Winter 11-10-2020

Formal Concept Analysis Applications in Bioinformatics

Sarah Roscoe
University of Nebraska-Lincoln, sroscoe@huskers.unl.edu

FORMAL CONCEPT ANALYSIS APPLICATIONS IN BIOINFORMATICS

by

Sarah R. Roscoe

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Jitender S. Deogun

Lincoln, Nebraska

November, 2020

FORMAL CONCEPT ANALYSIS APPLICATIONS IN BIOINFORMATICS

Sarah R. Roscoe, MS

University of Nebraska, 2020

Adviser: Jitender S. Deogun

Bioinformatics is an important field that seeks to solve biological problems with the help of computation. One specific field in bioinformatics is that of genomics, the study of genes and their functions. Genomics can provide valuable analysis as to the interaction between how genes interact with their environment. One such way to measure the interaction is through gene expression data, which determines whether (and how much) a certain gene activates in a situation. Analyzing this data can be critical for predicting diseases or other biological reactions. One method used for analysis is Formal Concept Analysis (FCA), a computing technique based in partial orders that allows the user to examine the structural properties of binary data based on which subsets of the data set depend on each other. This thesis surveys, in breadth and depth, the current literature related to the use of FCA for bioinformatics, with particular focus on gene expression data. This includes descriptions of current data management techniques specific to FCA, such as lattice reduction, discretization, and variations of FCA to account for different data types. Advantages and shortcomings of using FCA for genomic investigations, as well as the feasibility of using FCA for this application are addressed. Finally, several areas for future doctoral research are proposed.

DEDICATION

This thesis is dedicated to the people who have supported me through my time so far in graduate school. First and foremost, I thank my parents, who, from the very beginning, nurtured and encouraged my insatiable curiosity and desire to figure out how the world worked; who challenged me to see the value in hard work and thankless tasks, and to succeed and fail on my own merits. I thank my advisor, Dr. Jitender Deogun, as since I have met him, he has encouraged me and impressed on me the exact same, as well as the desire to, in everything, work hard and have faith. I thank my friends, officemates, and classmates, and siblings, especially Aubrey, Becca, Donna, Evan, Felicia, Molly, Ryan, and many more, for their unending friendship and support.

ACKNOWLEDGMENTS

I wish to thank Minal Khatri, who contributed to an early draft of Chapters 3 and 4, and Adam Voshall, for his helpful conversations about FCA and gene expression data.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computation has been used to solve several important biological problems in the past few decades. A principal example is the Human Genome Project [1], an effort in the early 2000's, which worked to produce a database of human genome sequences, or sequences of molecular structures that instruct a cell what to do in response to a situation. The database produced is still being analyzed today to determine which areas produce biological functions, and identify how variations and mutations in genes can lead to development of diseases, such as cancer or Alzheimer's. The project is part of the field of genomics, the study of genes, their functions, and their inter-relationships. Genomics is distinguished from genetics, which is the study of individual genes and their hereditary properties [3].

In studying groups of genes, it is helpful to see how a gene responds to a biological situation, whether lab- or nature-induced. This response is called gene expression, and expression data is often measured from microarrays, chips that store up to tens of thousands of individual DNA strands. Through a biological process called the Central Dogma of molecular biology [27], RNA transcriptions of genes can be analyzed with Northern blotting to examine gene responses [2, 71, 59].

After the success of the Human Genome Project, much attention was given to how large amounts of genomic data can be analyzed. Clustering [49] is an important area, as assigning genes with similar expression levels can reveal which genes work together in situations. Clustering can be done with Formal Concept Analysis (FCA), a theoretical framework of computing based on partial orders that allows the user to examine the structural properties of binary data based on which subsets of the dataset depend on each other. FCA was developed in the 1980s by Wille to revitalize lattice theory, part of abstract algebra, to provide benefits of examining the structure of data to other fields [91]. While FCA was originally developed for binary data, variations and extensions have been a popular area of research, such as for fuzzy [17, 65], probabilistic [31, 28, 88], or time series data [54]. Since its conception, FCA has benefited fields looking at the structure of its data, ontologies, and other areas where hierarchical analysis is helpful.

FCA works by first determining concepts, subsets of the dataset that are marked by maximal Trues. Then, a mathematical structure called a lattice is developed to hierarchically examine the structure of those maximal subsets. For smaller datasets, the lattice can also be visualized to uncover implicit information about the data and how certain areas depend on one another. Implementations to visualize the lattice, such as LatViz and Concept Explorer [6, 92], were developed in the early 2000s. For larger datasets, the lattice may be prohibitively complex due to the potentially exponential number of concepts. This exponential risk is one of the main drawbacks of using FCA.

