be applied to all three types of high-throughput measurements, our present work focus on the analysis of microarray data (*i.e.*, the measurements of mRNA levels in a sample). Therefore, the continuation of this chapter will focus on the elucidation of microarray technologies and the analysis methods developed for working with the data generated by the technology. The following description is derived from Drăghici's book [41].

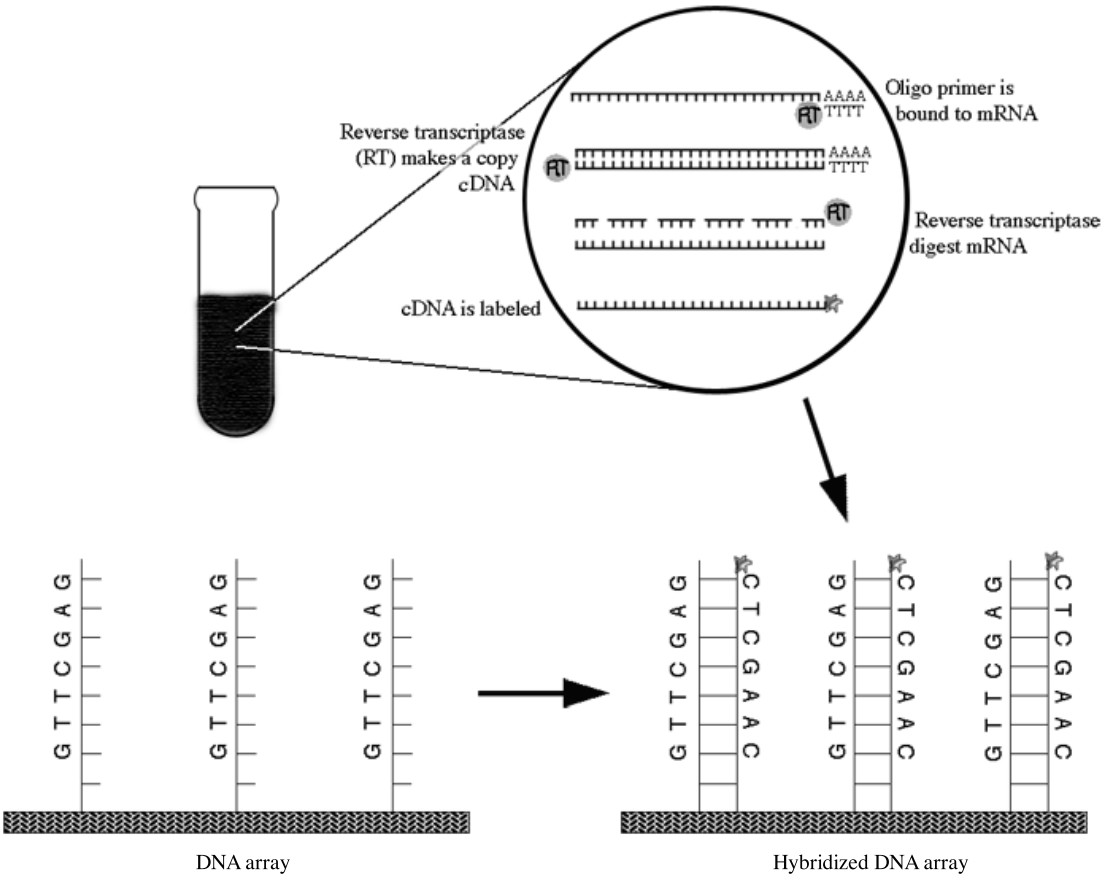
In the most generic of definitions, a microarray, or DNA array, is a synthetic slide (nylon membrane, glass, plastic, or silicon wafer) upon which various sequences of single stranded or complementary DNA (cDNA) are deposited. Usually, the cDNA are adhered to the slide in a regular grid-like arrangement. As proposed by Duggan et al. [44] the cDNA printed on the slides are referred to as probes.

When used in gene expression studies, the probes printed on the arrays typically correspond to reduced portions of known gene sequences. The DNA targets to be tested on the array are obtained by reverse transcription of the mRNA (see Figure 1.3). The targets have an affinity to bind with probes on the slide which have complementary nucleotide sequences. Since the targets are labeled with a fluorescent dye, a radioactive element or by some other detectable element, the hybridized spot on the array can be detected and quantified. The targets are allowed to hybridize to the array, after which time they are washed and scanned. The intensity of dye or radioactivity at each spot is interpreted as being related to the amount of mRNA present in the sample which, by the central dogma, is in turn related to the amount of protein produced by the gene corresponding to the location on the array.

The actual implementation of this process can be carried out in a number of ways. The two most widely used approaches are cDNA two channel and Affymetrix oligonucleotide technologies. In the two channel design, targets are labeled with two distinct dyes (most commonly cy3 and cy5). Typically, targets from a control sample are labeled with one dye and targets from the experimental sample labeled with the other. The ratio of the intensities of the two dyes is then interpreted as the fold-change of gene expression between control and experiment. In the Affymetrix design, there are two types of probes printed on the array: reference probes and mismatch probes. The reference probes match a target exactly. For every reference probe there is an adjacent probe with a sequence that differs from the reference only at the central base position–this probe is called a mismatch. Typically, the average difference between reference and mismatch intensities is taken as the expression value of the gene.

Both technologies have strengths and weaknesses and at this time there is no consensus as to which is superior for gene expression experiments. cDNA technologies are presently more flexible, allowing for the spotting of almost any sequence, while the Affymetrix technology is generally more reproducible [11, 81]. The field of molecular biology will eventually sort this out. For our purposes, the end result of either technology is the same: a quantitative value is assigned to each gene which is interpreted as corresponding in some way to the amount of mRNA expressed by the gene in the system being studied.

Figure 1.3: General overview of the design for a microarray experiment. mRNA collected from a sample are copied into cDNA which are labeled and then washed across a DNA array [41].



1.3 Standard approaches to microarray analysis

There are an estimated 20,000 - 25,000 genes encoded by the human genome, however, cellular responses to environmental changes are determined by the expression of only a small subset of these genes [110]. In many areas of research and medicine, the exact genes responsible for a particular biological function or response are not known. As such, microarrays have become the method of choice for many fields in which it is necessary to profile changes in gene expression levels under different conditions [47]. In the area of medical research, gene expresion signatures shows promise as a means of providing "individualized" treatment for different medical conditions [46]. The pharmaceutical industry uses the technology to monitor and detect potential biomarkers for therapeutic intervention as well as to monitor changes in expression levels in response to a treatment [36, 38]. In the area of cancer research, scientists are using microarrays to define new pathological subclasses for tumors,

discover genetic markers associated with different cancers, and develop means of predicting responses to treatment [14, 13].

Two aspects of microarray data make the analysis of gene expression patterns exceptionally challenging. First, even with recent advances in microarray technology, there is a high variability between slides making difficult the detection of statistically significant variance between expression patterns [112]. The variability is mostly caused by the probe labeling process as well as inconsistent experimental conditions (temperature, hybridization pattern, *etc.*). Second, if p genes are measured with a microarray, then the result is a p-dimensional random vector with mutually dependent components. Cost, both monetary and temporal, makes it prohibitive to generate more than a small number (with respect to p) of replicates. Due to the underdetermination of the system, classical statistical approaches to data analysis are inadequate for working with microarray data.

Existing clustering/classification approaches used for analyzing microarray data can be categorized into two broad categories: supervised and unsupervised [77]. The two approaches are distinguished by the degree to which prior information or hypotheses concerning the measured samples are used in the analysis of the sample. Supervised approaches are most useful when samples can be grouped into different classes via known classifications (for example, normal versus infected tissue). Such approaches can then be used to detect gene signatures or expression patterns that distinguish the different classes as well as provide means for predicting class membership [4, 19, 55, 77]. When little prior biological information concerning the samples is known, unsupervised methods are most appropriate for microarray data analysis [17]. Such approaches are important for discovering biological mechanisms as well as genetic regulatory systems [116, 22, 107]. Since the method proposed in this paper is an unsupervised analysis tool, we will further expound these techniques.

Unsupervised methods can be further decomposed into three categories: clustering approaches, model-based approaches, and projection approaches. Clustering approaches use similarity of behavior as the criterion for grouping genes and experiments [39, 22], making the data easier to analyze [67]. The broad hypothesis of clustering methods is that genes that are similarly expressed under similar experimental conditions are functionally related. Model-based methods propose a model, or family of models, that best describes the interactions between the biological entities in the system as well as experimental interactions (such as dye effect and chip location); the data sets are then used to train the parameters [129, 49, 39]. Discovery of differentially expressed genes is the most common use of modeling methods for the analysis of gene expression data. Projection methods decompose a data set into components that have desired properties [82, 2]. Most projection methods employed for microarray data analysis are based on either principal component analysis (PCA) or independent component analysis (IPA) [77] and are often used to either reduce the dimensionality of the data so as to employ other analysis tools or as a clustering technique.

1.3.1 Clustering methods

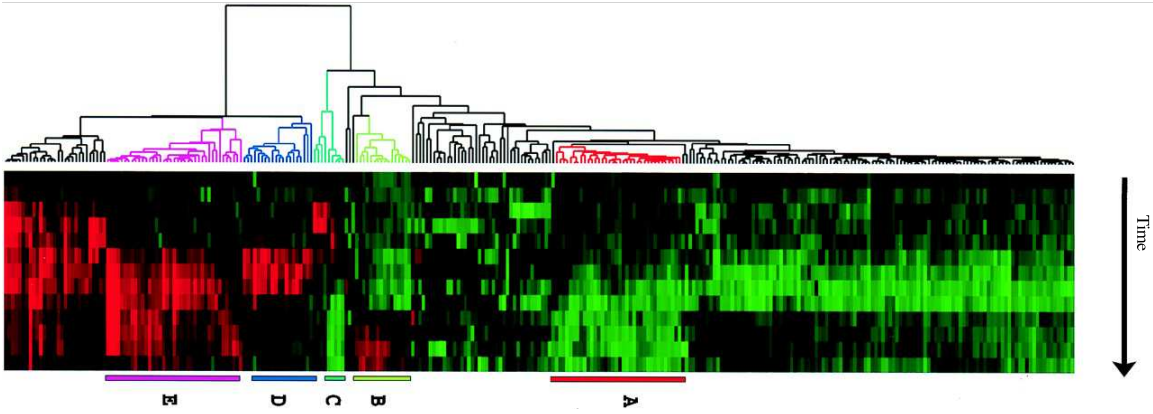
Scoring

The most straightforward approach for identifying and grouping genes of interest is to apply a scoring method. For example, each gene in the experiment can be assigned a value based on (a) whether its expression level is significantly changed in one experimental condition, (b) whether, over all experimental conditions, there has been significant aggregate change, (c) whether the expression pattern across experiments shows high diversity with respect to established information criteria [39]. In a comparative analysis of different scoring methods, Cunningham *et al.* [36] have shown that scoring based on Shannon entropy allows for the most conclusive distinction of drug-specific expression patterns .

Hierarchical clustering

Beyond general scoring methods, approaches based on a "guilt by association" (GBA) criterion are the simplest techniques for clustering genes or samples with possibly shared functional or regulatory signatures. Hierarchical clustering, a method long employed by the biological community for generating phylogenetic trees, is the most widely used GBA method [112]. Relationship among the genes (or samples) are represented by a tree with leaves labeled by the genes (samples) and branch lengths reflecting the degree of similarity between genes (samples) based on some metric defined on the sample space (see Figure 1.4) [45].

Figure 1.4: Example of hierarchical clustering of DNA microarray data [45]



As noted by Eisen *et al.* [45], when working with complex data sets, it is natural to first scan for large-scale features and then focus on the more interesting details. Hierarchical clustering organizes the data in a way that enables such an intuitive approach to analyzing them. Since few assumptions about the nature of the data are required, hierarchical clustering is an ideal approach for hypothesis-free data analysis.

However, hierarchical clustering does have a number of deficiencies as a method for gene expression analysis. Gene expression patterns can be similar in multiple distinct ways, however hierarchical trees are only capable of reflecting singular relationships [113]. It has also been noted that hierarchical clusters are not unique nor are they robust to noise and outliers [89]. Finally, hierarchical trees are best suited for representing data that has an underlying hierarchical structure; however, there is no support for such relationships in biological functions of different genes [112].

Self-organizing Maps

The method of self-organizing maps directly partitions genes into clusters thus avoiding some of the difficulties inherent in hierarchical clustering. First applied to expression data by Tamayo *et al.* [113], self-organizing maps construct ordered low-dimensional representations of a data space. In the approach, each gene is initially assigned a vertex in an n-dimensional lattice. Random vertices are selected and iteratively adjusted in the n-dimensional space according to the gene expression pattern. In the end, genes that belong to clusters that are close to each other in the geometry of the lattice are more similar (at least in terms of expression pattern) than genes that belong to clusters that are farther away from each other.

As with hierarchical clustering, no prior information about the distribution of the data is necessary for confidently constructing self-organizing maps. Also, the results are easy to visualize and, as with hierarchical clustering, allows for intuitive analysis of the data. The clustering algorithms are relatively fast and scalable to large data sets [111]. In a comparison with hierarchical clustering, Mangiameli *et al.* [84] found self-organizing maps more robust when used to analyze data sets with dispersion, irrelevant variables, outliers, and non-uniform densities.

The strength of self-organizing maps is also a weakness. The method does not require any knowledge concerning the distribution of the data, hence no statistical tools exist for providing a theoretical foundation. Without a statistical model underlying the method, no measure of confidence can be assigned to the findings [112].

K-means clustering

Similar to self-organizing maps, K-means clustering partitions the genes into distinct non-overlapping groups. However, in K-means clustering, the number of partitions must be decided upon beforehand and is not determined by the data. Initially, K centroids $c_1, \ldots, c_k$ are randomly chosen. Each expression value x_i is assigned to the cluster with centroid c_j such that the squared distance $||x_i - c_j||^2$ is minimized. The average value of each cluster is computed and becomes the new centroid values. The genes are re-assigned to the cluster with centroid that minimizes the squared distance from its expression value. This process is repeated until the centroid values do not change [29].

K-means clustering has many of the same positive attributes as self-organizing maps (*i.e.*, visually accessible and no prior information required). Tavazoie et al. [115] demonstrated the promise of K-means clustering with their work with *Saccharomyces cerevisiae* when they generated clusters, unbiased by prior biological knowledge, that were significantly enriched with genes of similar function.

This method seeks to minimize the least squares distance between gene expression values and the centroid of the cluster to which the genes are assigned. This is a combinatorial optimization problem and as such the global optimum will most likely not be found [29]. Different initializations therefore will lead to different clusters. As with self-organizing maps, there is no statistical theory underlying the method and therefore no means of assigning confidence [112].

These three are, by far, not the only clustering techniques that have been applied to microarray data; they are simply the most common. Other approaches are, but not limited to, independent and principal component analysis (see Section 1.3.3), supermagnetic fields and Superparamagnetic Clustering [40, 51], block clustering [43], coupled two-way clustering [114], and hybrid clustering methods [130].

1.3.2 Statistical modeling methods

A model is a mathematical idealization that is used as an approximation to represent the outputs of a system, and a statistical model represents the outputs in terms of probabilities [106]. The intensity values of the probes on a slide comprise the outputs of microarray models and the parameters used range from rates of change due to response to gene chip interaction to response to hybridization [78]. Models employed as well as the distribution of their parameters depend on the technology used to measure the expression levels. For example, a simple, but useful model, used for microarray data collected with Affymetrix chips would be:

$$y_{ij} = PM_{ij} - MM_{ij} = \theta_i \phi_j + \varepsilon_{ij}$$

where y_{ij} is the j^{th} probe for the gene in the i^{th} sample, MM_{ij} and PM_{ij} denote the intensity values of the mismatch and perfect match probes respectively, θ_i is the expression value of the gene, ϕ_j is the rate of increase of the PM response, and ε_{ij} is the error [78]. Obviously the parameters would have to be changed if the model were to work with measurements from a two-channel array.

Another generic model useful for detecting differentially expressed genes in an experiment with two conditions and K_1 and K_2 replicates of the first and second condition is:

$$Y_{jk} = \alpha_j + \beta_j \Upsilon_{K_1}(k) + \varepsilon_{jk}$$

where Y_{jk} is the j^{th} genes expression value on the k^{th} array, $\Upsilon_{K_1}(k) = 1$ if $k \leq K_1$ and zero otherwise, and ε is the random error. Then $\alpha_j + \beta_j$ and α_j are the mean expression levels of the j^{th} gene under the two respective conditions. Determining differential expression between conditions then reduces to testing for the null hypothesis

$$H_0 : \beta_j = 0 \text{ against } H_1 : \beta_j \neq 0$$

[95].

It is often assumed that in microarray experiments, the factors modeled are not independent and have a synergistic interaction in which the combined effect of the two or more factors at the same time is greater than the sum of their individual effects [41]. Two-way factorial designs model this assumption. A basic example would be:

$$y_{ijg} = \mu + A_i + D_j + G_g + (AD)_{ij} + (AG)_{ig} + (DG)_{jg} + \varepsilon_{ijg}$$

where y_{ijg} is the expression value of the gene on the i^{th} array, on the j^{th} channel under the g^{th} treatment; μ is the average expression value; A is the array effect; D is the dye effect, G is the gene effect, and ε is the random error.

Other models use the above models as a theoretical starting point and then define more sophisticated models to analyze the behavior of genes. Such models fall into two broad categories, those that handle data from a single array with only one gene per slide, and those that can work with (or require) multiple or replicate expression values for each gene [96]. Lee et al. [76] demonstrated that due to high signal-noise ratio replicates are necessary in order to guarantee confidence in any statistical finding . More important, some have demonstrated differential variability among various genes, hence replicates are necessary to assess the different variability [62, 91]. The three general techniques outlined below work with replicate microarray data.

ANOVA

Probably the most widely used statistical model for discovering differentially expressed genes, ANalysis Of VAriance (ANOVA) is a collection of statistical models that separates the observed variance of an experiment into assignable causes [41]. Related to the t-test, ANOVA tests between the means of two or more groups. However, unlike the t-test, the probability of making a type-I error (the null hypothesis is rejected even though it is true) does not increase with multiple tests [97].

A key of the ANOVA method is that it systematically estimates the parameters of a factorial design based on all relevant data as opposed to a piecemeal approach [68]. The method also allows one to study the dependence and variance of signal relative to different positions on

the array [105]. As an analysis tool, ANOVA has been widely used to estimate confidence intervals of expression levels [69, 118].

ANOVA models make assumptions that are not satisfied by microarray data. Two key assumptions are that the data is normally distributed and that the genes are independent. There is no experimental support for the first assumption, and the second assumption is clearly false. The key strength of ANOVA, however, is that it estimates all factors involved in the design as well as provides a means of estimating the quality of the results [41].

Normal mixture model

The application of mixture models for gene expression analysis has generally been with experimental designs in which two conditions are investigated and there are m and n replicate arrays measuring expression under two conditions X and Y respectively [95]. A t-type score Z_i is applied as the test statistic [118, 95]:

$$Z_i = \frac{\bar{X}_i - \bar{Y}_i}{\sqrt{v_{xi}/m + v_{yi}/n} + a_0}$$

for gene i where $\bar{X}_i$ and $\bar{Y}_i$ are the sample means, v_{xi} and v_{yi} are the sample variance, and a_0 a constant that stabilizes the denominator. To obtain an estimation of the distribution of the Z_i, Pan et al. [95] suggest permuting the labels and generating test statistics Z_i^σ for each permutation σ. The distribution of the Z_js corresponding to genes for which the null hypothesis is true (*i.e.*, the gene is not differentially expressed in the two conditions) has the same distribution as the Z_i^σs. By this observation, if f is the distribution of all the Z_is, f_0 is the distribution of all the Z_i^σs, and f_1 is the distribution of the Z_js for which the null hypothesis is not true, then

$$f = p_0 f_0 + p_1 f_1$$

where p_1 is the proportion of the genes which are differentially expressed and $p_0 = 1 - p_1$. A Normal mixture model is then used to estimate f and f_0:

$$f_0(z; \Omega_{g_0}) = \sum_{i=1}^{g_0} \pi_i \phi(z; \nu_i, V_i)$$

where $\phi(\cdot; \nu_i, V_i)$ is the normal density function with mean ν_i and variance V_i, the π_i are mixing proportions, and the Ω_{g_0} represent all unknown parameters. The distribution f is estimated using the same type of model.

The use of Normal components in a mixture distribution is natural when working with continuous data [95]. Tail probabilities are estimated with greater stability with Normal

mixture models rather than other standard kernel and local likelihood estimators [96]. Tail probabilities play a central role in determining statistical significance for the test statistic, hence the power of Normal mixed models.

Any statistical model makes assumptions concerning the distribution of its factors that may not be biologically supported, and Normal mixed models are no exception. The random errors are assumed to have symmetric distribution about 0 and the biological random errors all have the same distribution, up to suitable standardization [95]. The model also relies on subtle assumptions about the dependency between the genes that are difficult to verify [108].

Empirical Bayes model

Originally introduced by Newton et al. [91], empirical Bayes models begin with the same test statistic Z_i and distribution description, $f = p_0 f_0 + p_1 f_1$, as the Normal mixture model. Bayes' rule is then used to give the posterior probabilities $p_0(Z)$ and $p_1(Z)$ for determining differentially expressed genes:

$$p_0(Z) = \frac{p_0 f_0(Z)}{f(Z)}$$

and

$$p_1(Z) = 1 - p_0(Z).$$

The key difference is that instead of trying to estimate the overall proportion of differentially expressed genes, p_1 becomes an indeterminate parameter of the model [108]. In application, the overall proportion of differentially expressed genes, both detected and not, must be specified in advance. Adjusting this value will affect the posterior odds. However, the order in which the genes are ranked will remain unchanged.

The hypotheses concerning the distribution of random errors are identical to those for mixture models and so empirical Bayes models suffer from the same difficulty of working with assumptions that are difficult to validate. The model also requires multiple replicates in order to guarantee confidence in the results; this is still a limitation due to the prohibitive costs of microarray experiments.

There are probably as many statistical modeling methods applied to microarray data as there are statisticians working on the problem, hence the above is by far not a complete list of all modeling methods in use but provides an idea of the techniques that exist. Other promising modeling methods consist of, but are not limited to, the following: ratio-based decision methods [28], hierarchical models [91], maximum-likelihood analysis [62], gene shaving [58], and plaid models [75].

1.3.3 Projection methods

Projection methods decompose a data set into components that have desired properties. The data is linearly decomposed to reduce the dimensionality of the data in order to apply other analysis methods. As discussed above, they can also be used as a clustering approach. The two widely used methods are principal component analysis (PCA) and independent component analysis (ICA).

Principal Component Analysis

Formally, PCA is a linear transformation of the data that chooses an orthogonal basis for the feature space so that the k^{th} basis element points in the direction of the k^{th} greatest variance of the data [16]. The basis is constructed by applying singular value decomposition (SVD) to the covariance matrix of the data [132], that is, the principal components are the eigenvectors of the covariance matrix. In classic applications of PCA, only the first few components are used to represent the data since they capture the most variance and the last few components are assumed to capture only 'noise' in the data [132]. Jolliffe [65] proposes that PCA is the optimal dimension-reduction technique with respect to the sum of squares.

As a clustering method, PCA is used to reduce the dimensionality of the data. A clustering method, such as K-means or self-organizing maps, is applied to the first few principal components. The assumption is that the principal components may extract cluster structure from the data set [132]. There are theoretical results, however, that suggest that this may not be a valid assumption; Chang [26] showed that theoretically, the first few principal components may even contain less cluster information than other components. Yeung and Ruzzo [132] have demonstrated empirically that there is not a measurable benefit to clustering principal components as opposed to clustering the data directly.

Alter et al. [2] and Misra et al. [88] describe the use of PCA for representing microarray data as a means of analyzing the data directly. Components that are inferred to represent noise are filtered out of the data. PCA is then used to construct a picture of the system-wide dynamics of the data in which genes with biologically equivalent function or regulation have mathematically equivalent component signature. A similar analysis can be applied when components correspond to samples or experiments instead of genes, in which case comparable component signatures are interpreted as similar cellular state or biological phenotype.

PCA is a strong explorative tool when little prior knowledge concerning the data is known. However, since the focus of exploration is shifted from the analysis of individual genes to combined effects of genes, the method is not suitable as a classifier nor for detecting differentially expressed genes [88].

Independent Component Analysis

As with PCA, independent component analysis (ICA) is a method for separating data into additive subcomponents. However, in ICA, the components are constructed so that each component is statistically as independent from the others as possible and are not necessarily orthogonal [77]. In the linear model, it is assumed that the observed data can be linearly decomposed as

$$x = \sum_{k=1}^{n} a_k s_k$$

where the vector $s = (s_1, \ldots, s_n)$ of observed signals maximizes some function of independence. Nonlinear models have also been developed in which

$$x = f(s; \theta),$$

where $f(\centerdot; \theta)$ is some nonlinear mixing function.