Over the last few decades, FCA has been used to analyze various types of data relating to bioinformatics [44, 20, 68, 76, 41, 23, 81, 80]. The majority of studies, however, have focused on using Formal Concept Analysis to analyze gene expression data [79, 49, 75, 26, 21, 48, 22, 57, 53, 56, 47, 85, 40, 11, 12, 62, 58, 7, 8, 9]. Early uses of FCA to analyze gene expression data attempted to identify groups of genes with similar expression levels [79, 21, 75, 26]. These groups are called co-expressed genes. While some formative studies in the field, such as Potter [75] and Choi [26] used the concept lattice, most only examined the lists of concepts generated. These studies essentially reduced FCA's effectiveness to that of a clustering tool's, instead of using its full hierarchical capabilities. More recent studies investigate more complex variations on the initial problem, as well as disease similarity and the makeup of genes in certain disease subtypes. These areas are promising and may be further developed to make full use of of what FCA has to offer.

In addition to examining these studies and their effectiveness, a more important question is whether these studies using gene expression data should make use of FCA in the first place. What benefit does FCA-based clustering provide the genomics field that other clustering methods cannot? If the concept lattice is not used, examination of whether FCA still provides benefit is a worthy research question. If FCA is beneficial, discovering other fields of benefit is also necessary.

Several other areas where FCA may provide benefit should be examined as well. For example, classification is a similar problem to clustering, and is a common technique in data mining that FCA can be used for. Once the concept lattice is generated, predicting which concepts new data will be added to is an interesting problem, especially when combined with fuzzy data or other extensions of FCA [43].

Another area of possible worth is the framework of FCA as a bipartite graph. Some investigation into this framework has been done [39], but additional work in this field, especially with pairing FCA and graph algorithms (such as in [25]), may lead to interesting results for FCA as a theoretical framework and as a graph theoretic model.

The contributions of this thesis are as follows:

- We survey the literature that uses Formal Concept Analysis to study bioinformatics, particularly gene expression data, providing a study in depth and breadth of FCA's benefits and drawbacks to the field. In addition, we remark on FCA's feasibility for the field. A similar survey studying how FCA is used to discover knowledge from biological data, including gene expression data, was done in 2017 [78]. We expand on that portion of the work by providing a robust explanation of FCA and various data management strategies, as well as examining the feasibility of using FCA for such an application. This portion of the thesis is currently under preparation for submission to the ACM Computing Surveys journal.

- We identify areas of research in the intersection of Formal Concept Analysis and gene expression data.
- We define two concrete problems for future research for Formal Concept Analysis which can lead to our PhD dissertation.

The rest of this thesis is organized as follows. First, in Chapter 2, a basic biological background of gene expression data is provided. We provide insight into how FCA works, some variations of FCA, and methods of managing data in Chapter 3. Next, we examine how FCA has been used to analyze gene expression data in Chapter 4. Finally, in Chapter 5 we identify areas of future research into FCA and conclude.

Chapter 2

Biological Significance of Gene Expression Data

In this chapter we give an overview of the biological motivation for gene expression data.

A living organism can contain trillions of cells, which are a basic unit of life. Cells must perform certain functions to stay alive. Those functions are often completed with the help of molecules called proteins, that carry out instructions provided by the cell. The presence or absence of a protein can be measured to determine both how a cell performs basic functions and how a cell reacts to its environment. In order to measure this, we must first look at different molecules in the cell. To begin, the nucleus is a membrane-enclosed molecule that holds chromosomes, each of which include a string of deoxyribonucleic acid (DNA). DNA is where the organism's genes are stored, which are instructions for cell functions.

DNA is constructed with nucleotides, which are molecules including adenine (A), cytosine (C), guanine (G), thymine (T), and uracil (U). Genes are constructed with four out of these five nucleotides (A, C, G, and T), and various combinations of the four can result in millions of different genes, depending on the organism.

DNA is similar to another type of molecule, ribonucleic acid (RNA). Both RNA and DNA are constructed of nucleotides (RNA uses A, C, G, and U[1]). DNA is an

[1] Note that for reasons outside the scope of this paper, only DNA contains thymine and only

intersecting double helix of nucleotides, while RNA is only a single strand, and is obtained by copying sections of DNA. Although there are many different types and functions of RNA, we focus on only three main types for this explanation: messenger RNA, transfer RNA, and ribosomal RNA. Messenger RNA, or mRNA, contains a copied gene from DNA that serves as the instructions for the future process. Transfer RNA (tRNA) is a folded strand of RNA that is bonded with an amino acid. Ribosomal RNA (rRNA) is part of the ribosome (a structure outside the nucleus) that helps synthesize proteins.

We now look at how protein (and a sequence of proteins, called a protein chain), is constructed, which will allow us to discuss cell reactions to the environment. The first step is transcription. In this step, DNA is copied onto a strand of mRNA within the nucleus. The second step is translation. In this step, the contents of the mRNA interact with the other types of RNA to produce a protein chain. The mRNA polymerase moves down a strand of DNA, copying the contents of a gene onto the mRNA. Once the copying has concluded, the mRNA passes out of the nucleus.

The genes that the mRNA transcribes encode functions in the cell [2]. The levels of mRNA can be determined with the Northern blot method [59, 71]. Our aim is to analyze the mRNA levels.