Liebermeister [79] first applied linear ICA for microarray analysis and Lee et al. [77] have applied nonlinear models. In both applications, the authors used the method to detect differentially expressed genes as well as attempt to characterize cell behavior with respect to the different components. Lee et al. found that using nonlinear ICA outperformed many leading clustering methods with respect to consistently generating clusters composed of functionally related genes.

In order for ICA to determine a unique solution, it must be assumed that the samples are linearly independent, that the number of observations is greater than or equal to the number of independent sources, and that at most one of the signals has Gaussian distribution. The first and second conditions are not prohibitive when modeling gene expression as a combination of biological processes s_k. However, they are too restrictive if one wants to model cellular or sample response since it is assumed that molecular components of the cell are related (possibly linearly) and the forbidding expense of microarray experiments makes it impossible to make as many measurements as there are active molecular components. Since biological processes are assumed to be highly non-Gaussian [77], the third assumption is acceptable for working with gene expression data.

1.3.4 Novel methods with broader goals

Though clustering methods are capable of grouping samples according to a measurement of similarity, they have predominantly been successful at distinguishing genes which are similar in biological functionality or expression signature. PCA has been suggested as a means

for making global comparisons of microarray data, but, as a search of the PubMed database demonstrates [102], the majority of work has been in applying the method towards comparative analysis of gene response under multiple conditions. Recently two novel approaches have been proposed for directly comparing cellular response measured via microarray technology.

BlastSets

Proposed by Barriot et al. [10], BlastSets is a method in which directed acyclic graphs are created that capture particular attributes for a given set of genes (e.g. chromosome location, EC number, expression profile). The basic principles of the approach are as follows:

- sets of nucleotide sequences are used as a data structure,
- biological knowledge with respect to genes is converted into sets of sequences,
- the sets of sequences are stored in a database which supports public access as well as the import of new data,
- similarity between sets is measured via a hypergeometric probabilistic model.

Barriot et al. have demonstrated that the method is particularly well suited for the analysis of hierarchically clustered expression profiles. A publicly available web interface for the developed BlastSets software is available (http://cbi.labri.fr/outils/BlastSets/).

According to the authors [10], BlastSets is the first method that is capable of analyzing the integration of bio-molecular information at the cellular level. A strength of the method over standard clustering techniques is that the groupings of genes (*i.e.*, nucleotide sequences) are not independent but instead there are often inclusion relationships between the sets. Since the measure of similarity is a probabilistic model, confidence levels can be assigned to any findings by the method.

Though the BlastSets method does use heterogeneous information for comparisons, it does not integrate the data into a comprehensive model but rather compares groupings of genes that respect different biological characteristics. Within the BlastSets paradigm, once comparisons of different structures have been performed, analysis tools for further inspection of the results are not available.

Meta-Analysis

Rhodes et al. [103] and Khan et al. [70] have proposed a meta-analysis of multiple microarray data sets collected at different institutions that address similar hypotheses. The goal of the approach is to statistically assess all of the results from multiple institutions

simultaneously. The method is capable of removing artifacts of individual studies and yields sets of differentially expressed genes across experiments that are candidates for constructing cellular pathways or transcription factor networks.

The Meta-Analysis approach is complex with multiple steps; however, the general idea is that (1) groups of differentially expressed genes in each experiment are first detected and are used to define differential expression signatures, (2) the expression distributions for the different signatures are estimated via standard bootstrap methods, and (3) expression signatures are intersected, and statistically significant intersections are established and used to identify transcription profiles characteristic of the multiple samples and hypotheses tested.

The Meta-Analysis method has a well-defined statistical model underlying its approach and therefore measures exist for determining confidence in the results. Though the approach is designed for global comparisons of microarray data and can handle the difficulty of comparing results from different sources, its end goal is the discovery of differentially expressed genes across multiple experiments as opposed to identifying cellular response signatures determined by functionality as well.

1.3.5 Summary

As is evident in the abbreviated discussion above, there are myriad methods that have been developed or adapted to work with the unique data types that is produced in microarray experiments. Noticeably absent in all but a few approaches are techniques capable of integrating expression data with other types of heterogeneous biological data. Similarly missing are methods designed to make global comparisons of the cellular response; instead, most techniques are capable only of comparing gene response and detecting differentially expressed genes. It is imperative for the development of systems biology that methods be developed that are capable of incorporating knowledge and information from different domains and/or of making system wide comparisons of multiple experiments.

We propose a method capable of integrating data obtained from gene expression experiments and *a priori* biological characteristics or functions associated with measured genes; the integrated data is represented as a single mathematical object, *i.e.*, a concept lattice. The mathematical representation lends itself to biological discoveries in that relationships between genes, based on expression levels, genetic information, or both, are encoded in the lattice structure. In our method, the integrated data can also be realized as an acyclic directed graph and as such, employing appropriate graph measures, a reference experiment can be compared to samples from a database of similar experiments, and a ranking of similarity is returned. Indeed, we have christened our method microBLAST due to its operational resemblance to the popular BLAST tool for comparing a sequence of interest to a large database of sequences [3]. Also, like BLAST, using a reference gene expression pattern of interest, our method may be tuned to identify global similarity or local similarity from among comparable experiments in a database.

The remainder of this thesis will be focused on the foundation, description, analysis, and empirical support of the microBLAST method. The proposed method is mathematically founded in the theory of Formal Concept Analysis (FCA) and, more broadly, combinatorics. In order for this work to be self contained, the second chapter is devoted to introducing the reader to the basic theory of FCA as well as providing specific results necessary for later discussions. In Chapter 3, the microBLAST method is rigorously described, comparative measures are defined, and theoretical results are presented. In the fourth chapter, we present empirical results in support of our method as well as a comparison of microBLAST to other existing microarray analysis approaches. In the final chapter, we close with a brief discussion of future directions of our work.

Chapter 2

Formal Concept Analysis

The first section of this chapter provides the fundamental definitions and theorems of Formal Concept Analysis (FCA), as established by Wille and Ganter [50]. The second section explores FCA operations employed in the microBLAST method. In the next section, a new algorithm for constructing concept lattices is developed. The chapter closes with a brief description of the FCA community. All definitions in this chapter can also be found in [50]. Theorems, if not original, are appropriately cited.

The central ideas of Formal Concept Analysis revolve around the notion of a *formal context* and a *formal concept*. Of interest is the duality called *Galois connection* that arises naturally in different contexts. This duality is often observed between sets whose elements are related, such as objects and their attributes. In a Galois connection between two sets, the increase in size of one set corresponds to the decrease in size of the other set and vice versa. For example, an increase in the number of search terms used in a Google query corresponds, in general, to a decrease in the number of hits.

The use of the adjective "formal" is to stress that we are working with abstract mathematical objects which only reflect the notions behind the words "context" and "concept". For ease of reading however, we will often omit the word "formal" when discussing *contexts* and *concepts* provided it does not lead to confusion.

2.1 Contexts and concepts

Definition 2.1.1. A *formal context* is a triple (O, A, I) where O is called the set of *objects*, A the set of *attributes*, and $I \subset O \times A$ is the binary relationship or *incidence* between O and A ($(o, a) \in I$ is read "Object o has attribute a").

Example 2.1.1. Consider the set of objects $P = \{$Ivan Boesky, Bernard Ebbers, John Ehrlichman, Dennis Kozlowski, John McDougal, Michael Milken, Martha Stewart$\}$ of infamous white collar criminals and the set of attributes $C = \{$Conspiracy, Fraud, Grand Larceny, Insider Trading, Obstruction of Justice, Perjury$\}$ their possible crimes. With $G \subset P \times C$ the relation such that $(p, c) \in G$ if and only if the person p was found guilty of the crime c, (P, C, G) is a formal context.

A convenient way of representing a formal context is with a *cross table* which is a rectangular table with its rows labeled by the objects, the columns labeled by the attributes, and a cross in an entry if the corresponding pair is incident. The cross table for the above context (P, C, G) is presented in Table 2.1.

Definition 2.1.2. For a set $M \subseteq O$ of objects and a set $N \subseteq A$ of attributes we define

$$M' = \{a \in A \mid (m, a) \in I \text{ for all } m \in M\}$$

to be the *intent* of M. Similarly, we define

$$N' = \{o \in O \mid (o, n) \in I \text{ for all } n \in N\}$$

Table 2.1: Cross table for white-collar criminal context. C = Conspiracy, F = Fraud, GL = Grand Larceny, IT = Insider Trading, OJ = Obstruction of Justice, and P = Perjury.

$(P, C, G) =$

	C	F	GL	IT	OJ	P
Boesky				X		
Ebbers	X	X				X
Ehrlichman	X				X	X
Kozlowski	X	X	X			
McDougal	X	X				
Milken		X		X		
Stewart	X				X	X

to be the *extent* of N.

In words, the intent (extent) of a set X is the set of attributes (objects) maximal with respect to being incident to every element in X. In situations where the incidence relationship is not clear, we will use the incident variable to replace the $'$ notation; *i.e.*, for a context (O, A, I), $X^I = X'$. This notation will be useful when we are working with multiple contexts. The set of all intents and extents are denoted by $\mathfrak{E}(O, A, I)$ and $\mathfrak{I}(O, A, I)$ respectively.

Definition 2.1.3. A *formal concept* of the context (O, A, I) is a pair (M, N) with $M \subseteq O$, $N \subseteq A$, $M' = N$, and $N' = M$. The set of all concepts is denoted by $\mathfrak{B}(O, A, I)$.

For notational ease, $ext(C)$ and $int(C)$ denote the extent and intent of the concept C respectively.

Example 2.1.2. Continuing Example 2.1.1, setting

$$M = \{Obstruction\ of\ Justice\}' = \{Ehrlichman, Stewart\}$$

and

$$N = M' = \{Conspiracy, Obstruction\ of\ Justice, Perjury\},$$

then

$$N' = \{Ehrlichman, Stewart\}$$

and hence, (M, N) is a concept.

The above example suggests several relationships between intents and extents which are captured in the following proposition.

Proposition 2.1.1 ([50]). If (O, A, I) is a context, $M, M_1, M_2 \subseteq O$ are sets of objects and $N, N_1, N_2 \subseteq A$ are sets of attributes, then,

1. $M_1 \subseteq M_2 \Rightarrow M_2' \subseteq M_1'$
2. $N_1 \subseteq N_2 \Rightarrow N_2' \subseteq N_1'$
3. $M \subseteq M''$
4. $N \subseteq N''$
5. $M' = M'''$
6. $N' = N'''$
7. $M \subseteq N' \iff N \subseteq M' \iff M \times N \subseteq I$

Proof.

1. If $m \in M_2'$ then $(o, m) \in I$ for all $o \in M_2$ in particular $(o, m) \in I$ for all $o \in M_1$ since $M_1 \subseteq M_2$. Hence, $m \in M_1'$.
2. This follows from an argument identical to that in 1.
3. If $m \in M$ then $(m, n) \in I$ for all $n \in M'$ which implies $m \in M''$.
4. This follows from an argument identical to that in 3.
5. From 4. we know that $M' \subseteq M'''$. From 3., $M \subseteq M''$ which implies by 1. that $M''' \subseteq M'$.
6. This follows from an argument identical to that in 5.
7. If $M \subseteq N'$ then $N \subseteq N'' \subseteq M'$. If $N \subseteq M'$ then $(m, n) \in I$ for all $m \in M$ and $n \in N$ which implies that $M \times N \subseteq I$. Finally, If $M \times N \subseteq I$ then for all $n \in N$, $(m, n) \in I$ for any $m \in M$ and hence $M \subseteq N'$

□

Proposition 2.1.1 demonstrates that the prime operator forms a Galois correspondence between the power-set lattices $\mathfrak{P}(O)$ and $\mathfrak{P}(A)$. It also follows that for any $M \subseteq O$ and $N \subseteq A$, (M'', M') and (N', N'') are always concepts. This provides an algorithm for generating $\mathfrak{B}(O, A, I)$. First, for each element of the object set, the extent of the intent is computed. For each set created in the previous step, an object not in the set is added and the extent of the intent of the new set is computed. This process is repeated until all objects have been incorporated. Such an explanation is difficult to parse, so the pseudo-code is provided in Figure 2.1. The algorithm is by far not the most efficient; however, it does provide an easy method for building all the concepts by hand for a small context.

Figure 2.1: Pseudo-code for generating $\mathfrak{B}(O, A, I)$

INPUT $\leftarrow (O, A, I)$

OUTPUT $\leftarrow \bigcup_{o \in O} (\{o\}'', \{o\}')$

for each $X \in$ OUTPUT

    for each $o \in O \backslash ext(X)$

        $Y \leftarrow ext(X) \cup \{o\}$

        IF $(Y'', Y') \notin$ OUTPUT

            THEN append(OUTPUT,(Y'', Y'))

return OUTPUT

Definition 2.1.4. If C_1 and C_2 are concepts, C_1 is called a *subconcept* of C_2 provided $ext(C_1) \subset ext(C_2)$ (which is equivalent, by Proposition 2.1.1, to $int(C_2) \subset int(C_1)$). Under the same conditions, C_2 is called a *superconcept* of C_1 and we write $C_1 < C_2$. Then, $(\mathfrak{B}(O, A, I), <)$ is a partially ordered set that is denoted by $\underline{\mathfrak{B}}(O, A, I)$ and is called the concept lattice of the context (O, A, I).

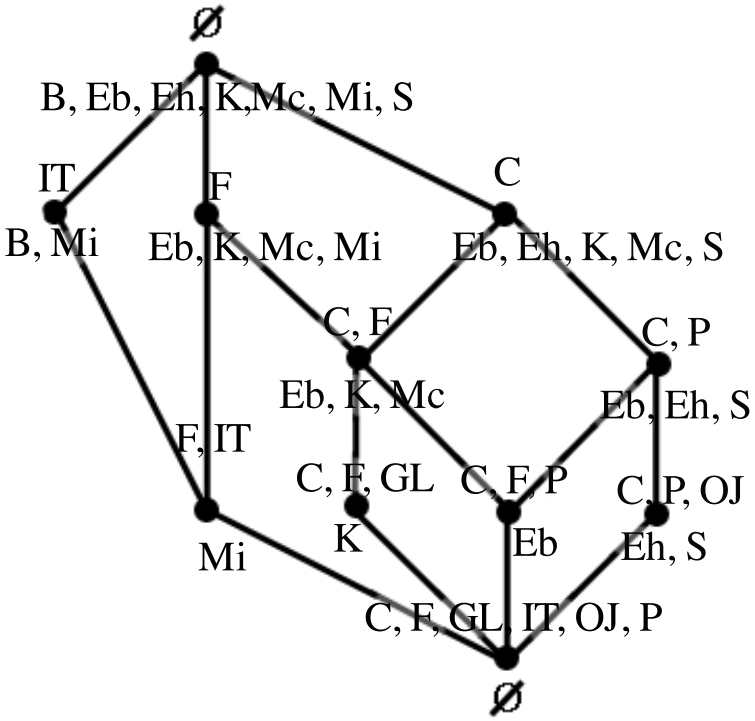
Example 2.1.3. The context in Example 2.1.1 has 11 concepts. The Hasse diagram in Figure 2.2 represents the concept lattice of this context.

The Hasse diagram in Figure 2.2 labels each node by the concept it represents. For large or complex concept lattices, this can lead to a very messy and confusing diagram. The labeling can be simplified considerably by using each object and each attribute to label the concept which it generates (*e.g.*, the object o would be a label for the concept $(\{o\}'', \{o\}')$). See Figure 2.3 for the reduced representation of the Hasse diagram in Figure 2.2. The intents and extents of each concept can still be read off of the reduced representation. For a node v in the diagram, the extents of the concepts can be read off from the label of v as well as the labels of any node that can be reached by a descending path from v. Similarly, the intents of the concept can be read off from the label of v as well as the labels of any node that can be reached be an ascending path from v.

The following theorem demonstrates that $\underline{\mathfrak{B}}(O, A, I)$ has been named appropriately and is a lattice.

Theorem 2.1.1 ([50] The Basic Theorem on Concept Lattices). The concept lattice $\underline{\mathfrak{B}}(O, A, I)$ is a complete lattice in which the infimum and supremum are given by

Figure 2.2: Concept lattice for the white-collar criminal context in Example 2.1.1. B = Boesky, Eb = Ebbers, Eh = Ehrlichman, K = Kozlowski, Mc = McDougal, Mi = Milken, S = Stewart. The concepts in the diagram are labeled such that the top label corresponds to the extent of the concept and the bottom label to the intent of the concept.



$$\bigwedge_{t\in T}(M_t, N_t) = \left(\bigcap_{t\in T} M_t, \left(\bigcup_{t\in T} N_t\right)''\right)$$

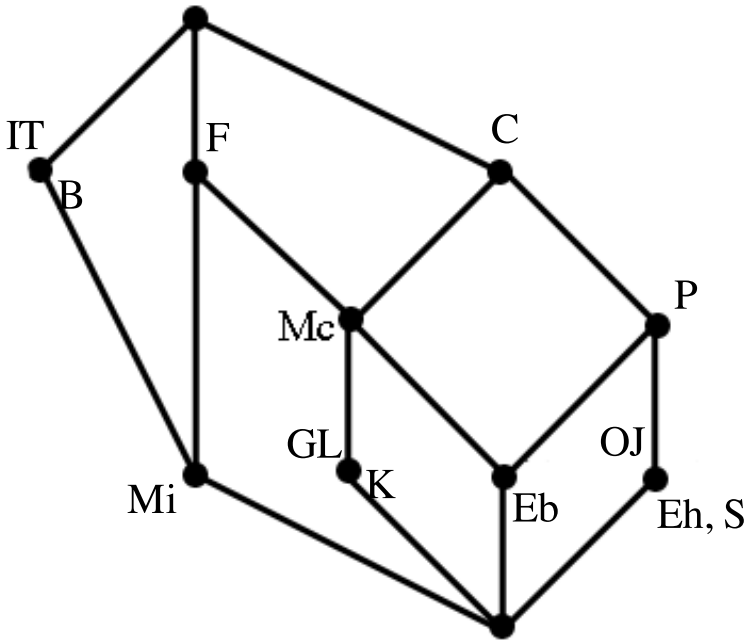
$$\bigvee_{t\in T}(M_t, N_t) = \left(\left(\bigcup_{t\in T} M_t\right)'', \bigcap_{t\in T} N_t\right)$$

To prove the Basic Theorem, we need a technical lemma.

Lemma 2.1.1. If T is an index set and, for every $t \in T$, $M_t \subseteq O$ is a set of objects (respectively $M_t \subseteq A$ is a set of attributes), then

$$\left(\bigcup_{t\in T} M_t\right)' = \bigcap_{t\in T} M_t'$$

Figure 2.3: Hasse diagram with reduced labeling



Proof. We will prove the result for a set of objects M.

$$\begin{aligned}\left(\bigcup_{t\in T} M_t\right)' &= \{a \in A \mid (m,a) \in I \text{ for all } m \in \bigcup_{t\in T} M_t\} \\ &= \{a \in A \mid (m,a) \in I \text{ for all } m \in M_t \text{ for all } t \in T\} \\ &= \bigcap_{t\in T} M_t'\end{aligned}$$

By the symmetry due to the duality of the correspondence, the proof is almost identical for M a set of attributes.

□

Proof. (**The Basic Theorem on Concept Lattices**)

$$\left(\bigcap_{t\in T} M_t, \left(\bigcup_{t\in T} N_t\right)''\right) = \left(\bigcap_{t\in T} N_t', \left(\bigcup_{t\in T} N_t\right)''\right) = \left(\left(\bigcup_{t\in T} N_t\right)', \left(\bigcup_{t\in T} N_t\right)''\right)$$

hence $\left(\bigcap_{t\in T} M_t, \left(\bigcup_{t\in T} N_t\right)''\right)$ is a concept. Obviously $\bigcap_{t\in T} M_t$ is the largest set contained in M_t for all $t \in T$ and therefore the infimum of $\{(M_t, N_t) \mid t \in T\}$. The proof for the supremum is analogous to the above proof.

Table 2.2: Many-valued context representing the vital statistics collected from 5 automobiles.

	Make	Type	Fuel Economy (City)
Accord	Honda	Sedan	25
Beetle	VW	Coupe	28
Civic	Honda	Coupe	39
Explorer	Ford	SUV	15
Jetta	VW	Sedan	27
Wrangler	Jeep	SUV	14

□

2.2 Many-valued contexts

In the common vernacular, "attributes" are not restricted to describing the properties an object may or may not have as is the case with the formal definition. For example, attributes such as "color", "religion", "gender","weight", "ethnicity", and "income" can all take on many *values* and as such we call them *many-valued attributes* as opposed to the *single-valued attributes* previously considered.

Definition 2.2.1. A *many-valued context* (G, M, W, I, ϕ) consists of sets G, M, and W, a binary relationship I between G and M (*i.e.* $I \subset G \times M$), and a map $\phi : I \to W$. As in the definition of a single-valued context, the elements of G are called *objects* and those of M (*many-valued*) *attributes*. The elements of W are called *attribute values*.

As with single-valued contexts, many-valued contexts can be displayed in table form with the rows and columns labeled by the objects and many-valued attributes. However, the attribute value $\phi(g, m)$ is entered in the cell with row and column labeled g and m respectively (provided $(g, m) \in I$).

For a fixed attribute $m \in M$ we will often abuse notation and write $m(g)$ for $\phi(g, m)$. In other words, the attribute m can be thought of as the map ϕ restricted to m.

Example 2.2.1. Many attributes associated with cars are single valued (*e.g.*, A/C, convertible, etc.) though most are many valued by nature. Table 2.2 is a many-valued context describing the relationship between six cars and three attributes.

The set of attribute values for the context in Table 2.2 is

$$\mathbb{N} \cup \{\text{Honda, VW, Ford, Jeep}\} \cup \{\text{Sedan, Coupe, SUV}\}$$

2.2.1 Scaling

In order to assign concepts to a many-valued context, the context is transformed into a single-valued context and the concepts of the single-valued context are interpreted as the concepts of the many-valued context (alternatives to this approach have been suggested but will not be explored here, see [57]). If the attribute values are numerical, as with Fuel Economy in the above example, then the transformation is simply a discretization of the values. Unfortunately, as with any discretization approach, there is no unique way to transform a many-valued context onto a single-valued context.

Formally, a many-valued context is transformed by constructing a *scaling* for each attribute. The scales are used to construct single-valued contexts for each attribute which are then combined or joined to form a single-valued context which represents the original many-valued context.

Definition 2.2.2. A **scale** for an attribute $m \in M$ from a many-valued context (G, M, W, I, ϕ) is a (single-valued) context $S_m = (G_m, M_m, I_m)$ with $G_m = \{m(g) \mid (g, m) \in I\}$.

Any context can be used as a scale; however, there are standard scales that are more natural than others. In our example, the values for Type are a finite number of categories so it is natural to use a scale that simply uses each value of Type as an attribute. On the other hand, the values for Fuel Economy are numeric; a scale that discretizes the values is more appropriate. Three examples of scales used for discretization are as follows:

Example 2.2.2. *(***Scales)**

Ordinal scale $\mathbf{O}_n = (n, n, \leq)$.

$\mathbf{O}_5 =$

	1	2	3	4	5
1	x	x	x	x	x
2		x	x	x	x
3			x	x	x
4				x	x
5					x

Interordinal scale $\mathbf{I}_n = (n, n, \{\leq, \geq\})$

$\mathbf{I}_4 =$

	≤ 1	≤ 2	≤ 3	≤ 4	≥ 1	≥ 2	≥ 3	≥ 4
1	x	x	x	x	x			
2		x	x	x	x	x		
3			x	x	x	x	x	
4				x	x	x	x	x

Intraordinal scale $\mathbf{A}_{n,S_m} = (n, S_m, \leq * <)$ where S_m is a totally ordered set

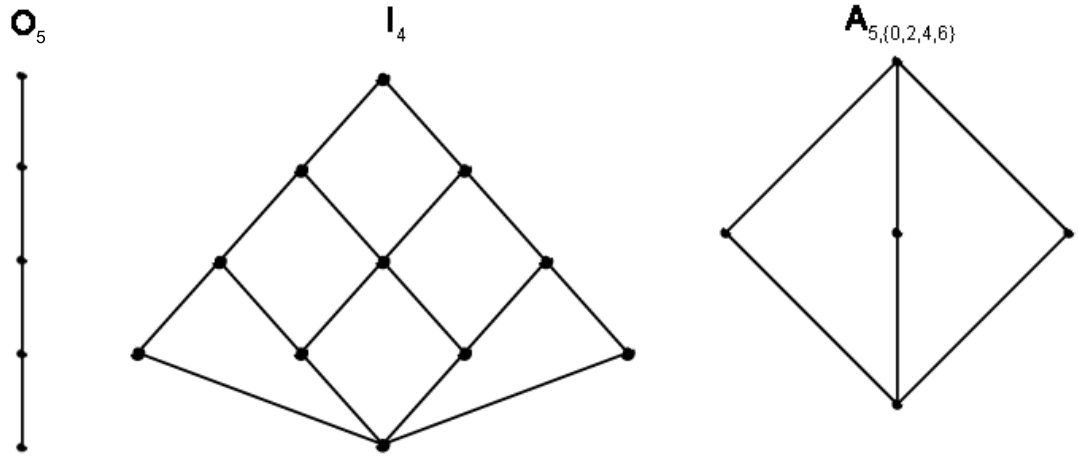
$\mathbf{A}_{5,\{0,2,4,6\}} =$

	$0 \leq * < 2$	$2 \leq * < 4$	$4 \leq * < 6$
1	x		
2		x	
3		x	
4			x
5			x

For an expanded list of useful scales see [50]

Figure 2.4 shows the lattices for the above three scales.