Experiments relating to this protein production involves cells undergoing situations in a laboratory, and scientists measuring the resultant protein output. Recent technologies allow for many experiments to be conducted at once in a microarray, which contains thousands of cells, each containing a single gene. From this type of experiment, the expression level, or amount of protein produced, of each gene is obtained, usually as a numerical floating-point value. The resultant matrix of genes tested versus situations tested in is called the gene expresion matrix (GEM), and the

RNA contains uracil; if RNA encounters thymine it is treated as uracil.

data in the matrix is called gene expression data (GED).

Chapter 3

Formal Concept Analysis and Data Management

To see how analysis of gene expression data is done with Formal Concept Analysis (FCA), we must first address its basic definitions. The following section enumerates the definitions, describes several variations on the traditional structure, and addresses how to manage incidents of high-complexity data.

3.1 Examples and Definitions

The input to Formal Concept Analysis is a formal context, a theoretical structure similar to a matrix. The rows of the context represent objects, and the columns represent attributes. A True (represented by an "X" in our examples) in a certain row and column of the context indicates the corresponding object has the corresponding attribute. Conversely, a False (respectively, the absence of an "X") indicates the corresponding object does not possess the corresponding attribute. Note that objects and attributes may be purely abstract.

Formally, the context is a triple $\mathbb{K} = (G, M, I)$ where G is the set of objects, M is the set of attributes, and I is the binary relation between the object and attribute sets. One simple example of a context, $\mathbb{K}_1$, is shown in Table 3.1. Various animals chosen (Dog, Cat, Fish, Parrot, Duck, and Snake) may or may not possess the various attributes (having Tail, Whiskers, Fur, Feathers, Scales, or Fins; able to Swim; making Noise). A naive observation of a collection of these animals results in data that indicates all the animals have a Tail; only Dog and Cat have Whiskers and Fur; Parrot and Duck have Feathers; Snake and Fish have Scales; Fish is the only animal with Fins; Dog, Fish, Duck, and Snake can Swim; and all except for Fish make Noise. This data is stored in the matrix representing the relation I between G and M. Later, in Section 3.2, we will address the concern that these attributes might not be true for all individuals of these species.

Table 3.1: Animals and their characteristics represented in a binary FCA context called $\mathbb{K}_1$.

$\mathbb{K}_1$	Tail	Whiskers	Fur	Feathers	Scales	Fins	Swims	Noise
Dog	X	X	X				X	X
Cat	X	X	X					X
Fish	X				X	X	X	
Parrot	X			X				X
Duck	X			X			X	X
Snake	X				X		X	X

Once the context is constructed, we must discover implicit information in the data by looking at its structure and hierarchy. This is achieved with a formal concept, referred to as a “concept”, for which FCA is named. The formal concept is a subset of the matrix with “X”s in all locations, according to some constraints addressed in Definitions 1 and 2.

Definition 1 (Intent). *For a set $A \subseteq G$ of objects, the* intent *of A is denoted by A', where*

$$A' := \{m \in M | \textit{ every object } g \textit{ in } A \textit{ possesses every attribute } m\}.$$

Alternatively, this is the corresponding set of attributes that all objects in A possess.

Example 1. *As a simple example, for the set $A = \{\textit{Dog, Cat}\}$, the set A' is $\{\textit{Tail, Whiskers, Fur, Noise}\}$.*

Definition 2 (Extent)**.** *For a set $B \subseteq M$ of attributes, the* extent *of B is denoted by B', where*

$$B' := \{g \in G |\ \textit{every attribute } m \textit{ in } B \textit{ is possessed by object } g\}.$$

Alternatively, this is the corresponding set of objects that possess all the attributes in B.

Example 2. *For the set $B = \{\textit{Swims, Noise}\}$, the set B' is $\{\textit{Dog, Duck, Snake}\}$.*

We can also obtain A'' from Example 1, which is $A'' = \{\text{Dog, Cat}\}$. Similarly, B'' from Example 2 is $\{\text{Tail, Swims, Noise}\}$.

Definition 3 (Formal Concept)**.** *A* formal concept *is a pair (A, B) of objects and attributes, respectively, such that $A' = B$ and $B' = A$. A is called the extent of the concept and B is called the intent of the concept.*

Put another way, a concept is simply a pair of objects and attributes (A, B) such that every object in A possesses all attributes in B, and every attribute in B is possessed by all objects in A. Due to this definition, (A, A') from Example 1 is a concept because $A' = A'$ trivially and $A'' = A$. However, (B', B) from Example 2 is not a concept because $B'' \neq B$. We define the set of all concepts of a context as $\mathfrak{B}(G, M, I)$.

With these definitions, another example concept is shown in Table 3.2. This concept is constructed by choosing the set A={Parrot, Duck}, then obtaining all attributes that satisfy A'. These attributes are B = {Tail, Feathers, Noise}. Even though Noise and Tail are both attributes that belong to objects outside the set A, the inclusion of the attribute Feathers in B restricts the object set A to only Parrot and Duck, since both objects are the only to share the attribute Feathers.

Table 3.2: A sample concept from the context $\mathbb{K}_1$ in Table 3.1. Notice that for the objects Parrot and Duck, all attributes in common to both objects (Tail, Feathers, Noise) are included in the concept.