Figure 2.4: The concept lattices for the 3 scales described in Example 2.2.2



In Tables 2.3 and 2.4 are the cross tables for scales which could be used to transform the car context in our running example.

Table 2.3: Scale for transforming the attributes Make and Type

$S_M =$

	Ford	Honda	Jeep	VW
Ford	X			
Honda		X		
Jeep			X	
VW				X

$S_T =$

	Coupe	Sedan	SUV
Coupe	X		
Sedan		X	
SUV			X

2.2.2 Using scales to construct a single-valued context

Once a scale $S_m = (W, M_m, J_m)$ is chosen for an attribute m from a many-valued context (G, M, W, I, ϕ), a single-valued context $C_m = (G, M_m, I_m)$ is defined with

Table 2.4: Scale for transforming the attribute Fuel Economy

$S_{FE} =$

	≤ 15	≤ 25	≤ 35	≥ 15	≥ 25	≥ 35
15	X	X	X	X		
16		X	X	X		
⋮			...			
40				X	X	X
41				X	X	X

Table 2.5: Single-valued contexts for the attributes Make and Type

$C_M =$

	Ford	Honda	Jeep	VW
Accord		X		
Beetle				X
Civic		X		
Explorer	X			
Jetta				X
Wrangler			X	

$C_T =$

	Coupe	Sedan	SUV
Accord		X	
Beetle	X		
Civic	X		
Explorer			X
Jetta		X	
Wrangler			X

$$(g_i, m_j) \in I_m \text{ if and only if } (g_i, m) \in I \text{ and } (m(g_i), m_j) \in J_m$$

Tables 2.5 and 2.6 are the cross tables for the attributes from the car example using the scales in Tables 2.3 and 2.4.

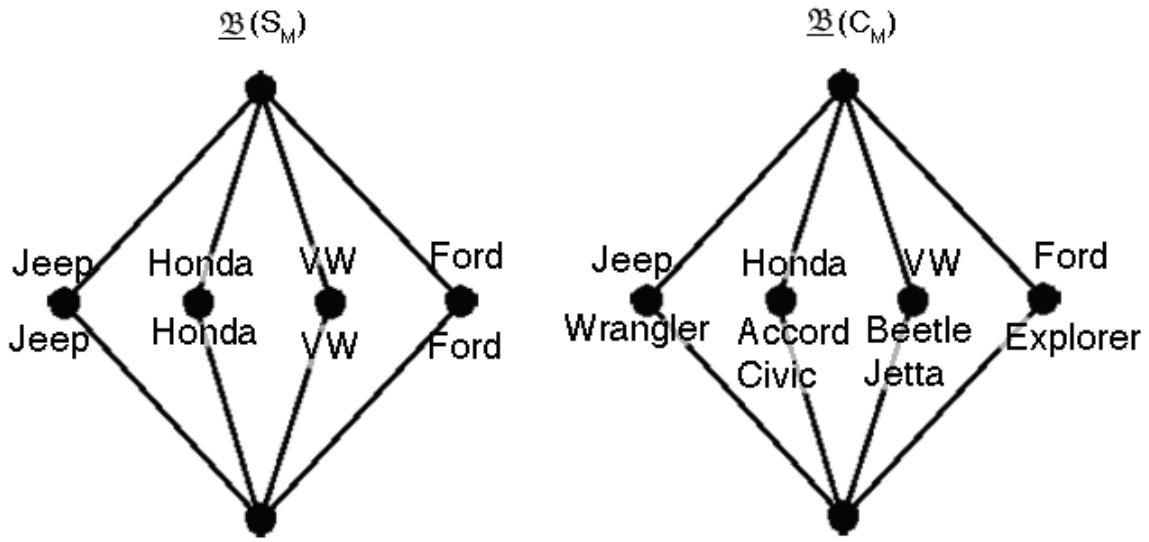
The scale used to transform a many-valued context can greatly determine the structure of the resulting lattice. In Figure 2.5 we see that the FCL of a scale and the context transformed by the scale can be isomorphic.

Table 2.6: Single-valued contexts for the attribute Fuel Economy

$C_{SF} =$

	≤ 15	≤ 25	≤ 35	≥ 15	≥ 25	≥ 35
Accord		X	X	X	X	
Beetle			X	X	X	
Civic				X	X	X
Explorer	X	X	X			
Jetta			X	X	X	
Wrangler	X	X	X			

Figure 2.5: Hasse diagram of $\underline{\mathfrak{B}}(S_M)$ and $\underline{\mathfrak{B}}(C_M)$



Lemma 2.2.1. Consider the many-valued context $K = (G, \{m\}, G \times \{m\}, W, \phi)$. Let $C_m = (G, M_m, I)$ be the single-valued context for K constructed using the scale $S_m = (E, M_m, J_m)$, with E the range of ϕ. If ϕ is defined on all $G \times \{m\}$ and $\{e\}^{J_m} \neq \emptyset$ for all $e \in E$, then, as lattices, $\underline{\mathfrak{B}}(S_m) \cong \underline{\mathfrak{B}}(C_m)$.

To prove two lattices are isomorphic, a bijective map between the two sets which preserves order must be demonstrated. In our situation, we need to construct a map between the concepts of the two lattices. Since the attribute sets of both contexts are identical, a map that preserves intents is the most promising candidate for an isomorphism between the two lattices.

Proof. Let $Y \subseteq M_m$. Then

$$\begin{aligned}
(Y^{J_m})^{J_m} &= \{n \in M_m \mid (\phi(g,m), n) \in J_m \text{ for all } \phi(g,m) \in Y^{J_m}\} \\
&= \{n \in M_m \mid (\phi(g,m), n) \in J_m \text{ for all } \phi(g,m) \in E \\
&\qquad \text{such that } (\phi(g,m), y) \in J_m \text{ for all } y \in Y\} \\
&= \{n \in M_m \mid (g,n) \in I \text{ for all } g \in G \\
&\qquad \text{such that } (g,y) \in I \text{ for all } y \in Y\} \\
&= \{n \in M_m \mid (g,n) \in I \text{ for all } g \in Y^I\} \\
&= (Y^I)^I
\end{aligned} \tag{2.1}$$

Define $\psi : \underline{\mathfrak{B}}(S_m) \to \underline{\mathfrak{B}}(C_m)$ via $\psi((X,Y)) = (Y^I, Y)$

By Equation 2.1, ψ is well defined. Now $\psi((X_1, Y_1)) = \psi((X_2, Y_2))$ if and only if $Y_1 = Y_2$. Also, for $(Y^I, Y) \in \underline{\mathfrak{B}}(C_m)$, $(Y^{I_{ev}}, Y) \in \underline{\mathfrak{B}}(S_{ev})$ by Equation 2.1 and therefore ψ is a bijection.

Finally,

$$\begin{aligned}(X_1, Y_1) \leq (X_2, Y_2) &\iff Y_2 \subseteq Y_1 \\ &\iff (Y_1^I, Y_1) \leq (Y_2^I, Y_2) \\ &\iff \psi(X_1, Y_1) \leq \psi(X_2, Y_2)\end{aligned}$$

□

2.2.3 Combining scaled contexts

When transforming a many-valued context (G, M, W, I, ϕ), there will be a single-valued context for each element of M; hence, if M has more than one element, then the multiple contexts need to be combined to form one unified context. *Apposition* is the operation employed to combine multiple contexts having the same set of objects in common.

Definition 2.2.3. Let $C_1 = (G, M_1, I_1)$ and $C_2 = (G, M_2, I_2)$ be contexts. Using the abbreviations $\dot{M}_j = \{j\} \times M_j$ and $\dot{I}_j = \{j\} \times I_j$ for $j \in \{1, 2\}$, then

$$C_1|C_2 := (G, \dot{M}_1 \cup \dot{M}_2, \dot{I}_1 \cup \dot{I}_2)$$

is the *apposition* of C_1 and C_2.

The corresponding *apposition lattice* will be denoted by either

$$\underline{\mathfrak{B}}(C_1|C_2) \qquad \text{or} \qquad \underline{\mathfrak{B}}(C_1) \circledast \underline{\mathfrak{B}}(C_2)$$

By using $\dot{M}_j$ we are requiring that the attribute sets of the two contexts being opposed are disjoint. This is simply a formality and will not be used if the attribute sets are naturally disjoint. Under the above definition, apposition is commutative but not associative. Since non-associativity is due solely to a notational technicality, we will identify the contexts

$$(C_1|C_2)|C_2 \quad \text{and} \quad C_1|(C_2|C_3).$$

The single-valued context and the lattice corresponding to the car context of our running example are in Table 2.7 and Figure 2.6.

Table 2.7: Cross table for the context $C_M|C_T|C_{FE}$ (A = Accord, B = Beetle, C = Civic, E = Explorer, J = Jetta, W = Wrangler, F = Ford, H = Honda, Jp = Jeep, Cp = Coupe, S = Sedan) with corresponding concept lattice.

	F	H	Jp	VW	Cp	S	SUV	≤ 15	≤ 25	≤ 35	≥ 15	≥ 25	≥ 35
A		X				X			X	X	X	X	
B				X	X					X	X	X	
C		X			X						X	X	X
E	X						X	X	X	X			
J				X		X				X	X	X	
W			X				X	X	X	X			

2.3 An algebraic description of apposition lattices

We will see in the next chapter that apposition plays a big part in the construction of the microBLAST representation of microarray data. Therefore, some time is spent now on the mathematical elucidation of the operation. By definition, the attributes of the apposition of two contexts $K_1|K_2$ is the disjoint union of sets of attributes of K_1 and K_2. The intent of any concept of $\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$ can also be described as the union of the, possibly empty, disjoint sets of attributes from the set of attributes of K_1 and K_2 respectively. Since the incidence relationship of $K_1|K_2$ is also defined to be disjoint on the set of attributes, the extent of the concepts of $\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$ can be described in terms of extents from $\underline{\mathfrak{B}}(K_1)$ and $\underline{\mathfrak{B}}(K_2)$. This is clarified in the following lemma.

Lemma 2.3.1. Let $K_i = (G, M_i, I_i)$ be contexts for $i \in \{1, 2\}$. Then

$$\mathfrak{E}(\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)) = \left\{ O_1 \bigcap O_2 \mid O_i \in \mathfrak{E}(\underline{\mathfrak{B}}(K_i)) \text{ for } i \in \{1, 2\} \right\}$$

Proof.

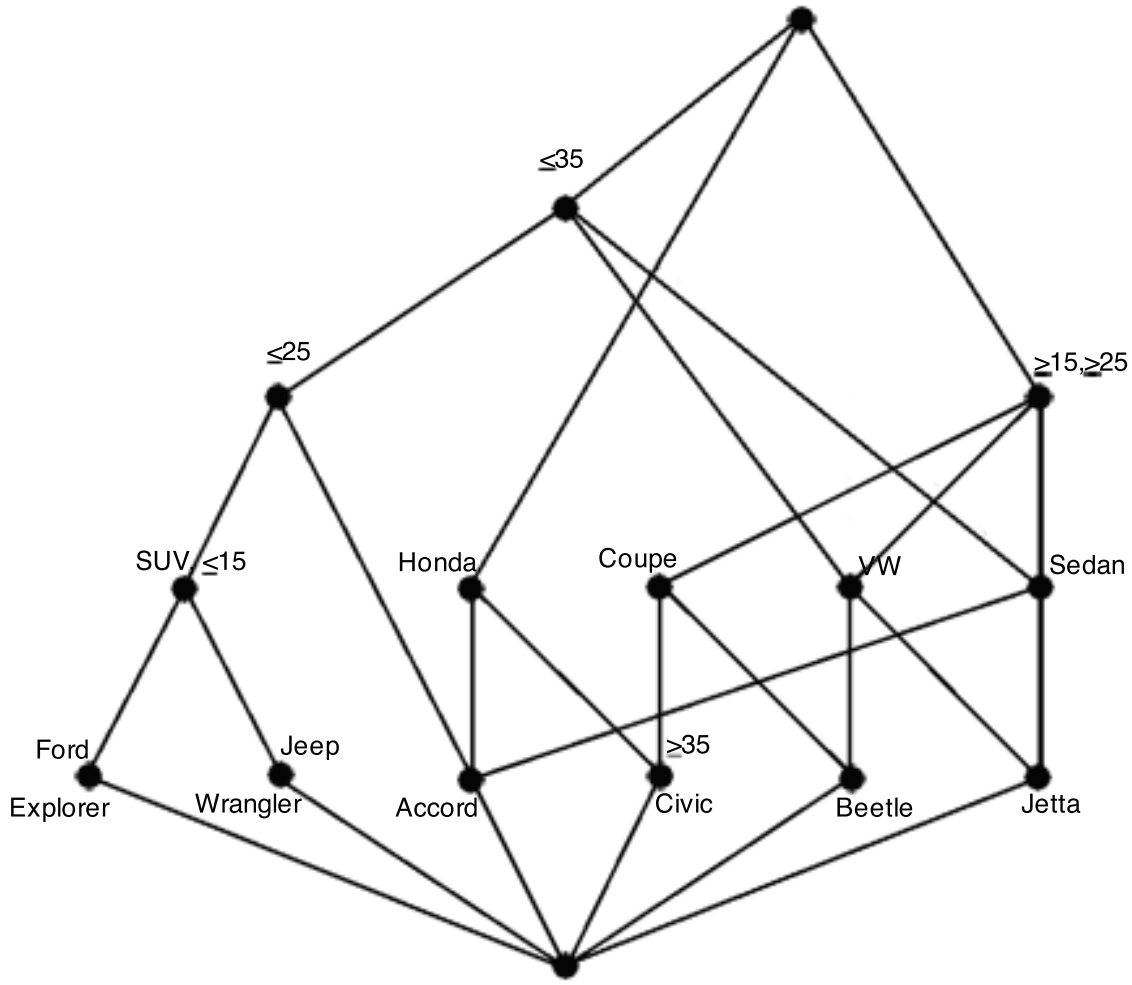
The $'$ notation will be used to denote the prime operator defined using the incidence relation from $K_1|K_2$ and I_i will be used to denote the prime operator restricted to the context K_i.

Let $O_i \in \mathfrak{E}(\underline{\mathfrak{B}}(K_i))$ for $i \in \{1, 2\}$. Then $(O_i, O_i^{I_i}) \in \underline{\mathfrak{B}}(K_i)$ for each i. By definition, $(\dot{O_i^{I_i}})' = O_i$, which implies that $(O_i, O_i') \in \underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$. Finally,

$$(O_1, O_1') \wedge (O_2, O_2') = (O_1 \cap O_2, (O_1 \cap O_2)')$$

and hence,

Figure 2.6: Hasse diagram for the lattice $\underline{\mathfrak{B}}(C_M) \circledast \underline{\mathfrak{B}}(C_T) \circledast \underline{\mathfrak{B}}(C_{FE})$



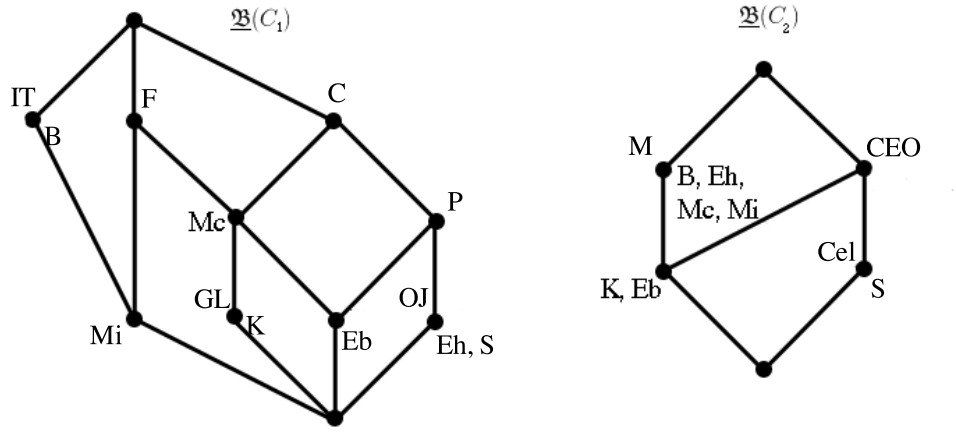
$$\left\{ O_1 \bigcap O_2 \mid O_i \in \mathfrak{E}(\underline{\mathfrak{B}}(K_i)) \text{ for } i \in \{1,2\} \right\} \subseteq \mathfrak{E}(\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)).$$

Let $(X, Y) \in \underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$. Then $Y = \dot{B}_1 \cup \dot{B}_2$ where $B_i \subseteq M_i$ for $i \in \{1,2\}$. Then $X = Y' = \dot{B}_1{}' \cap \dot{B}_2{}' = B_1^{I_1} \cap B_2^{I_2}$. Since $B_i^{I_i} \in \mathfrak{E}(\underline{\mathfrak{B}}(K_i))$ this completes the proof.

□

Since the extents of the concepts of $\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$ can be described as the intersection of two sets in the set of attributes from K_1 and K_2 respectively, it is tempting to assume that the union of intents of concepts from $\underline{\mathfrak{B}}(K_1)$ and $\underline{\mathfrak{B}}(K_2)$ would comprise an intent of a concept from $\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$. The following example demonstrates that such an assumption is invalid.

Figure 2.7: Hasse diagram of the two white-color criminal lattices.



Example 2.3.1. In order to gain some familiarity with lattice appositions, we will reconsider the infamous white collar criminals from Example 2.1.1 on page 23. Context C_1 is the context discussed in the earlier example and is presented again here in Table 2.8 for ease of analysis. The second context C_2 will also serve as a running example for the following section. The object set of C_2 consists of the same criminals as in C_1 however the attribute set is different. See Table 2.8 for the cross table for C_2. The Hasse diagrams are presented in Figure 2.7.

Table 2.8: Cross tables for the contexts C_1 (C = Conspiracy, F = Fraud, GL = Grand Larceny, IT = Insider Trading, OJ = Obstruction of Justice, and P = Perjury) and C_2 (Cel = Celebrity, M = Male, and CEO = Chief Executive Officer).

$C_1 =$

	C	F	GL	IT	OJ	P
Boesky				X		
Ebbers	X	X				X
Ehrlichman	X				X	X
Kozlowski	X	X	X			
McDougal	X	X				
Milken		X		X		
Stewart	X				X	X

$C_2 =$

	Cel	M	CEO
Boesky		X	
Ebbers		X	X
Ehrlichman		X	
Kozlowski		X	X
McDougal		X	
Milken		X	
Stewart	X		X

As a set in the apposition context $C_1|C_2$, the extent of the intent of $(\{C, OJ, P\} \cup \{CEO\})$ is

$$(\{C, OJ, P\} \cup \{CEO\})'' = \{S\}' = \{C, OJ, P, Cel, CEO\}.$$

Hence, $\{C, OJ, P\} \cup \{CEO\}$ is not an intent.

As the example suggests, the union of two intents is not necessarily large enough to be an intent in the apposition context. The following lemma describes the algebraic structure of the intents of an apposition context in terms of the original contexts.

Lemma 2.3.2. Let $K_i = (G, M_i, I_i)$ be contexts for $i \in \{1, 2\}$. For $X \in \underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$,

$$int(X) = \bigcup\{\dot{A}_1 \cup \dot{A}_2 \mid A_i \in \mathfrak{I}(\underline{\mathfrak{B}}(K_i)) \text{ and } ext(X) = A_1^{I_1} \cap A_2^{I_2}\}$$

Proof. By definition,

$$\begin{aligned} int(X) &= ext(X)' \\ &= \{m \in \dot{M}_1 \cup \dot{M}_2 \mid (o, m) \in \dot{I}_1 \cup \dot{I}_2 \text{ for all } o \in ext(X)\} \\ &= \bigcup_{i \in \{1,2\}} \{m \in \dot{M}_i \mid (o, m) \in \dot{I}_i \text{ for all } o \in ext(X)\} \\ &= \dot{X}_1 \bigcup \dot{X}_2 \qquad \text{where } X_i = ext(X)^{I_i} \text{ for } i \in \{1, 2\}. \end{aligned}$$

Now, $X_i^{I_i I_i} = X_i$ and $ext(X) = ext(X)'' = (\dot{X}_1 \cup \dot{X}_2)' = \dot{X}_1{}' \cap \dot{X}_2{}' = X_1^{I_1} \cap X_2^{I_2}$ which implies that

$$int(X) \subseteq \bigcup\{\dot{A}_1 \cup \dot{A}_2 \mid A_i \in \mathfrak{I}(\underline{\mathfrak{B}}(K_i)) \text{ and } ext(X) = A_1^{I_1} \cap A_2^{I_2}\}.$$

Let $\dot{a} \in \bigcup\{\dot{A}_1 \cup \dot{A}_2 \mid A_i \in \mathfrak{I}(\underline{\mathfrak{B}}(K_i))$ and $ext(X) = A_1^{I_1} \cap A_2^{I_2}\}$. Then there exists $A \in \mathfrak{I}(\underline{\mathfrak{B}}(K_i))$ for some $i \in \{1, 2\}$ such that $a \in A$ and $ext(X) \subseteq A^{I_i}$. Now $A = A^{I_i I_i} \subseteq ext(X)^{I_i} \subseteq ext(X)' = int(X)$ which implies that $a \in int(X)$. This demonstrates the reverse containment and completes the proof.

□

Example 2.3.2. (Revisiting Example 2.3.1)

As we have seen, $\{S\}'$ is an intent of $\underline{\mathfrak{B}}(C_1) \circledast \underline{\mathfrak{B}}(C_2)$. Since

$$\{S\} = \{S\} \cap \{Eh, S\} = \{S\} \cap \{Eb, Eh, S\} = \{S\} \cap \{Eb, Eh, Mc, S\} = \{Eb, K, S\} \cap \{Eh, S\}$$

(all intents of $\underline{\mathfrak{B}}(K_1)$ and $\underline{\mathfrak{B}}(K_1)$) then, according to Lemma 2.3.2,

$$\begin{aligned} \{S\}' =& \{Cel, CEO\} \cup \{C, OJ, P\} \cup \{Cel, CEO\} \cup \{C, P\} \cup \{Cel, CEO\} \cup \{C\} \\ & \cup \{CEO\} \cup \{C, OJ, P\} \\ =& \{C, Cel, CEO, OJ, P\} \end{aligned}$$

Lemmas 2.3.1 and 2.3.2 provide a means of describing the concepts of $\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$ completely in terms of intents of concepts in $\underline{\mathfrak{B}}(K_1)$ and $\underline{\mathfrak{B}}(K_1)$.

Theorem 2.3.1.

$(X, Y) \in \underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$ if and only if there exist $O_i \in \mathfrak{E}(\underline{\mathfrak{B}}(K_i))$ for $i \in \{1, 2\}$ such that

$$X = O_1 \cap O_2$$

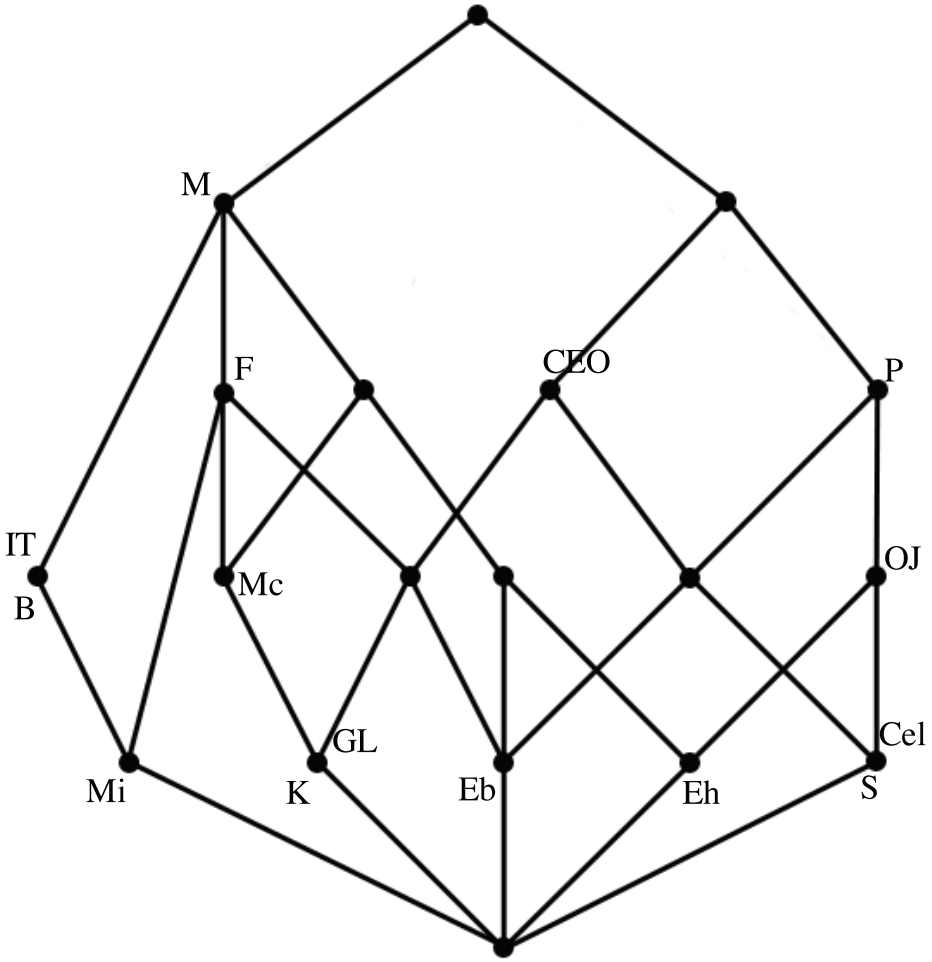
and

$$Y = \bigcup \{\dot{A}_1 \cup \dot{A}_2 \mid A_i \in \mathfrak{I}(\underline{\mathfrak{B}}(K_i)) \text{ and } O_1 \cap O_2 = A_1^{I_1} \cap A_2^{I_2}\}$$

Proof. This follows directly from Lemmas 2.3.1 and 2.3.2.

□

Figure 2.8: Hasse diagram of the FCL $\underline{\mathfrak{B}}(C_1) \circledast \underline{\mathfrak{B}}(C_2)$



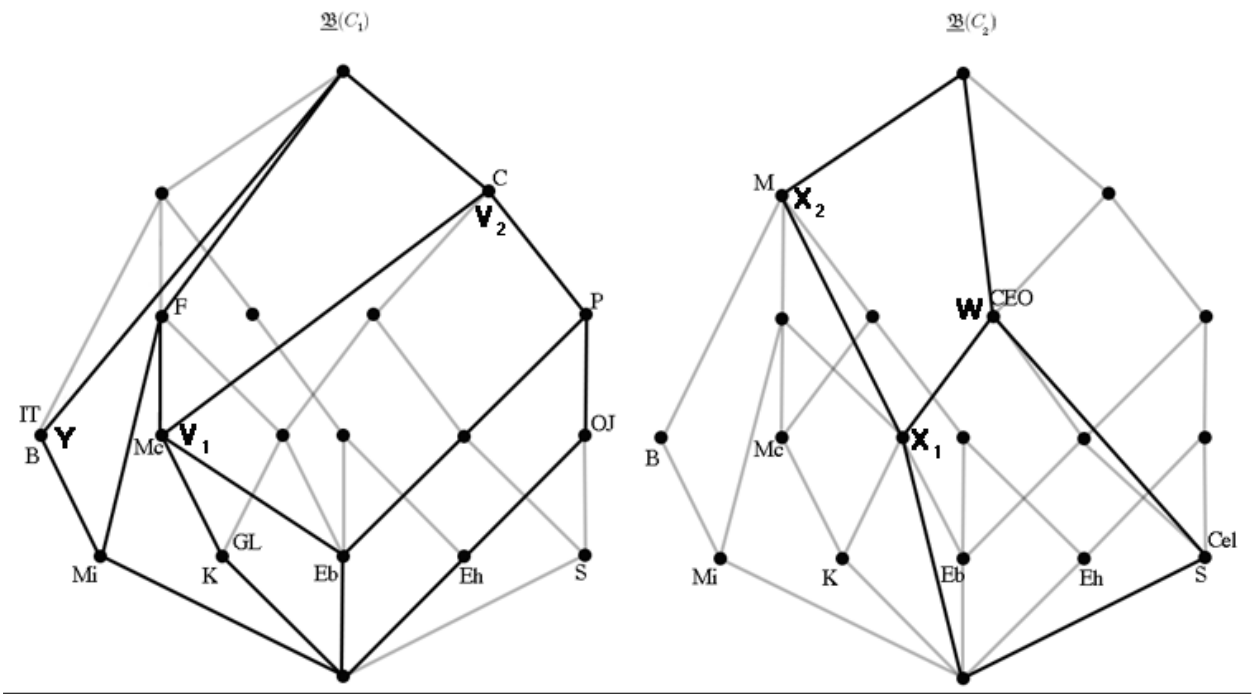
2.3.1 Algorithm for constructing the lattice $\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$

Theorem 2.3.1 suggests a method for constructing the concepts of an apposition lattice from the concepts of its generating lattices. Given two concept lattices $\underline{\mathfrak{B}}(K_1)$ and $\underline{\mathfrak{B}}(K_2)$, one simply needs to compute the intersection of the extents in $\underline{\mathfrak{B}}(K_1)$ and $\underline{\mathfrak{B}}(K_2)$ and the union of the appropriate intents. This process does not lend itself to constructing the actual lattice structure of the apposition lattice.

Given a concept $X \in \underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$ with extent $O_1 \cap O_2$, $O_i \in \mathfrak{E}(\underline{\mathfrak{B}}(K_i))$, it is reasonable to suspect that the neighbors of X are determined by the "intersection" of the neighbors of O_1 and O_2. Unfortunately it is not so straightforward; however, the neighborhood structure of $\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$ is determined by the neighborhood structures of both $\underline{\mathfrak{B}}(K_1)$ and $\underline{\mathfrak{B}}(K_2)$ as we will see in Theorem 2.3.2. In order to gain insight, we will continue with our running example.

Example 2.3.3.

Figure 2.9: Diagram of the lattices $\underline{\mathfrak{B}}(C_1)$ and $\underline{\mathfrak{B}}(C_2)$ embedded in $\underline{\mathfrak{B}}(C_1) \circledast \underline{\mathfrak{B}}(C_2)$



Consider the concepts

$$V_1 = (\{Eb, K, Mc\}, \{C, F\}) \text{ and } V_2 = (\{Eb, Eh, K, Mc, S\}, \{C\}) \text{ in } \underline{\mathfrak{B}}(C_1)$$

and the concept

$$W = (\{Eb, K, S\}, \{CEO\}) \in \underline{\mathfrak{B}}(C_2).$$

Then V_1 is a neighbor of V_2 in $\underline{\mathfrak{B}}(C_1)$ (see Figure 2.9).

Now, $\{Eb, K, Mc\} \cap \{Eb, K, S\} = \{Eb, K\}$ and $\{Eb, Eh, K, Mc, S\} \cap \{Eb, K, S\} = \{Eb, K, S\}$. Since $\{Eb, K\}$ and $\{Eb, K, S\}$ differ by only one element, namely S, their corresponding concepts in $\underline{\mathfrak{B}}(C_1) \circledast \underline{\mathfrak{B}}(C_2)$ must be neighbors.

Unfortunately, this is not always the case. Consider the concepts

$$X_1 = (\{Eb, K\}, \{CEO, M\}) \text{ and } X_2 = (\{B, Eb, Eh, K, Mc, Mi\}, \{M\}) \text{ in } \underline{\mathfrak{B}}(C_2)$$

and the concept

$$Y = (\{B, Mi\}, \{IT\}) \in \underline{\mathfrak{B}}(C_1).$$

Then X_1 is a neighbor of X_2 (see Figure 2.9).

Now $\{Eb, K\} \cap \{B, Mi\} = \emptyset$ and $\{B, Eb, Eh, K, Mc, Mi\} \cap \{B, Mi\} = \{B, Mi\}$. However, upon inspection of Figure 2.8, one can see that the concepts of $\underline{\mathfrak{B}}(C_1) \circledast \underline{\mathfrak{B}}(C_2)$ with extents $\emptyset$ and $\{B, Mi\}$ are not neighbors. However, one is a subconcept of the other, which will always be the case.

As the previous example demonstrates, given two concepts X and Y in $\underline{\mathfrak{B}}(K_1)$ and $\underline{\mathfrak{B}}(K_2)$ respectively, the intersection of $ext(X)$ with the extent of a neighboring concept of Y in $\underline{\mathfrak{B}}(K_2)$ is not guaranteed to be an extent of a neighbor of the concept with extent $ext(X) \cap ext(Y)$ in $\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$. However, as the following theorem demonstrates, the converse is true, that is, two neighboring concepts in $\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$ must have extents that are the intersections of extents of neighboring concepts in either $\underline{\mathfrak{B}}(K_1)$ or $\underline{\mathfrak{B}}(K_2)$.

Theorem 2.3.2. Let $K_i = (G, M_i, I_i)$ be contexts for $i \in \{1, 2\}$. For $\Lambda_1, \Lambda_2 \in \underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$, if $\Lambda_1 \prec \Lambda_2$ then either

1. There exist $U \in \underline{\mathfrak{B}}(K_1)$ and $V, W \in \underline{\mathfrak{B}}(K_2)$ such that:
   - $ext(\Lambda_1) = ext(U) \cap ext(V)$
   - $ext(\Lambda_2) = ext(U) \cap ext(W)$
   - $V \prec W$ in $\underline{\mathfrak{B}}(K_2)$

   or

2. There exist $U \in \underline{\mathfrak{B}}(K_2)$ and $V, W \in \underline{\mathfrak{B}}(K_1)$ such that
   - $ext(\Lambda_1) = ext(V) \cap ext(U)$
   - $ext(\Lambda_2) = ext(W) \cap ext(U)$
   - $V \prec W$ in $\underline{\mathfrak{B}}(K_1)$

Proof.

By Lemma 2.3.1, there exist $A, B \in \mathfrak{E}(\underline{\mathfrak{B}}(C_1))$ and $X, Y \in \mathfrak{E}(\underline{\mathfrak{B}}(C_2))$ such that $ext(\Lambda_1) = A \cap X$ and $ext(\Lambda) = B \cap Y$.

Since $\Lambda_1 \prec \Lambda_2$, $A \cap X \subset B \cap Y$ which implies that

$$A \cap X \subseteq B \cap (X \cap Y) \subseteq B \cap Y$$

and hence either $A \cap X = B \cap (X \cap Y)$ or $B \cap (X \cap Y) = B \cap Y$, otherwise Λ_1 could not be a neighbor of Λ_2.

Assume $A \cap X = B \cap (X \cap Y)$:

There exists a maximal chain, with respect to length,

$$X \cap Y = Y_1 \subseteq Y_2 \subseteq \ldots \subseteq Y_n = Y$$

such that $Y_i \in \mathfrak{E}(\underline{\mathfrak{B}}(K_2))$ for all i. Since $A \cap X = B \cap (X \cap Y) \subseteq B \cap Y_i \subseteq B \cap Y$ and $\Lambda_1 \prec \Lambda_2$, it follows that either $A \cap X = B \cap Y_i$ or $B \cap Y_i = B \cap Y$ for all i and hence there exist a j such that

$$B \cap Y_i = \begin{cases} A \cap X & i \leq j \\ B \cap Y & i > j \end{cases}.$$

By the maximality of the above chain, $(Y_j, Y_j^{I_2}) \prec (Y_{j+1}, Y_{j+1}^{I_2})$.

By setting $U = (B, B^{I_1})$, $V = (Y_j, Y_j^{I_2})$, and $W = (Y_{j+1}, Y_{j+1}^{I_2})$, the existence statements of Case 1 of the theorem have been demonstrated

Assume $B \cap (X \cap Y) = B \cap Y$:

As in the previous case, there exists a maximal chain

$$A \cap B = B_1 \subseteq B_2 \subseteq \ldots \subseteq B_n = B$$

such that $B_i \in \mathfrak{E}(\underline{\mathfrak{B}}(K_1))$ for all i. By the same argument as above, there exists a j such that

$$B_i \cap (X \cap Y) = \begin{cases} A \cap X & i \leq j \\ B \cap Y & i > j \end{cases}$$

with $(B_j, B_j^{I_1}) \prec (B_{j+1}, B_{j+1}^{I_1})$.

By setting $U = (X \cap Y, (X \cap Y)^{I_2})$, $V = (B_j, B_j^{I_1})$, and $W = (B_{j+1}, B_{j+1}^{I_1})$, the existence statements of Case 2 of the theorem have been demonstrated.

□

We are now in a position to describe the algorithm APPOSITION for constructing an apposition lattice $\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$ solely from information from its generating lattices $\underline{\mathfrak{B}}(K_1)$ and $\underline{\mathfrak{B}}(K_2)$. The extent of every concept $X \in \underline{\mathfrak{B}}(K_1)$ is intersected with the extent of every concept $Y \in \underline{\mathfrak{B}}(K_2)$ which, by Lemma 2.3.1, is an extent of $\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$. The intents of X and Y are added to the intent list for $ext(X) \cap ext(Y)$. According to Lemma 2.3.2, if all intersections are performed, such a process is guaranteed to construct the intent of $(ext(X) \cap ext(Y))$. Finally, the extents of all the concepts above X in $\underline{\mathfrak{B}}(K_1)$ and the extents of all the concepts above Y in $\underline{\mathfrak{B}}(K_2)$ are intersected with the extents of X and Y respectively and tested for being extents of neighbors of $ext(X) \cap ext(Y)$. See Figure 2.10 for the pseudo-code for the algorithm APPOSITION. In the code, M^* is the set of concepts directly above M in $\underline{\mathfrak{B}}(K_1) \circledast \underline{\mathfrak{B}}(K_2)$.

By Theorem 2.3.2, the extent of every upper neighbor of M is of the form $W \cap X$ or $V \cap Y$ for some representation $(X \cap Y, (X \cap Y)')$ of M where $Y \prec W$ and $X \prec V$. Then Temp (see Figure 2.10) are all possible candidates for upper neighbors of M. A candidate neighbor Z is added to the list if it is a subconcept of K, an element in M^* (K is subsequently removed from M^*), or if no concept in M^* is found to be a subconcept of Z. In this process, all neighbors of M will eventually be added to M^*, since all combinations of $W \cap X$ or $V \cap Y$ will eventually be considered where $M = (X \cap Y, (X \cap Y)')$, $Y \prec W$, and $X \prec V$. If a false neighbor K is added to M^* at some step, then there must be a concept N such that $M \prec N \leq K$, hence K will eventually be removed from M^* when the concept N is considered.

2.3.2 Complexity of APPOSITION

If $N = max\{|\underline{\mathfrak{B}}(K_1)|, |\underline{\mathfrak{B}}(K_2)|\}$, then the asymptotic complexity of APPOSITION is $O(N^2)$. The complexity of Lindig's algorithm **??**, considered to be the fastest algorithm for constructing FCLs, is given as $O(|L|\,|G|^2\,|M|)$ where $L = \underline{\mathfrak{B}}(G, M, I)$. Since both measures are based on different parameters, it is difficult to directly compare the complexity of the two algorithms. Empirical evidence published in the same article supports that the runtime of Lindig's algorithm grows quadratically with respect to $|L|$. We are guaranteed that $N < |L|$ (conceivably much less, see Figure 2.9) and so it is possible that the runtime of APPOSITION would be much faster than that of Lindig's algorithm.

One aspect of Lindig's algorithm that slowed down the runtime of our implementation of Lindig's algorithm has been the construction of intents and extents of the concepts in the lattice. When the contexts are very large (as is the case when working with tens of thousands of objects), constructing extents requires searching through a very large look-up space. In the APPOSITION algorithm intents and extents are constructed by intersecting and unioning sets already existing in memory. Since intents and extents must be calculated multiple

times for every step in Lindig's algorithm, each step of APPOSITION should run faster than an equivalent step in Lindig's algorithm. Unfortunately, APPOSITION has yet to be implemented so all discussions of improvement on runtime are completely theoretical.

2.4 The formal concept community

Formal Concept Analysis is a method for information or knowledge analysis, representation, and management. A branch of applied lattice theory that applies Galois connections within binary relationships in order to represent and analyze various different forms of information, FCA was invented by Rudolf Wille in the early 80s [128]. FCA was Originally developed by Wille, his students, and a small group of researchers; however, over the last 10 years, an international community has developed which applies the methods of FCA to myriad fields of research such as linguistics [99], software engineering [87, 54], psychology [109], web-browsing [31], machine learning [74], and information retrieval [100]. FCA, however, is still relatively unknown in the USA.

Growth in the FCA community is due in part to existing FCA open-source software. Two of the most all-encompassing software implementations of FCA are ConExp (sourceforge.net/-projects/conexp) and ToscanaJ (http://toscanaj.sourceforge.net/). Both programs are cross-platform compatible, easy to install and fairly easy to use. However, neither program is close to realizing the full potential of FCA [101]. Such shortcomings can be attributed to the complexity of the underlying lattice structures and of the visualizations. Conceivably, with proper funding such difficulties will be overcome.

Businesses built around the methods of FCA have also contributed to the field's growth. Navicon (www.navicon.de) in Germany, founded by students of Wille, was the first company to be built around FCA. Its original vision was to develop and apply FCA software to information management tasks that arise in the development of nautical/radio technical equipment. More recently, an Australian company has developed an email analysis tool based on FCA (www.mailsleuth.com).

The basic structure of FCA has been rediscovered over and over again supporting the hypothesis that the basic structures of FCA appear to be fundamental to information representation. Godin el al.'s work in 1989 [53] introduced Galois lattices (which are equivalent to the lattice of interest in FCA) as a means of information representation and retrieval. Their work was based on a discovery, independent of Wille, by Barbut and Monjardet in the early 70s [8]. Carpineto and Romano's work in information retrieval and data clustering [23, 24, 25] uses methods similar to both Godin and Wille's. These three groups have integrated over time to make up an international FCA community.

Figure 2.10: Pseudo-code for the algorithm APPOSITION

```
Input ← 𝔅(K_1) & 𝔅(K_2)
Obj ← {}
FOR all (X, X') ∈ 𝔅(K_1)
  FOR all (Y, Y') ∈ 𝔅(K_2)
    M ← X ∩ Y
    IF M ∉ Obj
     THEN
      Obj ← Obj∪{M}
      Att(M) ← X' ∪ Y'
      M* = {}
     ELSE
      Att(M) ← Att(M)∪X' ∪ Y'
  ————FINDING NEIGHBORS OF M————
    Temp ← {}
    FOR all (V, V') ∈ (X, X')*
      IF V ∩ Y ∉ M*
        THEN Temp ← Temp∪{V ∩ Y}
    FOR all (W, W') ∈ (Y, Y')*
      IF W ∩ X ∉ M* THEN
        Temp ← Temp∪{W ∩ X}
    FOR all K ∈ Temp
      Q ← 0
      Hold ← M*
      WHILE Q = 0 AND Hold ≠ {}
        CHOOSE Z ∈Hold
        Hold ← Hold\Z
        IF K ∩ Z = K THEN
          M* ← (M*\Z) ∪ K
          Q ← 1
        IF K ∩ Z = Z THEN
          Q ← 1
      IF Q = 0 THEN
        M* ← M* ∪ K
```

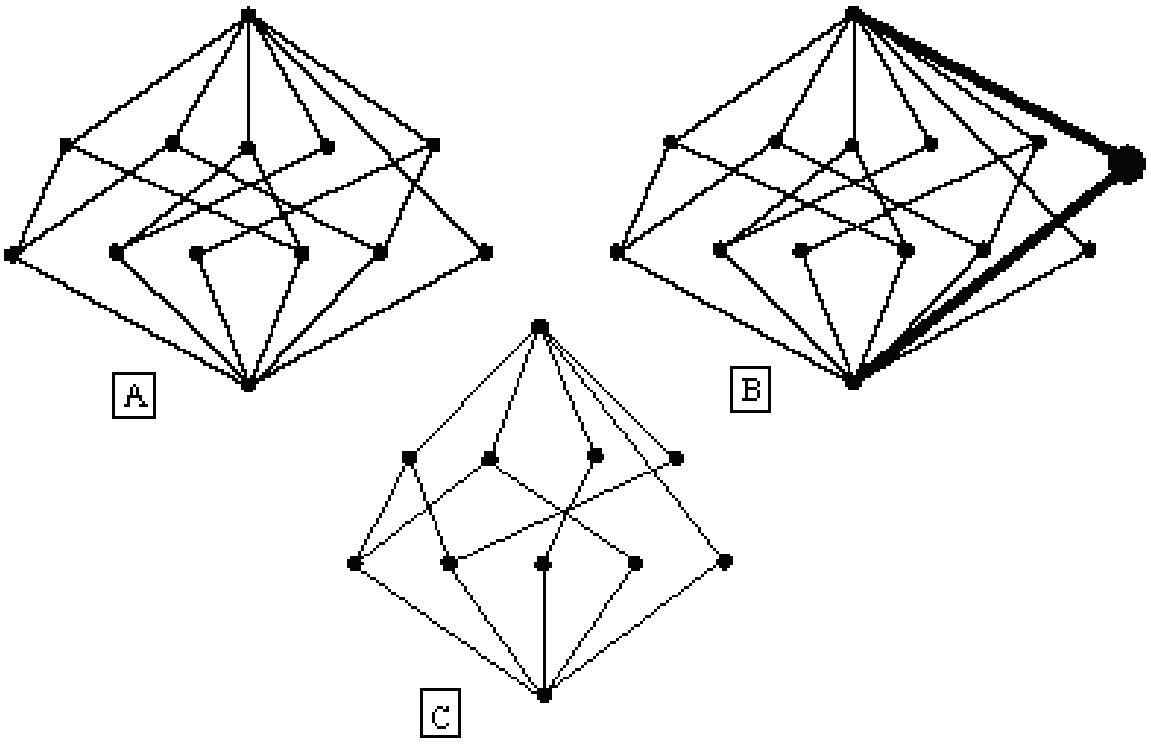
Chapter 3

microBLAST

We present a novel method, which we call microBLAST, that integrates both gene expression values as well as gene functionality in order to make comparisons of biological samples. The integrated product is represented as a lattice (visually represented as a Hasse diagram) in which elements correspond to groups of genes associated according to relationships derived from the integrated data. Graph measures are then employed to compare the Hasse diagram of a reference sample with other samples from the same set of experiments, from different labs, or from a gene expression database. For a more natural flow of discussion, lattices and their visual representations will be used interchangeably.

Figure 3.1 illustrates the general idea of microBLAST. The small-scale (only 10 genes are used) microBLAST representations of microarray experiments are presented. Protein motifs and gene expression values were used to generate the lattices in Figure 3.1. Lattice A and B represent samples collected from two individuals diagnosed with systemic lupus erythematosus (SLE) and Lattice C represents a sample collected from a control subject [7]. The central idea of microBLAST is that biologically "similar" samples will have "similar" lattices. Note that the lattices for the lupus samples differ by only two edges and a node (these are in bold on Lattice B in Figure 3.1) while the control lattice differs significantly from the other two.

Figure 3.1: microBLAST representation of gene expression profiles of ten genes from two lupus samples (Lattices A and B) and a control sample (Lattice C).



In order for the lattices in the above example to be visually accessible with little background experience, the expression levels of only 10 genes were used for this example. An increase in the number of genes used to build a microBLAST lattice of a sample directly affects the complexity of the representative lattice. For complex lattices, visual comparisons require greater sophistication from both the user as well as the visual representation. Graph measures can be used to supplement as well as complement the visual comparison of the lattice representations of microarray data. Suitable measures would allow the comparison of a reference sample to a large database of microarray samples in a reasonable amount of time, provided the database provides microBLAST representations of their samples.

The lattice representation of microarray data captures the gene expression value as well as user-defined biological information about the genes. In principle, this approach allows for increased investigative and comparative power as compared to representations which only take into account expression values. Also, the microBLAST approach groups genes into non-exclusive collections, allowing for genes to be present in multiple categories, potentially allowing for a more realistic representation of the relationships between gene and function than is possible with classical clustering methods.

3.1 Comparative measures

A key benefit to the representation of gene expression data as a concept lattice is that it allows the employment of existing well-established mathematical tools for the analysis of the lattice structure as well as for comparisons. Since concept lattices can be realized as acyclic labeled graphs, a natural approach to comparing microBLAST lattices is via metrics defined on their graphs.