	Tail	Feathers	Noise
Parrot	X	X	X
Duck	X	X	X

Because the concepts are specific subsets of the original context, some concepts may depend on each other. We demonstrate this with another valid concept in Table 3.3, and Definition 4.

Table 3.3: A concept (A, B) where A = {Cat, Parrot, Duck, Snake} and B = {Tail, Noise}.

	Tail	Noise
Cat	X	X
Parrot	X	X
Duck	X	X
Snake	X	X

Definition 4 (Subconcept, Superconcept)**.** *Given two concepts* $C_1 = (A_1, B_1)$ *and* $C_2 = (A_2, B_2)$*,* C_1 *is a* subconcept *of* C_2 *(denoted* $C_1 \leq C_2$*) iff* $A_1 \subseteq A_2$ *or* $B_1 \supseteq B_2$*. Equivalently,* C_2 *is said to be a* superconcept *of* C_1 *and* $C_2 \geq C_1$*.*

It is simple to see that every context will have two trivial concepts: the first has a potentially empty extent and an intent containing all attributes in the context. The second has an extent containing all objects in the context, and a potentially empty intent. We consider these potentially empty sets as valid parts of their concepts due to a possible coincidence that all objects share one or more attributes (as in the case of the attribute Tail in $\mathbb{K}_1$) or all attributes being possessed by one or more objects. For ease of repetition, we will call the all-object concept C_G and the all-attribute concept C_M.

FCA was created with lattice theory, an existing field of modern algebra, to reconnect abstract mathematics to the modeling of real-world situations and data. We now present some key definitions of lattice theory, beginning with the join and meet of a set in Definitions 5 and 6, then discuss how the definitions function in FCA.

Definition 5 (Join)**.** *A set S with a partial order $\leq$ has a* join *iff there is an element $c \in S$ such that $c \geq x$ for every $x \in S$. c is also called the supremum or least upper bound of S.*

Definition 6 (Meet)**.** *A set S with a partial order $\leq$ has a* meet *iff there is an element $c \in S$ such that $c \leq x$ for every $x \in S$. c is also called the infimum or greatest upper bound of S.*

The set of all concepts, $\mathfrak{B}(G, M, I)$, as well as its subsets, fulfills a pair of strict definitions involving the join and meet of its elements.

Definition 7 (Lattice)**.** *A lattice is a set S such that there is a join and meet for every pair of elements of S.*

Since pairs of concepts in $\mathfrak{B}(G, M, I)$ have at least the general supremum C_G and the general infimum C_M, $\mathfrak{B}(G, M, I)$ is a lattice. In addition, $\mathfrak{B}(G, M, I)$ has a structural property that every one of its subsets has a join and meet. The meaning of this in lattice theory is defined in Definition 8 and its implications are discussed in the following paragraphs.

Definition 8 (Complete Lattice)**.** *A complete lattice S is a lattice such that every subset of S has both a join and a meet.*

The term for the lattice of concepts ordered by the subconcept-superconcept relation is defined in Definition 9.

Definition 9 (Concept Lattice)**.** *The set of all concepts, ordered by $\leq$, is called the* concept lattice *and is denoted by $\underline{\mathfrak{B}}(G, M, I)$.*

The concept lattice $\underline{\mathfrak{B}}(G, M, I)$ is a complete lattice, the rigorous proof of which is out of the scope of this paper; however, it can be found in Theorem 3 in Ganter and Wille's book [38]. Instead, we offer a general understanding of the theorem. If $\underline{\mathfrak{B}}(G, M, I)$ was not complete, there would be some set $Y \subseteq \mathfrak{B}(G, M, I)$ for which the join or meet would not exist. Y could not be C_G or C_M, since these two concepts, always present in the concept lattice, are the join and meet of themselves: C_G is the join of itself and C_M, while C_M is the meet of itself and C_G. It remains for Y to be a singleton concept in the lattice. However, since C_G and C_M are the respective join and meet of the entire lattice, it is impossible for Y to be a singleton. This means that because $\underline{\mathfrak{B}}(G, M, I)$ is a complete lattice, it must be *connected* in a graph theoretic sense. In this way, although the data may be disjoint (concepts apart from C_G and C_M are not sub or superconcepts of each other), it will not be disconnected, and can therefore be analyzed accordingly.

A visualization of $\mathbb{K}_1$, made with LatViz [6], is shown in Figure 3.1. In this type of diagram, there are three main parts for interpretation: vertices, edges, and labels. Circles are concepts; lines represent subconcept-superconcept ordering, with a subconcept represented below the superconcept; labeling enables the reader to see which objects share certain attributes in a way that will be explained once the entire diagram is constructed. Even though one concept may be below another, physical distance is of no relevance in the diagram. We now present a more detailed example of how the concept lattice is constructed and how it may be interpreted in Example 3.

Figure 3.1: The concept lattice for context $\mathbb{K}_1$.

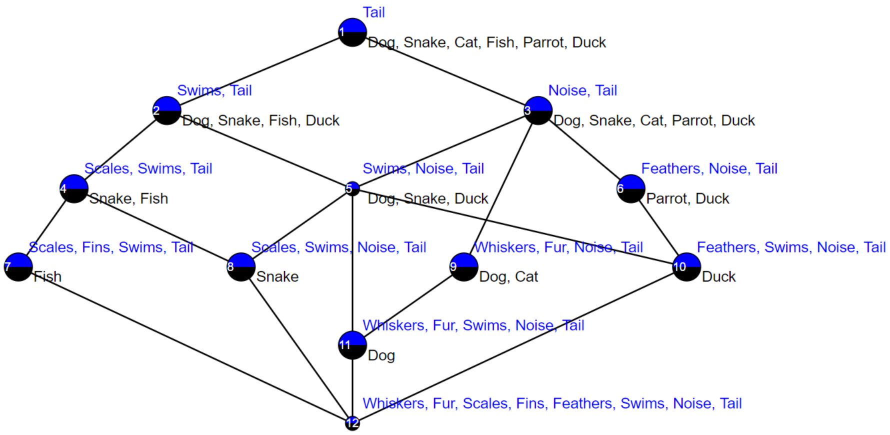

Tail
1
Dog, Snake, Cat, Fish, Parrot, Duck
Swims, Tail
2
Dog, Snake, Fish, Duck
Noise, Tail
3
Dog, Snake, Cat, Parrot, Duck
Scales, Swims, Tail
4
Snake, Fish
Swims, Noise, Tail
5
Dog, Snake, Duck
Feathers, Noise, Tail
6
Parrot, Duck
Scales, Fins, Swims, Tail
7
Fish
Scales, Swims, Noise, Tail
8
Snake
Whiskers, Fur, Noise, Tail
9
Dog, Cat
Feathers, Swims, Noise, Tail
10
Duck
Whiskers, Fur, Swims, Noise, Tail
11
Dog
Whiskers, Fur, Scales, Fins, Feathers, Swims, Noise, Tail
12

Table 3.4: A simplified context.

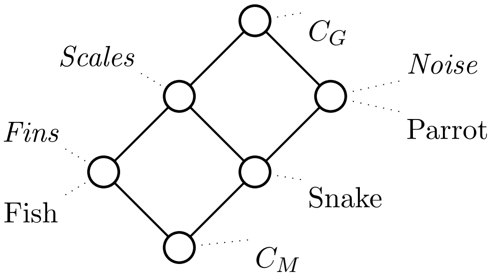
$\mathbb{K}_2$	Noise	Scales	Fins
Fish		X	X
Snake	X	X	
Parrot	X		

Example 3. *Taking a subset of data to be $\mathbb{K}_2$ (Table 3.4) from the original context $\mathbb{K}_1$, we compute 6 concepts: C_M with an empty object set, the set of all attributes the animal Snake possesses (C_2); the set of all attributes the animal Fish possesses (C_3); the set of all objects possessing the attribute Scales (C_4); the set of all objects possessing the attribute Noise (C_5); and C_G with an empty attribute set. According to our definitions, no other concepts apart from these six are possible, as it would*

involve one of the object or attribute sets not equaling the equivalent intent or extent, respectively.

Now that we have computed all the concepts, we determine if some are subconcepts or superconcepts of each other. By examining the corresponding object and attribute sets, we find $C_M \leq C_G$*, as expected. Additionally,* $C_2 \leq C_4, C_5$*, and* C_G*, while* $C_3 \leq C_4$ *and* C_G*. We find that* C_4 *and* C_5 *have no relation between each other because neither has a set that is a subset of the other's.* C_2 *and* C_3 *are similarly disjoint. All six concepts are superconcepts of* C_M*. All six concepts are subconcepts of* C_G*. Drawing these dependencies according to the previously mentioned specifications, we obtain the concept lattice (Figure 3.2).*

Figure 3.2: Concept lattice for the context X_2 from Table 3.4.



We label the objects on the concept lattice by first working bottom-up [37]. We label the concept with an object if it is the object's first appearance in the lattice is in that concept. Similarly, work top-down to label (in italics) the first occurrence of each attribute. When fully labeled, this allows us to work "up" the concept lattice to find an object, then see, by moving only to superconcepts of the concept where the object first appears, the attributes that object possesses. Similarly, if an attribute is found, moving only to subconcepts, we can see the objects that possess that attribute. In this way, we can see that fins are only possessed by fish in our simplified context, and that *a parrot's only attribute is that it makes noise. Thus the concept lattice allows for a more intuitive method of examining the data, rather than looking solely at the binary matrix.*

3.2 Variations of FCA

Although FCA was developed to use a binary context, most real-life applications may not use binary data. For example, gene expression data often uses floating-point values to represent protein concentration as a result of how genes react in different situations. There are several options to analyze such data with FCA. Either the multi-valued data can be binarized to 0 and 1, remain as multi-valued data and perhaps be used as fuzzy or probabilistic data, or be represented in pattern structures. Since application-specific binarization (discretizing data to binary values) methods will be discussed in Section 4.2.1, we will briefly describe the latter two options here.