3.1.1 Graph measures

Edit distance

Edit distance is a graph measure [86], which computes the number of edges and vertices that must be added or deleted to or from one graph to be transformed into another. Edit distance is the graph analogue of the Hamming distance between two strings, *i.e.*, the number of positions that differ between two strings of characters. For graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the edit distance between them is given by

$$ED(G_1, G_2) = |V_1| + |V_2| - 2|V_1 \cap V_2| + |E_1| + |E_2| - 2|E_1 \cap E_2|.$$

The edit distance is therefore fairly straightforward to compute.

One could also employ a weighted edit distance for graph comparisons in which the "expense" for adding or removing an edge or vertex is weighted by some scalar. If α_i was a weighting

function defined on the edges and vertices of G_i, then the weighted edit distance between G_1 and G_2 would be

$$ED(G_1, G_2; \alpha) = \sum_{v \in V_1} \alpha_1(v) + \sum_{v \in V_2} \alpha_2(v) - \sum_{v \in V_1 \cap V_2} (\alpha_1(v) + \alpha_2(v)) + \sum_{e \in E_1} \alpha_1(e) + \sum_{e \in E_2} \alpha_2(e) - \sum_{e \in E_1 \cap E_2} (\alpha_1(e) + \alpha_2(e))$$

Both of the above defined measures count the number of additions or deletions necessary two make the graphs equal. Edit distance can be modified further to count how many moves it would require to make two graphs "close" to being equal. For example, since the vertices of our graphs are labeled by genes, we can determine two vertices to be "close enough" if their intersection is greater than some threshold. Defining

$$V_i^k = \{v \in V_i \mid \exists\ w \in V_j \text{ such that } |v \cap w| \geq k\}$$

and

$$E_i^k = \{(e_h, e_t) \in E_i \mid \exists\ (f_h, f_t) \in E_j \text{ such that } |e_h \cap f_h| + |e_t \cap f_t| \geq 2k\},$$

then the edit distance between G_1 and G_2 with respect to k is given by

$$ED_k(G_1, G_2) = |V_1| + |V_2| - |V_1^k| - |V_2^k| + |E_1| + |E_2| - |E_1^k| - |E_2^k|$$

Eigenvalue metrics

For any graph $G = (V, E)$, an adjacency matrix A can be associated with rows and columns labeled by the vertices of G and

$$A(i,j) = \begin{cases} 1 \text{ if } (i,j) \in E \\ 0 \text{ else} \end{cases} .$$

The diagonal matrix D is also often associated with G in which $D(i,i) = \delta(i)$ the degree of the vertex i. The spectrum of A, $D - A$ and $D^{-1}A$ can be used to compute many graph invariants of G such as diameter and number of spanning trees (for details see [30, 37]). Since the eigenvalues of these three matrices encode invariants of the graph, a comparison of eigenvalues for two different graphs can be interpreted as a comparison of the graphs directly. One simple measure of similarity that uses eigenvalues is

$$\lambda(M(G_1), M(G_2)) := \frac{1}{min\{m_1, m_2\}} \sum_{i=1}^{min\{m_1,m_2\}} (\lambda_{1,i} - \lambda_{2,i})$$

where $\{\lambda_{j,i}, \ldots, \lambda_{j,m_j}\}$ are the eigenvalues of $M(G_j)$ indexed in ascending order.

Another means of using eigenvalues for defining measures on graphs is to assign a vector $I(G)$ to a graph G with entries numeric invariants of G (as calculated using the eigenvalues associated with G). The invariance measure between two graphs is defined as

$$Inv(G_1, G_2) := ((I(G_1) - I(G_2)) \cdot (I(G_1) - I(G_2)))^{1/2}$$

the Euclidean distance between the invariant vectors.

In either measure, the focus is on the "geometry" of the lattice and not on the concepts that make up the elements of the lattice.

3.1.2 Statistical metrics

Given two concept lattices $K_1 = \underline{\mathfrak{B}}(G, M, I_1)$ and $K_2 = \underline{\mathfrak{B}}(G, M, I_2)$ and a probability distribution P defined on $\mathfrak{P}(G)$, we can define a probabilistic similarity measure between concepts from the two lattices. For a concept $(X_i, Y_i) \in K_i$ define $P(X, Y) = P(X)$ and for $A \in K_1$ and $B \in K_2$ define

$$Sim(A, B) = \frac{\log P((int(A) \cap int(B))^{I_1}) + \log P((int(A) \cap int(B))^{I_2})}{\log P(A) + \log P(B)}.$$

This measure of similarity is based on the entropy of a class as defined by Bernstein et al. [12]. Since $(int(A) \cap int(B))^{I_i}$ is the smallest extent with $ext(A) \cap ext(B)$ as a subset, Sim specifies similarity as the probabilistic degree of overlap of ascending intents. The average similarity of K_1 and K_2 is defined as

$$Sim(K_1, K_2) := \frac{1}{|K_1||K_2|} \sum_{A \in K_1} \sum_{B \in K_2} Sim(A, B)$$

Similarly we can define the average maximum similarity to be

$$mSim(K_1, K_2) := \frac{1}{2} \left(\sum_{A \in K_1} \underset{B \in K_2}{max} \{Sim(A, B)\} + \sum_{B \in K_2} \underset{A \in K_1}{max} \{Sim(A, B)\} \right)$$

The probability of an intersection of size c between two sets of size q and t both contained in a set of size g is given by

$$pV(c, t, q, g) = \sum_{k=c}^{min\{q,t\}} \frac{\binom{t}{k}\binom{g-t}{q-k}}{\binom{g}{q}} \qquad \text{where } \binom{n}{m} = \frac{n!}{m!(n-m)!}$$

The probability distribution of pV values for formal concepts can be estimated using Monte Carlo simulations as in [9]. Such a probability would be a good candidate for the similarity measure defined above.

The above three measures detect different aspects of similarity between two graphs such as: equality (edit distance), geometric similarity (Eigenvalue measures), and probabilistic measures. In our first implementation of the microBLAST method, we have employed edit distance solely as a measure of similarity. This decision was based on the fact that edit distance is easy to implement, it results are intuitive to analyze, and it scales linearly. For the remainder of this work, if a measure need to be specified, edit distance will consistently be used.

3.2 A mathematical description of microBLAST

A natural application of FCA to the modelling of microarray data would be to use the expression values of the genes as the attribute and the genes as the objects. Such a context is many-valued (see Definition 2.2.1) because the relationship between the object and attribute set is continuous and not binary. We will consider the following small gene expression data set.

Example 3.2.1. (**Small gene expression data set**)

$G = \{g_1, \ldots, g_8\}$ are genes with the following measured expression values

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
ev	3.8	15.6	8.7	2.1	3.3	7.8	14.2	2.8

The many-valued context for this data set is

$$C = (G, \{ev\}, E_1, G \times \{ev\}, \phi)$$

where the values of $\phi(g_i, ev)$ are the entries in the above table and $E_1 =$ is the range of ϕ.

3.2.1 Expression lattices

If only the expression values from one microarray experiment are used as the attributes for the genes, there is no need to appose contexts when transforming the gene expression many-valued context into a single-valued context.

The scale to be used to transform the context is left to the discretion of the user as there is no unique scale available. At the very least, the scale decided upon should somehow reflect the

nature of the attributes. Since the intraordinal scale (see 2.2.2 page 30) clusters the objects with respect to given thresholds, it is a natural scale to use for transforming the context in Example 3.2.1. We will use the following scale in Table 3.1 to transform our example data set.

Table 3.1: Scale used to transform the multi-valued context in Example 3.2.1

$\mathbf{A}_{D,T} =$

	≥ 0 and ≤ 5	≥ 5 and ≤ 10	≥ 10 and ≤ 20
2.1	x		
2.8	x		
3.3	x		
3.8	x		
7.8		x	
8.7		x	
14.2			x
15.6			x

where $D = \{2.1, 2.8, 3.3, 3.8, 7.8, 8.7, 14.2, 15.6\}$ and $T = \{0, 5, 10, 20\}$.

The single-valued context C_{E_1} for the attribute ev (and hence the entire many-valued context) constructed using the scale $A_{D,T}$ has incidence relation as in Figure 3.2 (where P_1 = "≥ 0 and ≤ 5", P_2 = "≥ 5 and ≤ 10", and P_3 = "≥ 10 and ≤ 20").

The above example is the standard process of constructing a FCL from gene expression data. To facilitate further discussion, the process is formally defined.

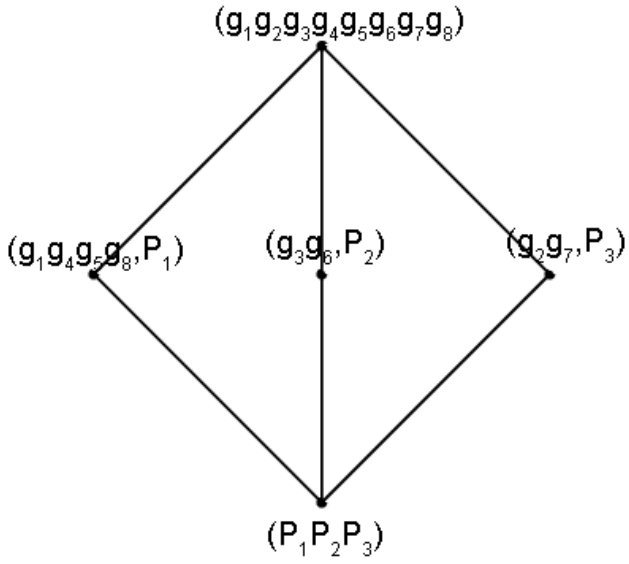
Definition 3.2.1. Given a set (G, E) of genes and their expression values and a scale S_{ev}, $\underline{\mathfrak{B}}((G, E); S_{ev})$ is the *expression lattice* for (G, E) using the context constructed from the multi-valued context $(G, \{ev\}, G \times \{ev\}, \mathbb{R}, \phi)$ using the scale S_{ev}.

It should be stressed that this formalism is introduced to allow mathematical analysis of the objects. In layman's terms, an expression lattice is simply a FCL constructed from a context with genes as objects and the attributes the categories to which the gene expression values are discretized.

As lattices, both the FCL for the scale $\mathbf{A}_{5,\{0,2,4,6\}}$ and $\underline{\mathfrak{B}}((G, E_1); \mathbf{A}_{5,\{0,2,4,6\}})$ are isomorphic (see figures 2.4 and 3.2). This is no coincidence since the entries in the cross table for both contexts are identical and there is a bijective correspondence between G and D. Note that, given two gene expression data sets, the first hypothesis of Lemma 2.2.1 is satisfied since $\phi(g, ev)$ is the expression value of g. Therefore, if the contexts for the two data sets are constructed using the same (or isomorphic) scales, then the lattices for the contexts will be isomorphic. In other words, the isomorphism between the expression lattices and the scale used to construct it can be used to show that the expression lattices of two gene expression data sets are isomorphic if they are constructed using isomorphic scales.

Figure 3.2: The expression context C_{E_1} and its corresponding FCL $\underline{\mathfrak{B}}(C_{E_1})$.

C_e	P_1	P_2	P_3
g_1	x		
g_2			x
g_3		x	
g_4	x		
g_5	x		
g_6		x	
g_7			x
g_8	x		



Proposition 3.2.1. Let (G, E_1) and (G, E_2) be gene expression data sets, and let S_1 and S_2 be scales in which each object is incident with at least one attribute and $\underline{\mathfrak{B}}(S_1) \cong \underline{\mathfrak{B}}(S_2)$. Then $\underline{\mathfrak{B}}((G, E_1); S_1) \cong \underline{\mathfrak{B}}((G, E_2) : S_2)$.

Proof. This is a restatement of Lemma 2.2.1 in the vocabulary of expression contexts. □

The edit distance between two lattices is zero only if the lattices are identical. Therefore, isomorphic lattices do not necessarily have edit distance zero between them. For example, let

$(G, E_2) =$

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
ev	3.8	1.6	8.7	12.1	3.3	7.8	14.2	7.8

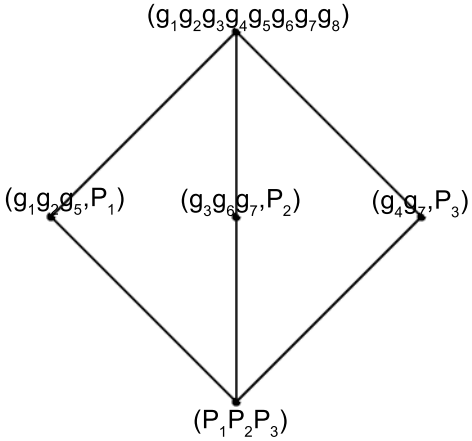
be another gene expression data set. In Figure 3.3 the expression context for (G, E_2) using the scale $\mathbf{A}_{E_2,T}$ as before, and the expression lattice $\underline{\mathfrak{B}}((G, E_2), \mathbf{A}_{E_2,T})$ is presented.

The edit distance between $\underline{\mathfrak{B}}((G, E_1), \mathbf{A}_{E_1,T})$ and $\underline{\mathfrak{B}}((G, E_2), \mathbf{A}_{E_2,T})$ is $5+5-2(2)+6+6-2(0) = 18$ which is $6(|T|-1)$. This is because the maximal and minimal elements of the lattice are the only identical concepts.

Lemma 3.2.1. Let (G, E_1) and (G, E_2) be two data sets of gene expression values and $\mathbf{A}_{E_i,T_i}$ the intraordinal scale used to construct $\underline{\mathfrak{B}}((G, E_i), \mathbf{A}_{E_i,T})$. If for all $t_1, t_2 \in T$ $t_1 \neq t_2$, we

Figure 3.3: The context for the data set (G, E_2) using the scale $\mathbf{A}_{E_2,T}$ as well as its corresponding FCL $\underline{\mathfrak{B}}((G, E_2), \mathbf{A}_{E_2,T})$.

C_e	P_1	P_2	P_3
g_1	x		
g_2	x		
g_3		x	
g_4			x
g_5	x		
g_6		x	
g_7			x
g_8		x	



have that $\{t_1\}' \cap \{t_2\}' = \emptyset$, $\underline{\mathfrak{B}}(\mathbf{A}_{E_1,T}) \cong \underline{\mathfrak{B}}(\mathbf{A}_{E_2,T})$ and $N^{I_1} \neq N^{I_2}$ for all $N \subset M_i$, then the edit distance between $\underline{\mathfrak{B}}((G, E_1), \mathbf{A}_{E_1,T})$ and $\underline{\mathfrak{B}}((G, E_2), \mathbf{A}_{E_2,T})$ is $6(|T| - 1)$.

Proof. Let C_i, for $i \in \{1, 2\}$ be the single-valued context for the data set (G, E_i) constructed using the scale $\mathbf{A}_{E_i,T}$.

By Corollary 3.2.1 and the proof of Lemma 2.2.1, there is an isomorphism $\psi : \underline{\mathfrak{B}}(C_1) \to \underline{\mathfrak{B}}(C_2)$ such that $\psi((X, Y)) = (Y^{I_2}, Y)$. By the hypothesis, $X = Y^{I_1} \neq Y^{I_2}$ provided $Y \notin \{\emptyset, M\}$. Hence, the edit distance between $\underline{\mathfrak{B}}(C_1)$ and $\underline{\mathfrak{B}}(C_2)$ is

$$2|\underline{\mathfrak{B}}(C_1)| - 4 + 2|\mathcal{E}(\underline{\mathfrak{B}}(C_1))|$$

Where $\mathcal{E}(\underline{\mathfrak{B}}(C_1))$ is the set of edges in the Hasse diagram of $\underline{\mathfrak{B}}(C_1)$.

Since $\{t_1\}' \cap \{t_2\}' = \emptyset$, by the proof of Lemma 2.2.1

$$\underline{\mathfrak{B}}(C_1) = \{(\{m\}^{I_1}, m) \mid m \in M_1\} \cup \{(G, \emptyset), (\emptyset, M)\}.$$

Therefore

$$|\underline{\mathfrak{B}}(C_1)| = |M_1| + 2 = (|T| - 1) + 2 = |T| + 1,$$
$$\mathcal{E}(\underline{\mathfrak{B}}(C_1)) = \{((G, \emptyset), K) \text{ or } ((\emptyset, M), K) \mid K \notin \{(G, \emptyset), (\emptyset, M)\}\},$$

and

$$|\mathcal{E}(\underline{\mathfrak{B}}(C_1))| = 2(|\underline{\mathfrak{B}}(C_1)| - 2) = 2(|T| - 1).$$

□

In the present application, the hypothesis $\{t_1\}' \cap \{t_2\}' = \emptyset$ simply states that an expression value can not be discretized into two states. $N_1^{I_1} \neq N_2^{I_2}$ is equivalent to saying that no two clusterings of the genes, with respect to their expression values in the two samples, are identical. This is a reasonable assumption when working with gene expression data since the data is fairly noisy and the likelihood that there will occur identical clusters for two gene expression data sets is very small. The exact probability is dependent upon the scale used to discretize the data as well as the distribution of gene expression values.

As is apparent, edit distance is not sensitive enough to be used for comparing expression lattices since the edit distance between any two such lattices has a high probability of being constant. It is then either necessary to use other metrics to compare expression lattices or integrate more information into the lattice representation of expression samples with the hope that edit distance will improve in performance.

There are other standard lattice measures that could be used to analyze expression lattices, as discussed earlier. However, we have decided to focus our present research on the integration of biological information into the lattice representation. We have had empirical success with this approach using edit distance as the metric (as will be further discussed in Chapter 4). We now focus on the method of incorporating different information into a concept lattice.

3.2.2 Biological lattices

We consider the inclusion of biological information into the construction of an FCL representation of a gene expression data set. Formally, if $\underline{\mathfrak{B}}((G, E), S_{ev})$ is an expression lattice, C_E the corresponding single-valued gene expression context, and C_B a context encoding biological information, then we will consider $\underline{\mathfrak{B}}(C_E|C_B) = \underline{\mathfrak{B}}(C_e) \circledast \underline{\mathfrak{B}}(C_B)$ as a microBLAST representation of (G, E) where $C_E|C_B$ is the apposition of C_E and C_B as in Definition 2.2.3 on page 34.

In general, any context in which the object set is a collection of genes will be considered a *biological context*. With this definition, an expression context is a biological context. In practice the attributes of a biological context will capture static characteristics of the genes such as function, location, or structural classification. Hence, expression contexts is a special case in which the attributes are not static (*i.e.*, they will change from experiment to experiment).

Example 3.2.2.

Let $G = \{g_1, \ldots, g_8\}$ be the same genes as in Example 3.2.1. Table 3.2 lists known molecular functions of the given genes.

As with the expression data, $GO(G)$ is a many-valued context $(G, F, W, G \times F, \phi)$ with $F = \{\text{Function}\}$, $W = \{\text{ATPase activity, ATP synthesis, ATP transport}\}$, and $\phi : I \to \mathfrak{P}(W)$ the power set of W. Since each element of W is a category, the ordinal scale $\mathbf{O}_W$ (see 2.2.2

Table 3.2: GO molecular function

$GO(G) =$

	Function
g_1	ATPase activity
g_2	ATP synthesis, ATP transport
g_3	ATP transport
g_4	ATPase activity, ATP synthesis
g_5	ATP transport
g_6	ATPase activity
g_7	ATPase activity, ATP synthesis
g_8	ATP synthesis, ATP transport

on page 30) is a natural scale to use to transform $GO(G)$ into a single-valued context. The transformed context, $C_B = (G, B, I_B)$ is in Figure 3.4 as well as the Hasse diagram of the FCL $\underline{\mathfrak{B}}(C_B)$ (note that since the range of ϕ is $\mathfrak{P}(W)$, Lemma 2.2.1 does not hold).

In general, the biological context will be many-valued. In all our applications of microBLAST, the attributes of the biological contexts have all been categorical in nature (as in the above example) and we have used the ordinal scale to transform them into single-valued contexts. Biological information such as chromosome location or relative position on the chromosome does not lend itself as naturally to scaling via the ordinal scale and some other scale would have to be devised to convert it into a single-valued context. The transformation of multi-valued biological contexts are not the focus of this paper; hence, when discussing biological contexts, it will be assumed, unless explicitly stated, that the context is single-valued.

Again, the use of scales is a formality that allows us to take advantage of theorems and techniques from Formal Concept Analysis. Intuitively however, a biological context is a collections of genes as well as a list of biological characteristics that could describe the genes or their product. The biological lattice then captures the interrelationships determined by the biological attributes of the genes.

3.2.3 microBLAST lattices

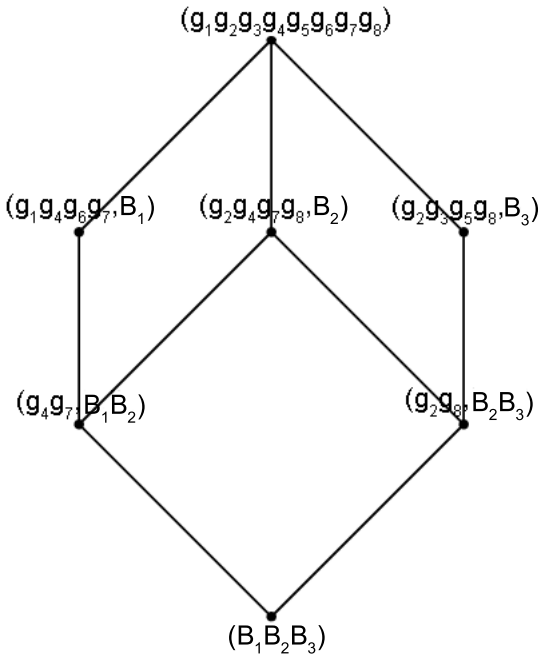
Definition 3.2.2. (**microBLAST lattice**)

Given a gene expression data set (G, E), a scale S_{ev}, and a biological context $C_B = (G, M, I)$, the *microBLAST* representation $\mu((G, E), S_{ev}, C_B)$ of (G, E) is the formal concept lattice $\underline{\mathfrak{B}}((G, E); S_v) \circledast \underline{\mathfrak{B}}(C_B)$.

Example 3.2.3. Let $\underline{\mathfrak{B}}((G, E); \mathbf{A}_{D,T})$ be the expression lattice constructed from Example 3.2.1 where $D = \{2.1, 2.8, 3.3, 3.8, 7.8, 8.7, 14.2, 15.6\}$ and $T = \{0, 5, 10, 20\}$, and C_B be the

Figure 3.4: Single-valued context C_B for $GO(O)$ as well as the Hasse diagram for the lattice $\underline{\mathfrak{B}}(C_B)$. B_1 = ATPase activity, B_2 = ATP synthesis, and B_3 = ATP transport.

	B_1	B_2	B_3
g_1	x		
g_2		x	x
g_3			x
g_4	x	x	
g_5			x
g_6	x		
g_7	x	x	
g_8		x	x

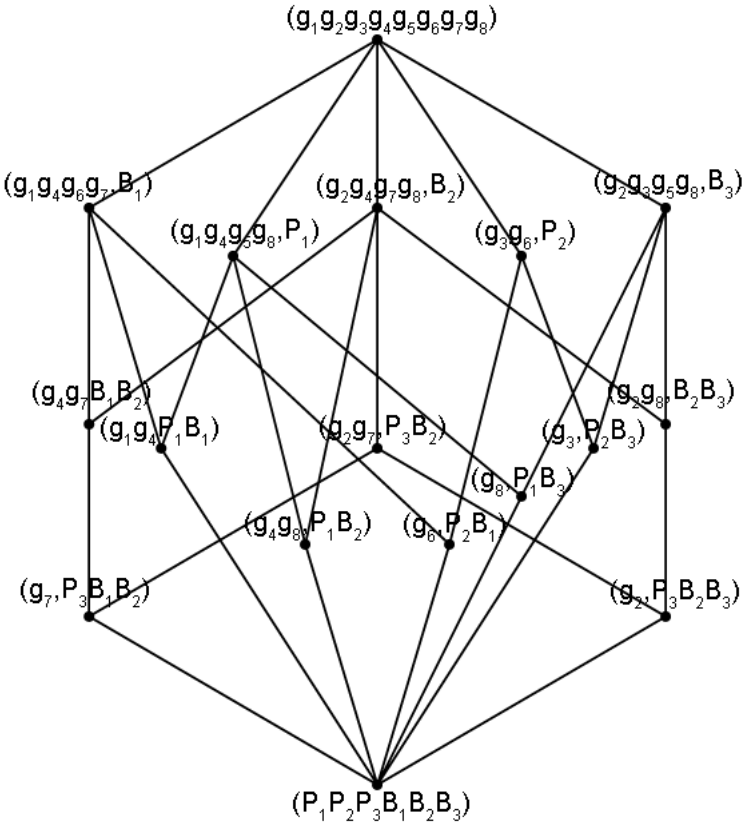


biological context as in Example 3.2.2. The Hasse diagram of the microBLAST lattice $\mu((G,E),\mathbf{A}_{D,T},C_B)$ is given in Figure 3.5.