To motivate Fuzzy FCA, we observe that different breeds of cats in the context $\mathbb{K}_1$ may have differing amounts of fur. Some breeds (Turkish Angora, British Longhair) have long fur, and others (Sphynx, Peterbald) are hairless or almost so. Other cats (Siamese, Tabby) can fall between the two extremes. To represent this variation accurately in our context, one option is to expand the "Fur" attribute into three: Fur-Long, Fur-Short, and Fur-None. We may do a similar refinement process for the attribute "Noise", since each of the animals in $\mathbb{K}_1$ make different noises. We may expand Noise to: Noise-Bark, Noise-Growl, Noise-Mew, Noise-Hiss, Noise-Squawk, and Noise-Quack. While this strategy may help clearly refine our small context and eliminate possible confusion of varying breeds, it is not necessarily scalable for the whole data set. For contexts with a high number of dimensions, this process could lead to an explosion of binary attributes, many of which might not be relevant for most objects that did not possess the attribute in the first place. This method of expanding attributes to suit the data is known as scaling, and will be further addressed in Section 4.2.1. As an alternative, Fuzzy FCA is sometimes used to accommodate variation in an object's possession of an attribute. A user-friendly explanation of Fuzzy FCA is detailed in [65], a small summary of which is given here.

Instead of a binary value, fuzzy contexts' attribute values are real-valued numbers, usually in the range of $[0, 1]$, where the attribute value represents the degree of attribute in the object (e.g. amount of fur on a cat). A fuzzy subset of a context is made up of fuzzy object and attribute sets, each one specifying how relevant it is to specify the element it contains. This is often represented as a threshold to determine an object or attribute's inclusion. To generate concepts, a residuum, or fuzzy implication, determines whether the threshold is met for each object and attribute. From here the concept lattice, a residuated lattice, can be constructed.

To avoid the explosion of number of attributes, pattern structures [36] may be used, which can capture information about numerical attributes without need for scaling. Pattern structures can also be connected with Fuzzy FCA [24]. A few authors have analyzed gene expression data with FCA and pattern structures [45, 51, 55, 14, 49, 53, 56]. This developing field is a promising alternative to scaling, and should be further explored to assess its user-friendliness and scalability.

Another way FCA can be applied is in probabilistic applications [50], the theory of which is based on rough sets. Unlike fuzzy sets, which involve the degree of an attribute manifesting in an object, rough sets are used to extract knowledge from missing or incomplete data by estimating the missing or incomplete set's boundary regions. These regions are computed to attempt to determine the certainty of an object possessing a given attribute. Rough set theory can also help reduce data redundancy by treating objects which possess the same attributes as indiscernible and thus identical. For example, if the pair (Cat, Swims) was present in the context $\mathbb{K}_1$, the objects Cat and Dog would be indiscernible for the purposes of rough sets, and the objects could be merged to create a reduced data set called the reduct. FCA can also be used with rough set theory to generate probabilistic association rules [88] and further discover maximal potentially useful rules in databases [30, 29].

Use of these variations in analysis of gene expression data is becoming more popular, but still lacks a widespread implementation.

3.3 Reducing Concept Lattice

Whether using a variation of FCA or not, a common issue in data analysis is that with larger datasets, the number of concepts greatly increases, as does the size of the lattice. This leads to a cluttered visualization and decreasing intuition unless preventative steps are taken to reduce the concept lattice into an interpretable structure. These steps may include algorithms to generate (select) only specific concepts, to simplify the context by removing redundant objects and/or attributes, or to simplify the concept lattice itself by eliminating concepts that capture information already represented in the lattice.

A comprehensive survey of this topic was written in 2015 by [32], which addresses the categories of selection, removing redundant information, and simplifying the lattice. We mention here strategies of lattice reduction in these categories that have emerged after the survey's publication or were not mentioned.

Selection of concepts is done while computing the initial list of all concepts. Algorithms to do so have been developed, with varying improvements, including computation requirements, threshold of concept inclusion, and incremental construction of the concept lattice. ParallelGenerate and FCbO [60, 61] require no synchronization when computing concepts. In-Close2 [10] allows for an input of minimum support threshold for a concept's object and attribute inclusion, which reduces the overall number of concepts and potentially allows more "significant" concepts to be included. AddIntent [86] constructs the concept lattice diagram incrementally, rather than computing all concepts and expensively building the diagram from scratch.

The other main methods of selecting concepts include functional dependencies [52, 67], an expression of FCA's attribute implications. However, functional dependencies are often defined over the numerical and categorical attributes, while in FCA, implications are defined over binary attributes. An approach for classical FCA was developed [64] and is quadratic for the number of objects in the context, thus not scalable for large data sets. To overcome this limitation, authors in [16] present an approach for transformation by introducing partition pattern structures to apply binarization over attributes rather than objects. In partition pattern structures, the dataset is partitioned on attributes with sets over objects with same value of attributes. This results in reduced concept lattice size but with equivalent results that may be scalable for large data sets.