Notice that there is a good deal more information contained in the microBLAST lattice in Figure 3.5 than in either of its generating lattices (see Figures 3.2 and 3.4). For instance, in the original expression lattice, the concept $K_e = (\{g_3, g_6\}, \{P_2\})$ is an atom (*i.e.* the only element below it is the minimal element $\hat{0}$). In the microBLAST lattice however, K has been refined and there are now two different concepts below it. This can be interpreted that the genes g_3 and g_6 are both similarly expressed, however, their protein products are involved in different functional activities (namely g_3 is involved in ATP transport while g_6 is involved in ATPase activity). Similarly, the concept $K_b = (\{g_4, g_7\}, \{B_1, B_2\})$ is also an atom in the biological lattice, however it has also been refined in the microBLAST lattice. An interpretation is that the protein encoded by the genes g_4 and g_7 have the same functionalities, however these genes are differently expressed in the experiment.

Refinements as discussed above are due to the integration of information represented in the expression lattice and the biological lattice. How powerful is this representation of the data?

Figure 3.5: Hasse diagram of the lattice $\mu((G, E), \mathbf{A}_{D,T}, C_B)$



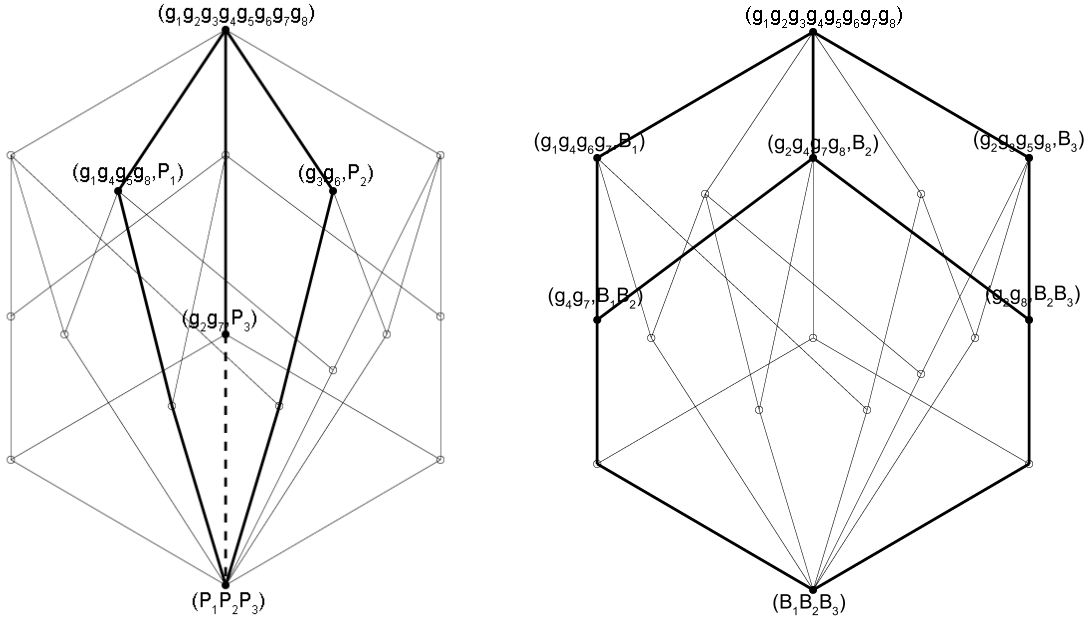
The remainder of this chapter describes the structure of a generic microBLAST lattice with a focus on what aspects are captured by computing the edit distance between two lattices.

The apposition of two contexts is fairly straightforward; however, as is evident in Figure 3.5, the algebraic structure of the lattice of two apposed contexts is not as easy to predict or interpret. Fortunately, 2.3.1 provides a general structure for the elements of the FCL of an apposition, however, a more exact description will assist in our understanding of the behavior of edit distance on the space of such lattices. Figure 3.6 illustrates how the lattices of C_{E_1} and C_B are embedded in the lattice of their apposition.

3.3 Edit distances between microBLAST lattices

The understanding of the general structure of $\underline{\mathfrak{B}}(C_{E_i}) \circledast \underline{\mathfrak{B}}(C_B)$ will assist in our understanding of edit distance as a metric for comparing microBLAST representations. Thankfully, Theorem 2.3.1 provides the structure of $\underline{\mathfrak{B}}(C_{E_i}) \circledast \underline{\mathfrak{B}}(C_B)$ allowing the analysis of its

Figure 3.6: The lattices $\underline{\mathfrak{B}}(C_{E_1})$ and $\underline{\mathfrak{B}}(C_B)$ are embedded in $\underline{\mathfrak{B}}(C_{E_1}) \circledast \underline{\mathfrak{B}}(C_B)$



concepts in terms of the concepts of $\underline{\mathfrak{B}}(C_{E_i})$ and $\underline{\mathfrak{B}}(C_B)$

1. A concept of $\underline{\mathfrak{B}}(C_{E_i})$ is a concept of $\underline{\mathfrak{B}}(C_{E_i}) \circledast \underline{\mathfrak{B}}(C_B)$ precisely when its extent is not contained in any extent of the elements of $\underline{\mathfrak{B}}(C_B)$.

2. A concept of $\underline{\mathfrak{B}}(C_B)$ is a concept of $\underline{\mathfrak{B}}(C_{E_i}) \circledast \underline{\mathfrak{B}}(C_B)$ precisely when its extent is not contained in any extent of the elements of $\underline{\mathfrak{B}}(C_{E_i})$.

3. For concepts (W, X) and (Y, Z) in $\underline{\mathfrak{B}}(C_{E_i})$ and $\underline{\mathfrak{B}}(C_B)$ respectively, $W \cap Y \neq \emptyset$ if and only if $(W \cap Y, (W \cap Y)')$ is a concept of $\underline{\mathfrak{B}}(C_{E_i}) \circledast \underline{\mathfrak{B}}(C_B)$.

We can also use Theorem 2.3.1 to make observations concerning the likelihood that two microBLAST lattices $\underline{\mathfrak{B}}(C_{E_1}) \circledast \underline{\mathfrak{B}}(C_B)$ and $\underline{\mathfrak{B}}(C_{E_2}) \circledast \underline{\mathfrak{B}}(C_B)$ have concepts in common where 1 and 2 refer to two different experiments (not necessarily the example experiments previously discussed).

For the analysis we must establish the scale used to discretize the data. For a gene expression data set (G, E_i) we will assume that the intraordinal scale A_{E_i,T_i} has been used to construct the single-valued context C_{E_i}. We will also assume the hypothesis concerning the lattices $\underline{\mathfrak{B}}(A_{E_i,T_i})$ of Lemma 3.2.1 are satisfied, that is, $ext(t_{i,1}) \cap ext(t_{i,2}) = \emptyset$ for all $t_1, t_2 \in T_i$ and $\underline{\mathfrak{B}}(A_{E_1,T_1}) \cong \underline{\mathfrak{B}}(A_{E_2,T_2})$. Because of the isomorphism, we can assume that the elements of T_i are labeled such that $t_{1,n}$ and $t_{2,n}$ are equivalent discrete states.

The size of the intersection of the extents of $t_{1,n}$ and $t_{2,n}$ in the two lattices will be of importance. Though an oversimplification, we will assume that each gene has equal probability of being discretized into one of the k attributes of T (i.e. $P((\phi(g, ev), t_{i,j}) \in J_i) = \frac{1}{k}$).

Observation 3.3.1. Assume $ext(t_{1,n}) \cap ext(t_{2,n})$ is large for every $t_{i,n} \in T_i$. In other words, all the corresponding clusters for the two experiments have many genes in common. Then, there is a greater likelihood that for $(X, Y) \in \underline{\mathfrak{B}}(C_B)$,

$$X \not\subset ext(t_{1,n}) \Leftrightarrow X \not\subset ext(t_{2,n}).$$

Hence by Item 2 above, there is a greater likelihood that

$$(X, Y) \in \underline{\mathfrak{B}}(C_{E_1}) \circledast \underline{\mathfrak{B}}(C_B) \Leftrightarrow (X, Y) \in \underline{\mathfrak{B}}(C_{E_2}) \circledast \underline{\mathfrak{B}}(C_B).$$

Therefore, a greater "similarity" in terms of expression value implies a greater "similarity" of microBLAST lattices.

Observation 3.3.2. Item 3 above will assist in our understanding of how the biological information "refines" the gene expression clusters and vice versa. Since we have assumed that $ext(t_{i,n}) \cap ext(t_{i,m}) = \emptyset$, every non-maximal or minimal concept of $\underline{\mathfrak{B}}(C_{E_i})$ is of the form $(ext(t_{i,n}), t_{i,n})$. By Theorem 2.3.1, the extent of every concept of $\underline{\mathfrak{B}}(C_{E_i}) \circledast \underline{\mathfrak{B}}(C_B) \backslash (\underline{\mathfrak{B}}(C_{E_i}) \cup \underline{\mathfrak{B}}(C_B))$ is of the form $(X \cap ext(t_{i,n}))$ where $(X, X') \in \underline{\mathfrak{B}}(C_B)$. Therefore, we want to know when $(X \cap ext(t_{1,n})) = (X \cap ext(t_{2,n}))$.

As in the above observation, if $ext(t_{1,n}) \cap ext(t_{2,n})$ is large, then there is a greater likelihood that $(X \cap ext(t_{1,n})) = (X \cap ext(t_{2,n}))$. Again, a greater "similarity" in terms of expression value implies a greater "similarity" of microBLAST lattices.

Observation 3.3.3. There is an inverse relation between the number of vertices identical between two graphs and their edit distance. As the above observations elucidate, if the gene expression signatures are similar, then there is a greater likelihood that edit distance between their microBLAST lattices will be relatively small. As was demonstrated in Section 3.2.1, edit distance performs poorly when comparing expression lattices. With the apposition of biological lattices, the performance of edit distance theoretically improves. Unfortunately, it has yet to be determined what the best type of biological information is. We have had success using protein motifs to construct biological lattices (see Chapter 4). Though such empirical results provide confidence, theoretical results determining the best structure or type relationships would provide a stronger foundation for the method. It is conceivable however, that the biological lattices to be apposed will be dependent on the situation and experimental conditions, as well as on the questions to be answered.

Observation 3.3.4. As has been seen in observations above, the larger the intersection $ext(t_{1,n}) \cap ext(t_{2,n})$ the more likely there will be a large number of concepts in common between the lattices $\underline{\mathfrak{B}}(C_{E_1}) \circledast \underline{\mathfrak{B}}(C_B)$ and $\underline{\mathfrak{B}}(C_{E_2}) \circledast \underline{\mathfrak{B}}(C_B)$. It is important to reiterate the assumptions made:

- Each gene had an equal probability of belonging to any cluster $t_{i,n}$. This is surely an oversimplification of reality, however, it allows us to gain a general sense of what the microBLAST lattices look like.
- An intraordinal scale was used to discretize the data and construct the expression lattice. Though such a scale has been used in all implementations of the microBLAST method, it is definitely not the only scale available. In fact, there is some concern that it is too simple a scale for discretization since it ignores most of the statistical information concerning the gene expression data. However, its simplicity does lend itself to ease of analysis as witnessed above.
- The scales used were isomorphic. As mentioned, this is a simple approach to discretizing the data which ignores the distribution of the gene expression values.

Chapter 4

Empirical Results

In the previous chapter, we outlined the algebraic structure of a microBLAST lattice as well as the theoretical abilities of such representations for microarray comparisons. Theoretical findings are of little use to the scientific community at large without a demonstration of the abilities of the method when working with actual data. In the following chapter we will describe two different experiments in which microBLAST lattices were constructed for all the microarray experiments in a database and edit distance was then used as a comparative metric for similarity detection. The empirical results are supportive of the microBLAST method and provide applicable understanding of the theoretical results described in Chapters 2 and 3.

4.1 Description of microBLAST software

According to the contemporary literature [101, 122], the fastest algorithm for constructing a FCL (both the concepts and the lattice structure) is LATTICE developed by C. Lindig [80]. In Lindig's algorithm, first the subconcepts of the maximal element $\hat{1}$ are constructed by computing the extent of each object and performing a local test to check if the corresponding concept is actually a subconcept of $\hat{1}$. The subconcepts of the subconcepts of $\hat{1}$ are constructed in a similar manner. This iterative process is repeated until all concepts are constructed. For a more detailed explanation of, as well as the theory behind, the algorithm, see [80].

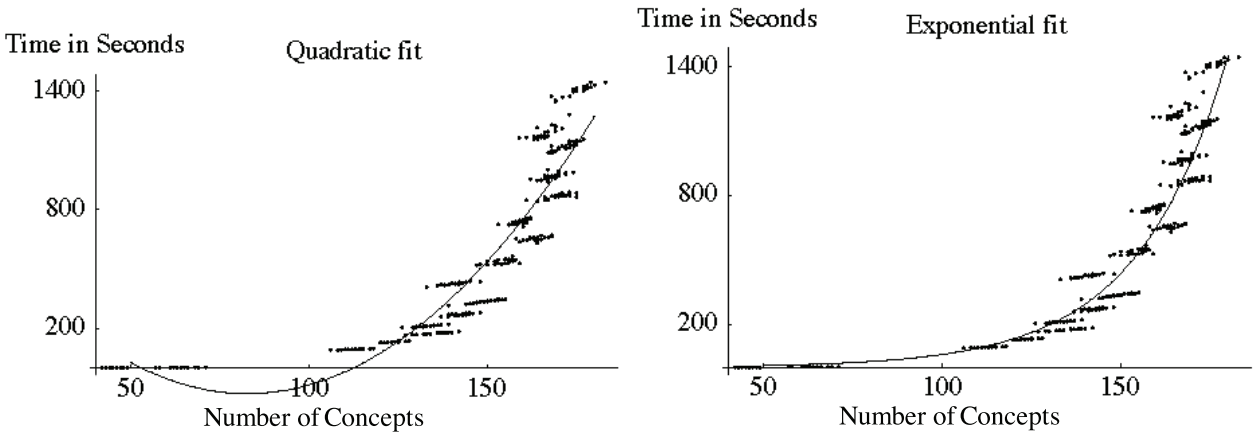
The asymptotic complexity of LATTICE for a context (G, M, I) is $O(|L| \times |G|^2 \times |M|)$ where $|L|$ is the number of concepts in the concept lattice [80]. In the same paper, Lindig presents empirical results that the run time for his implementation of the LATTICE algorithm is quadratic with respect to $|L|$ when working with contexts with sparsely filled context tables. Lindig's implementation of his algorithm was run on an otherwise idle 200 MHz AMD K6 Linux 2.0 system.

We implemented the LATTICE algorithm using the Java architecture. We were unable however to reproduce Lindig's impressive running time for our implementation. Conducting tests similar to Lindigs on an otherwise idle 1.33 GHz PowerPC G4 Mac OS X system, the run time grew exponentially with respect to $|L|$ (see Figure 4.1). It is assumed that Lindig used a more efficient data structure than ours and hence the difference in runtimes. The Java implementation of LATTICE was used to generate all lattices discussed in this chapter.

The algorithm for computing edit distances between lattices was also implemented in the Java architecture. The program uses unique numerical labeling of the concepts of a lattice to determine equality. This approach allows for a considerable speed of up runtime compared to an approach using direct comparisons of sets.

The entire package (lattice construction as well as comparisons via edit distance) will be referred to as the microBLAST software. As input, the software takes a folder of contexts saved in CSV format. It is assumed that each context has the same set of objects and

Figure 4.1: Quadratic and exponential fit to experimental run time of the implementation of the LATTICE algorithm.



attributes. The first row and column of the CSV file should be the labels for the attributes and objects respectively. A cell in an input file should have a value of 1 if the object has the attribute that labels the cell and 0 otherwise. The microBLAST software will (1) construct the FCL for each file in the input folder and (2) compute the edit distance between each lattice computed in the first step. A table of the edit distances between each pair of lattices is returned in CSV format by the program as well as TXT files for each lattice constructed that provides the following information about each concept in the lattice:

- the objects that comprise the concept,
- the attributes that comprise the concept,
- all the concepts directly above the concept,
- all the concepts directly below the concept.

The microBLAST software allows only the analysis of contexts provided by the user. Future versions of the software should be able to take a folder of contexts as input, compute their lattices, and then use edit distance to compare them to previously constructed lattices in an existing database. Another necessary improvement would be the ability to take raw gene expression data, a discretizing scale for the data, and a biological lattice as input and use the APPOSITION algorithm to compute the lattices.

4.2 Data sets employed in testing

4.2.1 Simulated gene expression data

Published data was selected from four microarray experiments reported in the literature using custom 2-color arrays and one unpublished experiment performed by Incyte Genomics Inc. using the dual-channel microarray format. All values were $\log_2$ ratios of infected samples versus mock-infected controls. The five *in vitro* experiments represent the following: three clinical isolates of cytomegalovirus (CMV) [56]; early host responses to the intracellular parasite *Toxoplasma gondii* [15]; early and intermediate host responses to *Trypanosoma cruzi* infection [121]; three strains of varicella-zoster virus [66]; and an influenza A lab strain A/WSN/33 [42]. The host cells were human foreskin fibroblasts, with the exception of the influenza experiment, where the cells were HEK293.

Due to the heterogeneity of the microarray platforms used in the different experiments, only 140 genes were common to all five experiments. The microBLAST method requires that the object sets are identical in all microBLAST lattices to be compared. Therefore, the 140 genes common to all five experiments comprised the object set of each microBLAST lattice representation.

All microarrays were dual channel style and were synthesized at Stanford University, save for the influenza samples which were synthesized by Incyte Genomics. The $\log_2$ expression ratios were averaged to create a standard host response for the strain or the time point. Missing values (less than 3% of the total) were assigned as zero, to reflect unchanged status in both conditions. Each pathogen's standard response was used as the basis for creating a population of simulated samples representing that class.

We created 24 copies of the gene expression vector associated with the standard host response and added pseudo-random noise to mimic the population structure of possible responses. The added noise was computed for each gene by randomly choosing a normal distribution, an exponential distribution or a uniform distribution and then adding noise reflecting the distribution. Three different standard deviation were used to create the different distributions in order to introduce different levels of noise. The maximum noise value added was 0.25, 0.35, and 0.50 to demonstrate degradation in performance as noise increases. Actual gene expression ratios ranged from approximately 5-fold up-regulated to 3-fold down-regulated. The end result was 175 microarray results (5 actual and 170 simulated) at three different maximum noise levels. See Figure 4.2 for the algorithm used to generate the simulated data.

Figure 4.2: Pseudo-code for generating simulated data generated from actual gene expression data

INPUT ← {ExpressionData, Number, σ}
ExpressionData is an $m \times n$ gene expression matrix with the columns corresponding to the expression values of the genes measured in the m different experiments
Number is the number of simulated experiments desired
σ is the standard deviation to be used to construct the different distributions

FOR each ROW of ExpressionData

COUNT = 0

WHILE COUNT < Number

CHOOSE $x \in \{1, 2, 3\}$ with $\mathrm{P}(x = j) = \frac{1}{3}$
CASES

x = 1, DIST = normal probability distribution with mean 0 and standard deviation σ

x = 2, DIST = uniform probability distribution with with variance $\frac{\sigma^2}{3}$

x = 3, DIST = exponential probability distribution with standard deviation σ^{-1}

NOISE ← random vector of length n with the entries chosen from DIST

multiply each entry of NOISE by -1 with probability $\frac{1}{2}$

NOISE ← NOISE + ROW

OUTPUT ← APPEND[OUTPUT, NOISE]
COUNT ← COUNT + 1

RETURN OUTPUT

An intraordinal scale with three attributes was used to transform the gene expression data into a single-valued context. The attributes consisted of three values with the first attribute, a_1, a threshold for which 1/3 of the genes in the experiment had expression value less than a_1, the second attribute, a_2, a threshold for which 2/3 of the genes in the experiment had expression value less than a_2, and the third attribute, a_3, the maximum gene expression value for the experiment. The expression lattices were all isomorphic to $\underline{\mathfrak{B}}(\mathbf{A}_{5,\{0,2,4,6\}})$ (see Figure 2.4 on page 31). The three concepts, other than the maximal and minimal elements, correspond to genes in the 33^{rd}, 66^{th}, and 100^{th} percentiles respectively.

ProSearch [73] was used to determine the motifs associated with the amino sequence of each gene. An ordinal scale was used to transform the motif information into a single-valued

context. The expression lattice for each gene expression sample was then apposed with the biological lattice to form the microBLAST representation of the microarray sample.

4.2.2 Reported gene expression data

Seventy-seven microarray experiments reported at NCBI Gene Expression Omnibus (www.ncbi.-nlm.nih.gov/geo) using the single-channel microarray format were selected. The *in vitro* experiments were obtained from three different labs and can be categorized as follows: 12 time-point expression profiles of foreskin fibroblasts infected by human cytomegalovirus (HCMV) [21]; 7 time-point expression profiles of HeLa cells infected with coxsackievirus B3 (CVB3) or control PBS [131] ; gene expression in macrophages and dendritic cells following exposure to 12 different pathogens that produce variable chronic infections [27]. All microarray platforms were single-channel Affymetrix HG-U95.

To account for background noise and measurement error in the different experiments, only genes with expression value greater than 100 in all 75 experiments were considered for constructing microBLAST lattices. The set of genes used was reduced further by defining the object set to consist only of the 500 genes with the highest variance across all 75 experiments. This reduction was performed for two reasons. First, the goal of the experiment was to test the microBLAST method with regards to distinguishing between biologically similar and dissimilar samples. It was hypothesized that the genes of greatest variance across the experiments were better candidates for distinguishing the host response being measured. Second, in order to construct the microBLAST lattices on a personal computer in a reasonable amount of time, a reduced number of genes needed to be used (for a plot of the algorithm;s runtime, see Figure 4.1).

The same type of intraordinal scale as used with the simulated data was used to construct the expression lattices.

Twenty-one protein motifs, describing the function of the proteins encoded by the genes considered in the above experiments, were used to construct a biological lattice. As in the simulated data experiment, ProSearch [73] was used to determine the motifs associated with the amino sequence of each gene. An ordinal scale was used to transform the motif information into a single-valued context. The expression lattice for each gene expression sample was then apposed with the biological lattice to form the microBLAST representation of the microarray sample.

4.3 Results

As a proof of concept, the two different types of microarray data sets described above were used to conduct microBLAST comparison experiments.

4.3.1 Using simulated data, microBLAST identified similarity

Edit distance was used to compare the microBLAST lattice of a reference experiment against the other 174 experiments in the database for the specified noise level. At all noise levels, the edit distance between samples derived from the same actual sample was typically less than the edit distance between pairs of samples derived from different actual samples. As more noise was added, this tendency decreased. Figures 4.3 and 4.4 shows the percentage of pairs of similar and dissimilar samples with edit distance within a given range. The results for the data sets built using 25% and 50% noise are shown.

Figure 4.3: The average edit distance distribution between pre-determined similar and dissimilar mock samples with 25% noise added.