Methods to remove objects and attributes from the context include algorithms to change specificity of binary data, locally reduce attributes instead of globally, and theoretical methods to find the reduct of the context. The algorithms Fold [89] and Unfold [93] are used to increase granularity (fineness) of attributes. This is a significant step in binarization of a multi-valued matrix to a formal context, but the resulting set of attributes impacts the total number of concepts, thus meriting inclusion in this discussion. Another method to reduce attributes is to use decision rules to determine local attribute reductions [77], to avoid loss of information locally. This method can help improve three-way classification. Other theoretic methods include [25], which proposes method for constructing simplified discernibility matrix which does not require to generate all formal contexts. Further, a fast heuristic algorithm is designed based on graph theory to greedily find the vertex cover corresponding to the attribute set, and so obtain the reduct of the formal decision context. Using the Min-Ex algorithm [72], intrinsically noisy data can be reduced by mining association rules, then extracting sets of objects that are more significant than the naive explosion of concepts through traditional means. This produces a so-called "fault-tolerant" FCA, with concepts bounded by a number of dimensions.

To simplify the lattice by choosing representative concepts from the total list of concepts, attribute clustering, theoretical extensions of FCA, and minimal generators are used. Fuzzy attribute clustering [63] allows for more flexibility in multi-valued data, while clustering attributes with the Jaccard similarity [84] can be completed with binary data.

Theoretical extensions of FCA can also assist in choosing representative concepts. Authors in [16] demonstrated that increasing uncertainty in formal fuzzy concept analysis leads to reduction in size of the concept lattice. In their previous work [15], they presented an attribute-oriented concept lattice where attributes were derived as positive and negative information from the given data. In the present work [16] they show that the uncertainty is naturally modelled in Fuzzy FCA which, in turn, naturally leads to lattice size reduction.

A fairly common approach is to identify the minimal combination of objects or attributes (called minimal generators) that distinguish the objects of one concept from other. In 2002, an incremental algorithm to mine minimal generators [73] was proposed. However, the minimal generators may still contain redundant attributes. Therefore, in 2005, a new algorithm was proposed to remove the redundancy [34]. A depth first search approach is followed to build a depth first tree representing all attribute set for a given set of attributes. Then, useless branches are removed from the tree. Their approach is experimentally validated on two real test datasets, UCI Mushroom and colon tumor gene expression [5]. The results shows that the succinct approach can deal with high dimensional and large real datasets.

Each of these methods can assist the user in generating an interpretable concept lattice, thus aiding in analysis. This concludes our discussion of how FCA works and FCA-specific data management strategies.

Chapter 4

Analyzing Bioinformatics Data with Formal Concept Analysis

4.1 General Bioinformatics Applications

Formal Concept Analysis has been used as an approach for knowledge discovery in various bioinformatics fields. A comprehensive survey of use of FCA for knowledge discovery was done by [74].

FCA being used in bioinformatics areas other than gene expression data include [44], which analyzes metabolomic data with FCA as a key step to find patterns in the database. Another area concept analysis has been used for is finding ecological traits in hydrobiological data [20]. Fuzzy FCA is used, and scaling of the multi-valued data is done with histogram scaling. Phylogeny has been approached as a possible area of interest in FCA, especially with the concept lattice as a visual structure to analyze evolutionary data. An attempt at using phylogeny with FCA for information-retrieval and conceptual clustering was done by [68]. Evolutionary analysis with FCA was also attempted by [76], who compared the concept lattice to median networks. However, it is shown in [41] that FCA is not equivalent to a median graph, and therefore the area requires more formalism and caution before FCA can be applied to phylogeny.

The survey [74] also cites areas FCA has been used for in biochemistry and medicine. These areas include applications such as predicting the biological activity in chemical compounds [23], identifying similar test results [81], and examining drug reactions [80]. The majority of FCA's effect in bioinformatics has been in analyzing gene expression data.

4.2 Gene Expression Data Analysis

The structure of FCA, specifically its concept generation from data, lends itself to to generating groups of genes that are expressed similarly. These groups are commonly called synexpression groups or co-regulated genes. In this section, we first examine how gene expression data can be discretized to work with classical FCA (Section 4.2.1). Then, we discuss the various studies that use FCA (Section 4.2.2), beginning with synexpression groups and surveying other developments topically, including disease applications in Section 4.2.3. Finally, We conclude with a discussion of the feasibility of such applications in Section 4.3. A brief summary of the methods surveyed and their purposes can be found in Table 4.1.

Table 4.1: A summary of authors' purposes for using FCA to analyze gene expression data.

Category Regarding	**Papers**
Feasibility Studies	[79], [49], [75], [26]
Transcription Factors	[21]
Multiple Experiments	[48]
Synexpression Groups	[22], [57],[53],[56]
Gene networks/Negative Synexpression	[47], [85],[40]
Biological Functions	[11], [12]
Diseases	[62], [58], [7], [8], [9], [69]

4.2.1 Discretization of Gene Expression Matrices

Before examining the applications in detail, how the gene expression data is discretized must be discussed. Several common techniques are used to discretize and/or binarize (discretize to 2 values) gene expression data data. These will be addressed in full in the applications, but we briefly explain them here. The easiest method of binarization is measuring whether a gene is expressed or not (E/NE); this is determined by assigning 1 if the value exceeds a threshold for each gene [79]. This method is also used to capture whether a gene is overexpressed in a situation [72, 22]. Strong expression, whether positive or negative, can be captured is a similar method [21]. A variant of this is first transforming the matrix to a 3-valued matrix (often -1, 0, 1) based on each row's value varying from the average. From there, the matrix is discretized to a binary matrix by doing a similar computation [46, 85, 47].