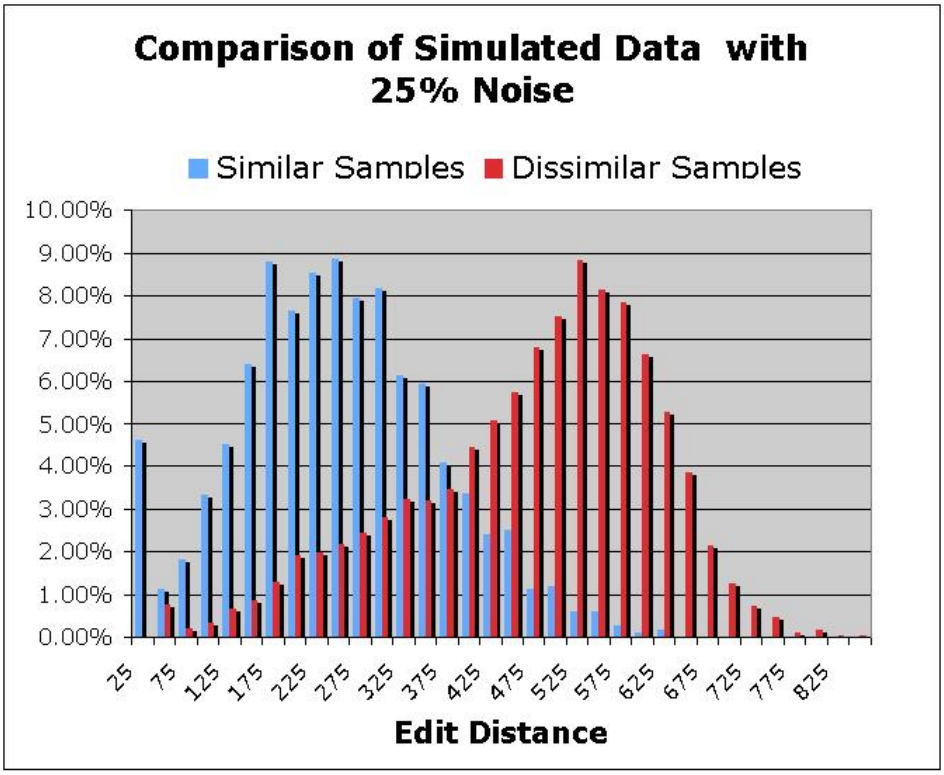
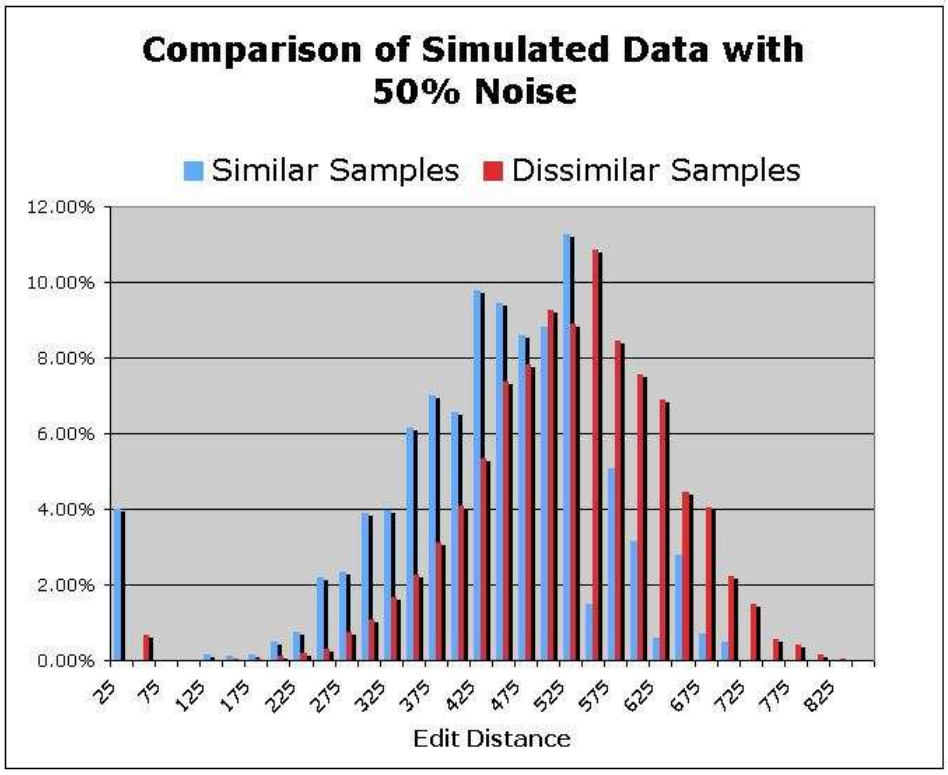


Figure 4.4: The average edit distance distribution between pre-determined similar and dissimilar mock samples with 50% noise added.

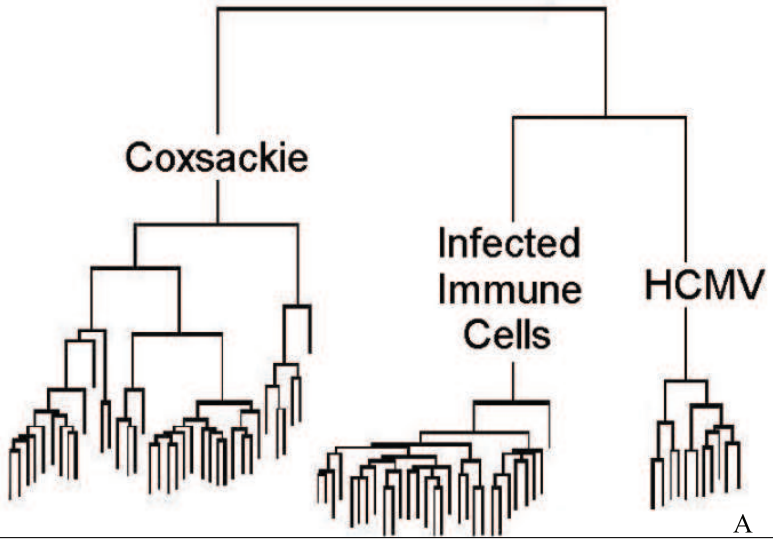


4.3.2 Using reported experimental data, microBLAST identified similarity

Edit distance was used to compare the microBLAST lattice of a reference experiment against the other 76 experiments in the database. Biological knowledge as well as hierarchical clustering of the samples was used to determine similar samples to be used in assessing microBLAST's capability of detecting similarities. Samples infected with the same pathogen were *a priori* considered more similar than those infected with a different pathogen. This assumption is supported by the hierarchical cluster in Figure 4.5.

The plot in Figure 4.6 demonstrates a typical analysis using microBLAST,*i.e.*, a reference sample is compared, via edit distance, to all other samples in a database of microarray experiments. To capture the cumulative results of comparing every sample in our database to the other 76 samples, the distributions for the edit distances between similar and dissimilar samples are plotted together in Figure 4.7.

Figure 4.5: Hierarchical clustering (A) of experimental data using Euclidean distance



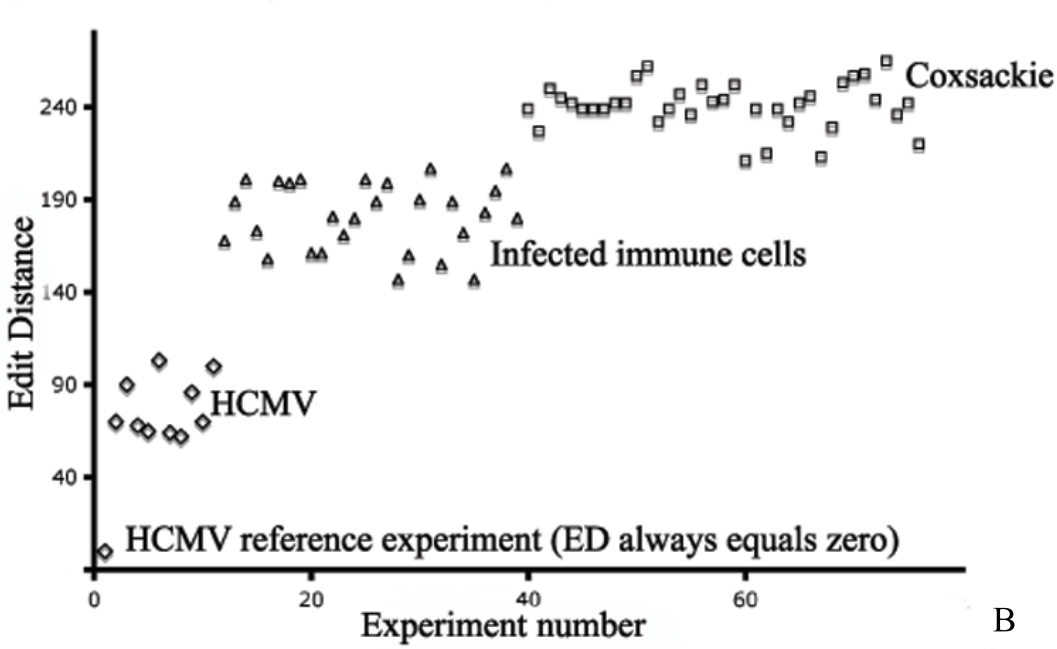
4.3.3 Systematically randomized data used to test validity of microBLAST findings

To further test microBLAST's ability to distinguish dissimilar data, the method was used on a nonsense data set of 22 gene expression samples. The data from the HCMV experiments were used as a template in that the expression level of a given gene in a given HCMV sample was randomly reassigned to a different gene as well as experiment with uniform probability. An intraordinal scale, as described in Section 4.3.1, was used to construct the expression lattice for the 22 nonsense data sets. The biological lattice constructed in Section 4.3.2 was apposed with each expression lattice to form the microBLAST representation. Edit distance was used to measure the differences between the microBLAST lattice representations for every pair in the non-sense data set. Figure 4.8 shows both the edit distance between the pairs of real HCMV time-series samples as well as between pairs of nonsense samples. The x- and y-axis are labelled by the sample (*i.e.*, time point) and color represents the edit distance value between the given samples. Black corresponds to edit distance zero and lighter colors to higher values.

4.3.4 Systematically perturbed data used to test robustness to noise

Samples in which progressively more and more genes are perturbed by noise were used as a second way to systematically assess the ability for microBLAST to detect similarities and

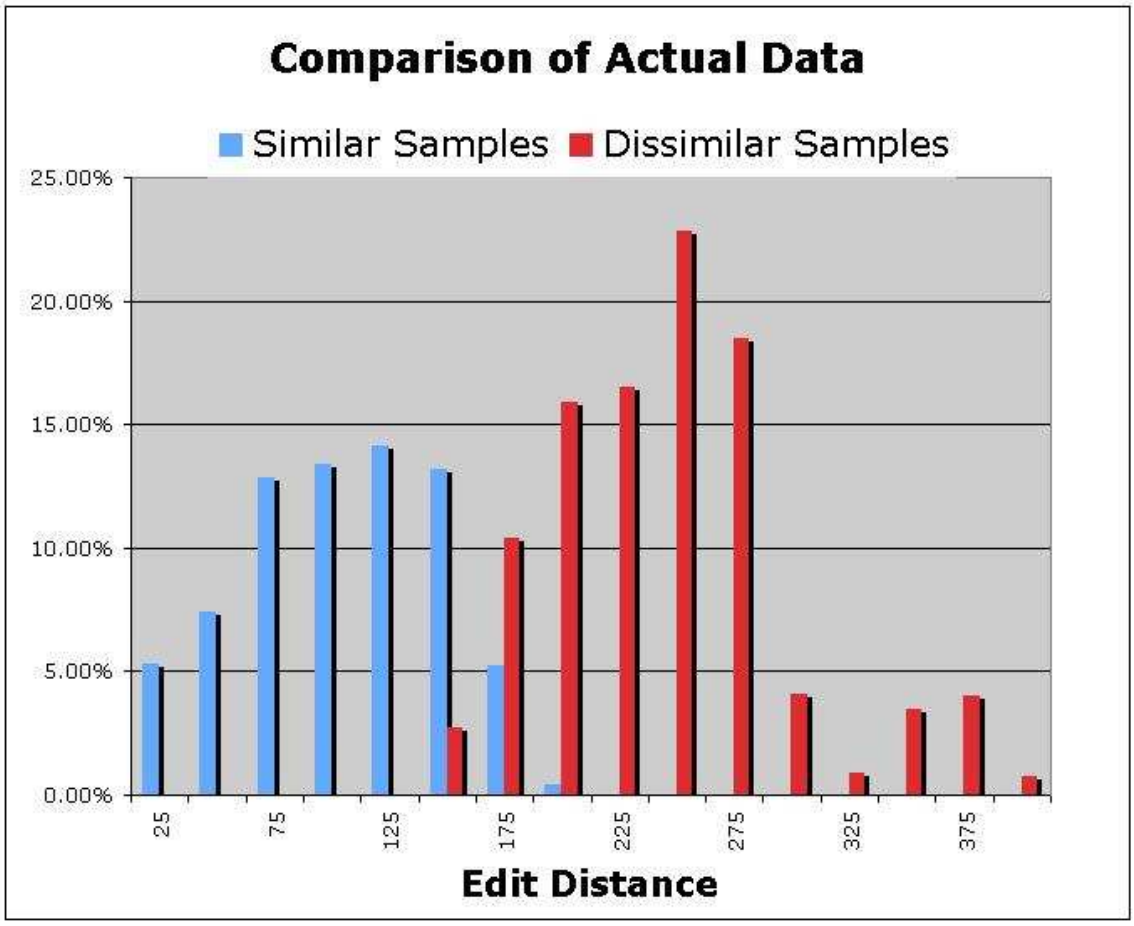
Figure 4.6: The plot (B) of the edit distance between a reference experiment and the other samples in the database.



dissimilarities. Ten actual samples were chosen and from each template a data set of 51 samples was created. The first simulated sample was created by adding normally distributed noise to 10 genes, chosen with a uniform probability, from the actual sample. The second simulated sample was created by adding noise to 10 previously unperturbed genes from the first simulated sample. As in the first step, genes were chosen with uniform probability and the noise was normally distributed. Subsequent samples were generated in this manner until all genes had been modified, *i.e.*, noise has been added to their expression values.

The same process as described in Section 4.3.3 was used to construct the microBLAST lattice representation for each of the 51 samples that were created. Figure 4.9 shows the average edit distance between the microBLAST lattice for the template sample and the lattice for its perturbed samples. The x-axis corresponds to the number of genes perturbed and the y-axis corresponds to the edit distance. The error bars measure the standard deviation from the mean.

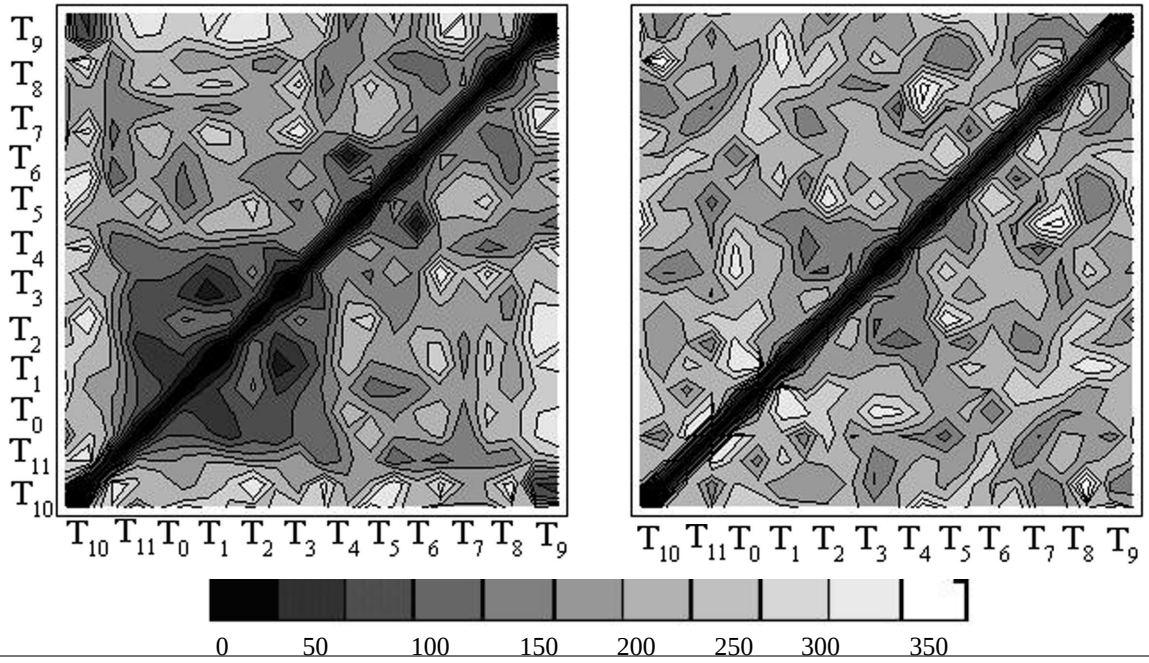
Figure 4.7: The average edit distance distribution between pre-determined similar and dissimilar samples.



4.4 Discussion of results

In Section 4.3.1 and 4.3.2, the edit distance (ED) between similar samples was on average less than the ED between dissimilar samples, supporting the validity of microBLAST as a method for detecting similar samples. With the distribution of EDs between similar samples differing very little from the distribution of EDs between dissimilar samples when 50% noise is added is Section 4.3.1, the robustness to noise of the method is brought into question. However, in Section 4.3.4 it was demonstrated that ED grows linearly with respect to increase in noise. Also, the distributions of EDs between similar and dissimilar samples are strikingly different in Section 4.3.2 suggesting that the noise added does not model realistic population structures. In light of these observations, we hypothesize that microBLAST, using edit distance, is robust with respect to noise.

Figure 4.8: Contour plots of the edit distances between samples from the real (left) and randomly redistributed (right) microarray experiments. Though the labels for the random data are not biologically significant, they are preserved for ease of readability.

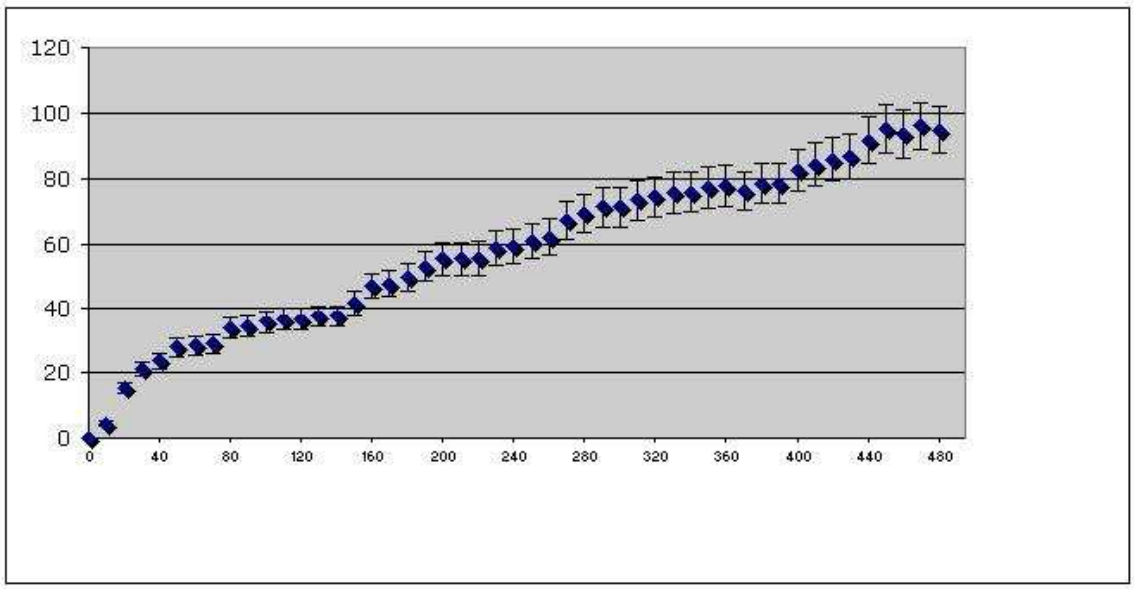


In Section 4.3.2 there is no pair of dissimilar samples with ED less than 150 (maximum ED between similar samples); it cannot be said with any statistical confidence that a pair of dissimilar samples is determined similar by ED,*i.e.*, the false discovery is zero in this experiment (see distribution of similar samples in Figure 4.7). A possible explanation is the heterogeneity of the data: all three sets are from different cell types infected with biologically unrelated pathogens. Differences in experimental conditions at the time of gene expression extraction could also explain the lack of unexpected similarities. Future standardizations and improvements of gene expression sampling will minimize such laboratory bias. Neither hypothesis is statistically supported at this point nor do they cover all possible explanations.

In Section 4.3.3, it is demonstrated that the discovery of similarity via edit distance cannot be attributed to random chance. The dark islands in Figure 4.8 correspond to groupings of samples with small edit distance between them. However, no such islands exist in the contour plot for the random samples.

The largest grouping of samples found in Figure 4.8 corresponds to samples that are temporally close to each other. Upon closer inspection of the edit distances between the HCMV samples, it was found that 97% of the samples collected before 17 hours post-infection (T_6) and 100% of the samples collected after 17 hours post-infection had edit distance less than 100 between them. However, only 25% of the pre-17 hour samples had edit distance less than

Figure 4.9: The average edit distance between a sample and its subsequently perturbed samples. The x-axis corresponds to the number of genes perturbed. The error bars for the standard deviation from the mean are also plotted.



100 from a post-17 hour sample. This might suggest that a shift in the cells' immune defense system occurs after the 17^{th} hour of infection; however, appropriately designed experiments would have to be performed to test this hypothesis.

As observed in Section 4.3.2, the groupings of similar experiments via the microBLAST method were comparable to groupings constructed using hierarchical clustering via Euclidean distance. Unlike hierarchical clustering, once groups of similar and dissimilar samples are established via the microBLAST method, one can analyze the microBLAST lattices to determine possible biological signatures that are common to all samples in a group. For instance, when analyzing the attributes that contribute to the detection of dissimilarity between the HCMV and CVB3 samples, it was found that genes that encode for surface receptors (a protein motif used to generate the biological lattice) had high expression values in the CVB3 samples. However, the same receptor genes were not highly expressed in the HCMV samples. In mouse models infected with CVB3, Taylor, *et al.*, found that the upregulation of a particular receptor gene appears to contribute to cell survival [117]. This suggests a possible functional difference between the survival strategies of HeLa cells infected with CVB3 and foreskin fibroblasts infected with HCMV.

The results obtained from the performed experiments support microBLAST as a method for identifying similarities and dissimilarities between microarray samples. Not only were expected similarities identified, but unexpected differences in the HCMV data were detected. It is unclear whether the differences in the HCMV data are due to biological changes in the

cell, though the detection of such differences allows one to design experiments in order to confirm or refute the hypothesis.

4.5 A comparison of microBLAST to other analysis methods

microBLAST lattices can be viewed as a clustering of the genes based on expression values as well as other biological information. In structure, they are most similar to hierarchical clusters in that at the top of the lattice are groups of genes that comprise large scale features of the data while groups of genes lower on the lattice capture more refined and specific details of the data. In classic clustering methods, groups of genes are identified according to their gene expression signature and the clusters are validated by the number of genes with similar biological function comprising the grouping. In a microBLAST lattice, biological functions are incorporated into the representation allowing for such validation analysis directly.

Recently there have been novel non-clustering approaches to microarray data analysis and sample comparisons. Related to the microBLAST method is BlastSets as discussed in the introduction. The acyclic graphs constructed in the BlastSets approach can be realized as concept lattices (though the authors of the method give no mention of FCA). For comparative analysis, the nodes of the graphs are intersected, and collections of genes that comprise statistically significant intersections are discovered. As we have seen, the microBLAST lattice is constructed by intersecting all the concepts of two different lattices (the expression and biological lattices). Therefore, the microBLAST lattice can be used to make discoveries similar to those made by BlastSets. In particular, if a statistical measure were used to analyze a microBLAST lattice, it is conceivable that all BlastSets findings could be reproduced. Our method is fundamentally different from BlastSets in that the attribute clusters are not the end result but are used to make comparisons of two samples to detect "global" or "system-wide" similarities.

ROAST is also like microBLAST in that it identifies microarray samples that are similar to a reference sample (Rosetta Biosoftware, Seattle WA). As ROAST is a proprietary program, publicly available descriptions of its underlying algorithm are unavailable. We have inferred from the available information [104] that ROAST takes only expression values as input and does not incorporate *a priori* biological information. The Lebesgue measures L_1 and L_2 have also been used for comparing microarray samples and do not require any reduction of data to make comparisons; as with ROAST, these measures do not incorporate biological knowledge into their comparisons.

The usefulness of such incorporated information has not been fully demonstrated; however the potential use in both the clinical and experimental setting is great. Attributes can capture information such as the chromosome on which the gene is located, protein motifs that the gene encodes, or biological pathways in which the proteins contribute. As we have

previously discussed, the incorporated biological information directly affects the structure of the microBLAST lattice and can be mathematically interpreted as a refinement of the groupings of the genes with respect to their expression values.

Chapter 5

Discussion

The size of databases in the public, and presumably private, sector that store high-throughput profiling experiments is growing rapidly. This new wealth of information provides an unprecedented resource to the research community. We have presented the foundations of a new method for the comparative analysis of microarray data and, theoretically, other types of high-throughput "-omics" data. The novel characteristics are two fold: a mathematical representation (the microBLAST lattice) of the gene activity is constructed that encodes information obtained from gene expression experiments as well as heterogeneous biological information; similarity of cellular response at the system level can be discovered via measures defined on the lattices.

Our method is founded in Formal Concept Analysis, a growing mathematical area of research with an international community of support. Though a relatively new area of analysis, FCA has strong roots in classic combinatorics thus allowing for a wide breadth of mathematics to build upon. The basic constructions of FCA are intuitively accessible to the novice, allowing for the use of our method for scientific discovery with little background knowledge of the field.

A deficiency of the present analysis is that only one metric of similarity has been investigated in detail. In Chapter 3 we have listed a number of other measures that could be employed for the comparative analysis of microBLAST lattices. As detailed previously, edit distance is a very strict measure of similarity and may be too restrictive for discovering subtle similarities. A benefit to the advancement of the microBLAST method would be a well-designed experiment in which multiple measures are used to compare microBLAST lattices. The ideal design would be one in which all or most of the biological aspects underlying the system were understood. Since no biological system is completely understood, a simulated gene regulatory system could be used to provide the required data.

Another aspect of the present method that may prove to be sub-optimal is that all expression lattices are isomorphic. This is a consequence of the scales used to discretize the gene expression data. An interesting approach would be to use a clustering method to group the genes and construct the gene expression lattice from these clusters. For instance, a hierarchical clustering of the data can be realized as an acyclic graph and hence a lattice. The hierarchical cluster would then be the expression lattice which is apposed with a biological lattice.

A natural approach to the analysis of mathematical objects is to decompose them into structurally simpler parts. Concept lattices are no exception. Conceivably, a decomposition of a microBLAST lattice may assist in the identification of particular biological pathways or substructures of functional importance to the underlying system. Further, lattice decompositions may provide theoretical understanding and/or classification of microBLAST lattices in general. Concept lattice decompositions will therefore be a future area of research.

Bibliography

[1] A Agrawal, *New institute to study systems biology*, Nature Biotechnology **17** (1999), no. 8, 743–744.

[2] Orly Alter, Patrick O Brown, and David Botstein, *Singular value decomposition for genome-wide expression data processing and modeling*, Proceedings of the National Academy of Science USA **97** (2000), no. 18, 10101–10106.

[3] S F Altschul, T L Madden, A A Schaffer, J Zhang, Z Zhang, W Miller, and D J Lipman, *Gapped BLAST and PSI–BLAST: a new generation of protein database search programs*, Nucleic Acids Research **25** (1997), 3389–3402.

[4] T Ando, M Suguro, T Hanai, T Kobayashi, H Honda, and M Seto, *Fuzzy neural network applied to gene expression profiling for predicting the prognosis of difuse large b-cell lymphoma*, Japan Journal Cancer Research **93** (2002), 1207–1212.

[5] Annenberg/CPB, *Determining gene function from sequence information*, http://www.learner.org/channel/courses/biology/textbook/genom/genom_6.html.

[6] Kellar Autumn, *Look up*, Ph.D. thesis, University of California and Berkeley, 1994.

[7] Emily C Baechler, Franak M Batliwalla, George Karypis, Patrick M Gaffney, Ward A Ortmann, Karl J Espe, Katherine B Shark, William J Grande, Karis M Hughes, Vivek Kapur, Peter K Gregersen, and Timothy W Behrens, *Interferon-inducible gene expression signature in peripheral blood cells of patients with severe lupus*, PROCEEDINGS OF THE NATIONAL ASSOCIATION OF SCIENCE USA **100** (2003), no. 5, 2610–2615.

[8] M Barbut and B Monjardet, *Ordre et classification*, Algebre et Combinatoire Tome II, Hachette, Paris, 1970.

[9] R Barriot, J Poix, A Groppi, A Barre, N Goffard, D Sherman, I Dutour, and A de Daruvar, *New strategy for the representation and the integration of biomolecularv knowledge at a cellular scale*, cbi.labri.fr/outils/data/blastsets/supplementary.pdf.

[10] ———, *New strategy for the representation and the integration of biomolecular knowledge at a cellular scale*, Nucleic Acids Research **32** (2004), no. 12, 3581–3589.

[11] L R Baugh, A A Hill, E L Brown, and H C P, *Quantitative analysis of mRNA amplification by in vitro transcription*, Nucleic Acids Research **29** (2001), no. 5.

[12] A Bernstein, E Kaufmann, C Buerki, and M Klein, *How similar is it? towards personalized similarity measures in ontologies*, Wirtschaftsinformatik, 2005, pp. 1347–1366.

[13] F Bertucci, *DNA arrays: technological aspects and applications*, Bulletins Cancer **88** (2001), no. 3, 243–522.

[14] _____, *Molecular typing of breast cancer: transcriptomics and DNA microarrays*, Bulletins Cancer **88** (2001), no. 3, 277–286.

[15] I J Blader, I D Manger, and J C Boothroyd, *Microarray analysis reveals previously unknown changes in Toxoplasma gondii-infected human cells*, Journal of Biological Chemistry **276** (2001), 24223–24231.

[16] Rudolf K Bock and Werner Krischer, *Data analysis briefbook*, Springer, 1998.

[17] Alvis Brazma and Jaak Vilo, *Gene expression data analysis*, Federation of European Biochemical Societies Letters **480** (2000), 17–24.

[18] S Brenner, F Jacob, and M Meselson, *An unstable intermediate carrying information from genes to ribosomes for protein synthesis*, Nature **190** (1961), 576–581.

[19] M Brown, W N Grundy, D Lin, N Critianini, C W Sugnet, T S Furey, M Ares, and D Haussler, *Knowledge-based analysis of microarray gene expression data by using support vector machines*, Procedings of the National Academy of Science **97** (2000), 262–267.

[20] P Brown and D Botstein, *Exploring the new world of the genome with DNA microarrays*, Nature Genetics Supplement **21** (1999), 33–37.

[21] E P Browne, B Wing, D Coleman, and Shenk T, *Cytomigalovirus*, 2003.

[22] H J Bussermaker, H Li, and E D Siggia, *Regulatory element detection using correlation with expression*, Nature Genetics **27** (2001), 167–174.

[23] C Carpineto and G Romano, *Galois: An order-theoretic approach to conceptual clustering*, Proceding of the 10th Conference on Machince Learning (Amherst MA), Kaufmann, 1993, pp. 33–40.

[24] _____, *Information retrieval through hybrid navigation of lattice representations*, International Journal of Human-Computer Studies **45** (1996a), no. 5, 553–578.

[25] _____, *A lattice conceptual clustering system and its application to browsing retrieval*, Machine Learning **24** (1996b), no. 2, 1–28.

[26] W C Chang, *On using principal components before separating a mixture of two multivariate normal distributions*, Applied Statistics, 267–275.

[27] D Chaussabel, R T Semnani, M A McDowell, D Sacks, A Sher, and T B Nutman, *Pathogen exposure*, http://www.ncbi.nlm.nih.gov/geo/gds/gds_browse.cgi?gds=260, 2004.

[28] Y Chen, E R Dougherty, and M L Bittner, *Ratio based decisions and the quantitative analysis of cDNA microarray images*, Journal of Biomedical Optics **2** (1997), 364–374.

[29] H Chipman, T Hastie, and R Tibshirami, *Clustering microarray data*, Statistical analysis of gene expression microarray data (T Speed, ed.), Chapman & Hall/CRC, 2003, pp. 159–200.

[30] F Chung, *Spectral graph theory*, AMS, Providence, RI, 1997.

[31] R Cole and G Stumme, *Cem-a conceptual email manager*, Conceptual Structures: Logical, Linguistic and Computational Issues. LNAI 1867 (B Ganter and G Mineau, eds.), Berlin: Springer, 2000, pp. 438–452.

[32] GO Consortium, *An introduction to the gene ontology*, http://www.geneontology.org/GO.doc.shtml.

[33] _____, *What is go?*, http://www.ebi.ac.uk/faq/cgi-bin/go?editCmds=hide-&file=10&keywords=login&showAttributions=hide&showLastModified=hide-&showModerator=hide.

[34] F H C Crick, L Barnett, S Brenner, and R J Watts-Tobin, *General nature of the genetic code for proteins*, Nature **192** (1961), 1227–1232.

[35] Francis Crick, *Centeral dogma of molecular biology*, Nature **227** (1970), 561–563.

[36] M J Cunningham, *Genomics and proteomics: the new millennium of drug discovery and development*, Journal of Pharmacological Toxicology Methods **44** (2000), no. 1, 291–300.

[37] D M Cvetkovi'c, M Doob, and H Sachs, *Spectra of graphs*, Academic Press, 1980.

[38] C Debouck and P N Goodfellow, *DNA microarrays in drug discovery and development*, Nature Genetics **21** (1999), no. 1, 48–50.

[39] Patrik D'haeseleer, Shoudan Liang, and Roland Somogyi, *Genetic network inference: from co-expression clustering to reverse engineering*, Bioinformatics **16** (2000), no. 8, 707–726.

[40] E Domany, *Superparamagnetic clustering of data — the definitive solution of an ill-posed problem*, 1999.

[41] Sorin Drăgici, *Data analysis tools for DNA microarrays*, Chapman & Hall/CRC, 2003.

[42] K Duca, *Host responses to influenza a infection in 293 cells*, 2003, unpublished.

[43] D E Duffy and A J Quiroz, *A permutation-based algorithm for block clustering*, Journal of Classification **8** (1991), 65–91.

[44] D Duggan, m Bittner, Y Chen, P Meltzer, and J Trent, *Expression profiling using cDNA microarrays*, Nature Genetics **21** (1999), 10–14.

[45] M B Eisenand, P T Spellmanand, P O Brownand, and D Botstein, *Cluster analysis and display of genome-wide expression patterns*, Procedings fo the National Acadademy of Science USA **95** (1988), 14863–14868.

[46] K M Eyster and R Lindahl, *Molecular medicine: a primer for clinicians part xii: DNA microarrays and their application to clinical medicine*, S D J Med **54** (2001), no. 2, 57–61.

[47] A Fadiel and F Naftolin, *Microarray applications and challenges: a vast array of possibilities*, Int Arch Biosci (2003), 1111–1121.

[48] Andrew G Fraser and Edward M Marcotte, *Aprobabilistic view of gene function*, Nature Genetics **36** (2004), no. 6, 559–564.

[49] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er, *Using bayesian networks to analyze expression data*, Journal of Computational Biology **7** (2000), no. 3-4, 601–620.

[50] B Ganter and R Wille, *Contextual attribute logic*, Conceptual Structures: Standards and Practices. Proceedings of the 7th International Conference on Conceptual Structures, LNAI 1640 (W Tepfenhart and W Cyre, eds.), Berlin: Springer, 1999, pp. 377–388.

[51] Gad Getz, Erel Levine, and Eytan Domany, *Coupled two-way clustering analysis of gene microarray data*, Proceeding of the National Academy of Science USA **24** (2000), no. 22, 12079–12084.

[52] A Gierer, *Theoretical approaches to holistic biological features: Pattern formation, neural networks and the brain-mind relation*, Journal Bioscience **27** (2002), 195–205.

[53] R Godin, JJ Gecsei, and C Pichet, *Design of browsing interface for information retrieval*, Proceedings SIGIR '89 (N J Belkin and C J van Rijsbergen, eds.), 1989, pp. 32–39.

[54] R Godin and H Mili, *Building and maintaining analysis-level class hierarchies using galois lattices*, OOPSLA '93 ACM Sigplan Notices, vol. 28, 1993, pp. 394–410.

[55] T R Golub, D K Slonim, P Tamayo, C Huard, M Gaasenveek, J P Mesirov, H Coller, M L Loh, J R Downing, and M A Caligiuri, *Molecular classificatin of cancer: class discovery and class prediction by gene expression monitoring*, Science **286** (1999), 531–537.

[56] L Graf, *Cytomegalovirus microarray data, geo accession: Gse501*, 2003, Torok-Storb Lab, Clinical Research Division.

[57] Ralf Gugisch, *Many-valued context analysis using descriptions*, Proceedings of the 9th International Conference on Conceptual Structures: Broadening the Base (Harry S Delugach and Gerd Stumme, eds.), pp. 157–168.

[58] T Hastie, R Tibshirani, M B Eisen, A Alizadeh, R Levy, L Staudt, W C Chang, D Botstein, and P O Brown, *'gene shaving' as a method for identifying distinct sets of genes with similar expression patterns*, Genome Biology **1** (2000), 1–21.

[59] Celia M Henry, *Systems biology*, Chemical and Engineering News **83** (2005), no. 7, 47–55.

[60] Mae-Wan Ho, *Death of the centeral dogma*, Science in Society **24** (2004).

[61] Leroy Hood and David Galas, *The digital code of DNA*, Nature **421** (2003), 444–448.

[62] T Ideker, V Thorsson, A F Siehel, and LE Hood, *Testing for differentially-expressed genes by maximum likelihood analysis of microarray data*, Journal of Computational Biology **7** (2000), 805–817.

[63] Trey Ideker, Timothy Galitski, and Leroy Hood, *A new approach to decodin life: Systems biology*, Annual Review of Genomics and Human Genetics **2** (2001), no. 1, 343–372.

[64] JeannPierre Issa, *Cpg island methylator phenotype in cancer*, Nature **4** (2004), 988–993.

[65] I T Jolliffe, *Principle component analysis*, Springer, 1986.

[66] Jeremy O Jones and Ann M Arvin, *Microarray analysis of host cell gene transcription in response to varicella-zoster virus infection of human t cells and fibroblasts in vitro and scidhu skin xenografts in vivo*, Journal of Virology **77** (2003), no. 2, 1268–1280.

[67] N Kaminski and N Friedman, *Practical approaches to analyzing results of microarray experiments*, American Journal of Respiratory Cell and Molecular Biology **27** (2002), 125–132.

[68] M Kerr, M Martin, and G Churchill, *Analysis of variance for gene expression microarray data*, Journal of Computational Biology **7** (2000), 819–37.

[69] M K Kerr and G A Churchill, *Statistical design and the analysis of gene expression microarray data*, Genet Res **77** (2001), no. 2, 123–128.

[70] J Khan, L H Saal, M L Bittner, Y Chen, J M Trent, and P S Meltzer, *Expression profiling in cancer using cDNA microarrays*, Electrophoresis **20** (1999), 223–229.

[71] Hiroaki Kitano, *Systems biology: Toward system-level understanding of biological systems*, Foundations of Systems Biology (H Kitano, ed.), MIT Press, Cambridge, MA, 2001.

[72] ———, *Systems biology: a brief overview*, Science **295** (2002), 1662–1664.

[73] L F Kolakowski, J A Leunissen, and J E Smith, *Prosearch: fast searching of protein sequences with regular expression patterns related to protein structure and function*, Biotechniques **13** (1992), no. 6, 919–921.

[74] S Kuznetsov, *Machine learning and formal concept analysis*, Concept Lattices: Second International Conference on Formal Concept Analysis, LNCS 2961 (P Eklund, ed.), Berlin: Springer, 2004, pp. 287–312.

[75] L Laxxeroni and A B Owen, *Plaid models for gene expression data*, Tech. report, Department of Statistics, Stanford University, 2000.

[76] M-L T Lee, F C Kuo, G A Whitmore, and J Sklar, *Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations*, Proceedings of the National Academy of Science **97** (2000), 9834–9839.

[77] Su-In Lee and Serafim Batzoglou, *Application of independent component analysis to microarrays*, Gene Biology **4** (2003), no. 11.

[78] C Li, G Tseng, and W Wong, *Model-based analysis of oligonucleotide arrays and issues in cDNA microarray analysis*, Statistical analysis of gene expression microarray data (T Speed, ed.), Chapman & Hall/CRC, 2003, pp. 1–34.

[79] W Liebermeister, *Linear modes of gene expression determined by independent component analysis*, Bioinformatics **18** (2002), 51–60.

[80] C Lindig, *Fast concept analysis*, Ph.D. thesis, Harvard University: Cambridge, 2000.

[81] R Lipshuytz, S fodor, T Gingeras, and D Lockhart, *High density synthetic oligonucleotied arrays*, nature Genetics **21** (1999), no. 1, 20–24.

[82] Li Liu, Douglas M Hawkins, Sujoy Ghosh, and S Stanley Young, *Robust singular value decomposition analysis of microarray data*, PROCEEDINGS OF THE NATIONAL ASSOCIATION OF SCIENCE USA **100** (2003), no. 23, 13167–13172.

[83] Brenda Maddox, Nature **421** (2003), 407–408.

[84] P Mangiameli, S K Chen, and D West, *A comparison of som neural network and hierarchical clustering methods*, European Journal of Operational Research **93** (1996), 402–417.

[85] Gregor Mendel, *Versuche ber plflanzen-hybriden*, Verhandlungen des naturforschenden Ver-eines in Brnn **IV** (1866), 3–47.

[86] Bruno T Messmer and H Bunke, *Subgraph isomorphism in polynomial time*, Tech. Report 95-003, IAM, 1995.

[87] H Mili, E Ah-Ki, R Godin, and H Mcheick, *Another nail to the coffin of faceted controlled-vocabulary component classification and retrieval*, ACM SIGSOFT Software Engineering Notes **22** (1997), no. 3, 89–98.

[88] Jatin Misra, William Schmitt, Daehee Hwang, Li-Li Hsiao, Steve Gullans, George Stephanopoulos, and Gregory Stephanopoulos, *Interactive exploration of microarray gene expression patterns in a reduced dimensional space*, Genome Research **12** (2002), no. 7, 1112–1120.

[89] B J T Morgan and A P G Ray, *Non-uniqueness and inversions in cluster analysis*, Applied Statisics **44** (1995), 114–134.

[90] T H Morgan, *Sex-limited inheritance in drosophila*, Science **32** (1910), 120–122.

[91] M A Newton, C M Kendziorski, C S Richmond, F R Blattner, and K W Tsui, *On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data*, Journal of Computational Biology **8** (2001), 37–52.

[92] U.S. Department of Energy Genomics:GTL Program, http://doegenomestolife.org.

[93] Stephen G Oliver, *Functional genomics: lessons from yeast*, Philosophical Transactions: Biological Sciences **357** (2002), no. 1417, 12–14.

[94] Board on Life Sciences, *Seeking security: Pathogens, open access, and genome databases*, Tech. report, Board on Life Sciences, 2004.

[95] W Pan, *A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments*, Bioinformatics **18** (2002), no. 4, 546–554.

[96] Wei Pan, Jizhen Lin, and Chap T Le, *A mixture model approach to detecting differentially expressed genes with microarray data*, Funct Integr Genomics **3** (2003), 117–124.

[97] Mildred L Patten, *Understanding research methods: An overview of the essentials*, Pyrczak Publishing, 2002.

[98] Scott D Patterson and Ruedi H Aebersold, *Proteomics: the first decade and beyond*, Nature Genetics Supplement **33** (2003), 311–323.

[99] U Priss, *Relational concept analysis: Semantic structures in dictionaries and lexical databases*, Ph.D. thesis, Technische Hochschule Darmstadt, Germany, 1998.

[100] ———, *Linguistic applications of formal concept analysis*, Formal Concept Analysis-State of the Art, Proceedings of the First International Conference on Formal Concept Analysis, Berlin: Springer, 2004.

[101] Uta Priss, *Formal concept analysis in informaition science*, Annual Review of Information Science and Technology **40**.

[102] PubMed, http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed.

[103] D R Rhodes, J Yu, K Shanker, N Deshpande, R Varambally, D Ghosh, T Barrette, A Pandey, and A M Chinnaiyan, *Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression*, Proceedings of the National Academy of Science USA **101** (2004), 9309–9314.

[104] ROAST, *Roast*, 2005, http://www.rosettabio.com/products/resolver/multidim.htm.

[105] C Sabatti, *Statistical issues in DNA microarrays*, Current Genomics **3** (2002), 7–15.

[106] S S Sahapiro, *statistical modeling techniques*, Marcel Dekker, INC, 1981.

[107] E Segal, A Battle, and D Koller, *Decomposing gene expression into cellular processes*, Proceedings of the Eighth Pacific Symposium on Biocomputing (R B Altman, A K Durker, L Hunter, T A Jung, and T E Klein, eds.), World Scientific Publishing Company, 2003, pp. 89–100.

[108] Gordon K Smyth, Yee Hwa Yang, and Terry Speed, *Statistical issues in cDNA microarray data analysis*, Functional Genomics: Methods and Protocols (Methods in Molecular Biology) (M J Brownstein and A B Khodursky, eds.), vol. 224, Humana Press, pp. 111–136.

[109] N Spangenberg and K E Wolff, *Datenreduktion durch die formale begriffsanalyse von repertory grids*, Einfuhrung in die Repertory Grid Technik (J W Scheer and A Catina, eds.), Verlag Hans Huber, 1993, pp. 38–54.

[110] L D Stein, *Human genome: End of the beginning*, Nature **431** (2004), 915–916.

[111] M-C Su and H-T Chang, *Fast self-organizing feature map algorithm*, IEEE-NN **11** (2000), no. 3, 721.

[112] A Szabo, K Boucher, W L Carroll, L B Klebanov, A D Tsodikov, and A Y Yakovlev, *Variable selection and pattern recognition with gene expression data generated by the microarray technology*, Math Bioscience **176** (2002), no. 1, 71–98.

[113] P Tamayo, D Slonim, J Mesirov, Q Zhu, S Kitareewan, E Dmitrovsky, E S Lander, and T R Golub, *Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation*, Procedings of the National Academy of Science USA **96** (1999), 2907–2912.

[114] Chun Tang, Li Zhang, Aidong Zhang, and Murali Ramanathan, *Interrelated two-way clustering: An unsupervised approach for gene expression data analysis*, 2nd IEEE International Symposium on Bioinformatics and Bioengineering, IEEE Press, 2001, pp. 41–48.

[115] S Tavazoie, J D Hughes, M J Campbell, R J Cho, and G M Church, *Systemaic determination of genetic network architecture*, Nature Genetics **22** (1999), 281–285.

[116] _____, *Systematic determination of genetic network architecture*, Nature Genetics **22** (1999), 281–285.

[117] L A Taylor, C M Carthy, D Yang, K Saad, D Wong, G Schreiner, L W Stanton, and B M McManus, *Host gene regulation during coxsackievirus b3 infection in mice: assessment by microarrays*, Circulatory Research **87** (2000), no. 4, 328–334.

[118] V G Tusher, R Tibshirani, and G Chu, *Significance analysis of microarrays applied to the ionizing radiation response*, Proceedings of the National Academy of Science **98** (2001), 5116–5121.

[119] unkown, *Editorial*, Nature **226** (1970), 1198.

[120] _____, *Proteomics, transcriptomics: what's in a name?*, Nature **402** (1999), 715.

[121] Silvia Vaena de Avalos, Ira J Blader, Michael Fisher, John C Boothroyd, and Barbara A Burleigh, *Immediate/early response to Trypanosoma cruzi infection involves minimal modulation of host cell transcription*, J Biol Chem **4** (2002), no. 277(1), 639–644.

[122] P Valtchev, R Missaoui, and P Lebrun, *A fast algorithm for building the hasse diagram of a galois lattice*, Tech. report, Labratory for Research on Technology for Ecomerce, 2000.

[123] Ludwig von Bertalanffy, *General system theory*, George Brazillier and Inc., 1968.

[124] Andrew C von Eschenbach, *A vision for the national cancer program in the united states*, Nature Review Cancer **4** (2004), 820–828.

[125] J D Watson and F H Crick, *A structure for deoxyribose nucleic acid*, Nature **171** (1953), 737–738.

[126] Norbert Wiener, *Cybernetics or control and communication in the animal and the machine*, MIT Press and Cambridge and MA, 1948.

[127] Wikipedia, *Ec number*, http://en.wikipedia.org/wiki/EC_number.

[128] R Wille, *Restructuring lattice theory: an approach based on hierarchies of concepts*, Ordered sets (I Rival, ed.), Reidel, Dordrecht-Boston, 1982, pp. 445–470.

[129] Russell D Wolfinger, Greg Gibson, Elizabeth D Wolfinger, Lee Bennett, Hisham Hamadeh, Pierre Bushel, Cynthia Afshari, and Richard S Paules, *Assessing gene significance from cDNA microarray expression data via mixed models*, Journal of Computational Biology **8** (2001), no. 6, 625–637.

[130] Ching-Chang Wong and Chia-Chong Chen, *A hybrid clustering and gradient descent approach for fuzzy modeling*, IEEE Transactions on Systems, Man, and Cybernetics, Part B **29** (1999), no. 6, 686–693.

[131] B Yanagawa, H Luo, N Rezai, Z Hollander, R T Ng, J Yuan, J Zhang, D Yang, T J Triche, and B McManus, *Coxsackievirus*, 2003.

[132] K Y Yeung and W L Ruzzo, *Principal component analysis for clustering gene expression data*, Bioinformatics **17** (2001), no. 9, 763–774.

Vita

An eighth-generation native of Southern California, Dustin Potter grew up with his Mother, Aunt, Sister, and two cousins. The eldest, he is the first in his family to earn a graduate degree. Following high school, he enrolled in the school of hard-knocks with a focus in travel, construction, and culinary arts. In 1996, Dustin attended South Puget Sound Community College where he received an A.A.. He continued his education at The Evergreen State College where he earned a B.S. with a focus in Mathematics. He subsequently earned a M.S. and PhD in mathematics at Virginia Tech in 2001 and 2005 respectively. In 2004 and 2005 he held a graduate research position at the Virginia Bioinformatics Institute. He is currently a Postdoctoral Research Assistant at the Cancer Research Center at The Ohio State University working under the direction of Dr. Tim Huang. He is a member of American Mathematical Society, Society for Advancement of Chicanos and Native American in Science, and Society for Industrial and Applied Mathematics.