Some of these methods, especially overexpression or E/NE, risk losing the strength of expression by setting a threshold. To retain a measure of expression strength, interordinal scaling is used [75, 57]. Using binary attributes of one-sided intervals to partition the multi-valued data was proposed [75]. However, if an attribute exceeds a certain threshold, then the object will possess all the relevant attributes, not just one. A following study proposed two-sided intervals (called interordinal scaling) to address this issue [57]. The intervals chosen in interordinal scaling correspond to a scale proposed by [26] and [70]. Measuring the similarity of concepts discretized with interordinal scaling was done by [35]. Interordinal scaling can also be combined with pattern structures [53].

Another way to binarize is to use cluster membership when clusters of genes are computed as a preprocessing step [48]. Groups of genes can be also be used using Rough Set Theory, where all permutations of objects are paired together [82, 83]. The attribute for the new paired object is measured by how the second attribute's value compares to (by $<$ or $\geq$) the first. Classification results shows reasonable results. However, this method results in quadratic complexity, as discussed in [14], so using a similar process with pattern structures is proposed instead, as it is more scalable and feasible for the lattice construction. However, this method has been used to analyze the effect of discretization, and not on the biological impact of such calculations.

Whatever method of discretization is used, it must be directly related to the purpose of the study. As discretization has inherent data loss, care must be taken as to which method is used.

4.2.2 Applications of Gene Expression Data

We now address how gene expression data is analyzed with FCA. Two authors initially examine the feasibility of clustering gene expression data with similar methods to FCA. In a 2003 study, [79] examines Galois concepts in discovering co-regulated genes. Results of different discretization methods are also examined, which shows insight into its importance. A 2004 survey explored the feasibility of clustering gene expression data [49]. Although Formal Concept Analysis is not explicitly mentioned, the survey describes how clustering (including biclustering, a similar method to FCA) can be used to gain information from a microarray gene expression matrix . In particular, clustering methods specifically used for gene expression data as well as class validation and the reliability of produced clusters are discussed. Other early attempts to do gene expression analysis with FCA include finding transcription factors, proteins that regulate whether a gene is expressed or not. Using the algorithm D-Miner, formal concepts with Galois operators are mined [21]. The concepts then reveal the transcription factors.

Potter, in his doctoral thesis, is one of the first to use FCA with microarray data. In his algorithm microBLAST, Potter uses interordinal scaling as one way to discretize the gene expression matrix, which is done by comparing the expression level of each gene to an inequality. If a gene does fulfill an inequality, it is said to possess that attribute. An example of this transformation is shown below in Tables 4.2 and 4.3. One potential issue is that if an attribute exceeds a certain threshold, then the gene will possess all the relevant attributes, not just the closest one. This issue is addressed in [57]. Potter then creates the microBLAST lattice from the biological lattice, which is simply the concept lattice of the discretized biological data. No mention is made of synexpression groups or other specific applications to gene expression data.

Table 4.2: Original sample gene expression matrix.

	s_1	s_2
g_1	5.3	0.7
g_2	3.36	1.2
g_3	2.5	0.25
g_4	0.43	0.96

Table 4.3: Gene expression matrix discretized with interordinal scaling.

	$s_1, \geq 1$	$s_1, \geq 2$	$s_1, \geq 3$	$s_1, \geq 4$	$s_2, \geq 0.5$	$s_2, \geq 0.75$	$s_2, \geq 1$
g_1	X	X	X	X	X		
g_2	X	X	X		X	X	X
g_3	X	X					
g_4					X	X	

Potter's seminal work is then implemented in Choi et al [26]. The authors approach the problem of binarization in a method similar to interordinal scaling, by dividing the number line according to the “gaps” in expressed numerical values. Then recursively partition the intervals into subintervals until a desired partitioning is reached. This allows any given gene in the context to possess the one-valued attribute of being in its interval. Concept lattices are generated for each separate experiment, and then compared according to a defined distance. The distance Choi et al propose is a measure of which genes each vertex (concept) in the lattice shares with another. This allows for comparison and analysis of different experiments with mouse lung tissue gene expression data.

Using multiple experiment data is further explored in [48], which proposes multiple experiments can be beneficial to analysis because the risk of biases in individual experiments is reduced, and combining data can result in a higher confidence of results. Binarization is done by first partitioning then consensus clustering the genes. Then, the context objects are genes while the attributes are whether each gene belongs to each cluster. Analysis is done with a time series dataset of fission yeast, and the biological significance of these results are discussed.

Identifying the synexpression groups of a dataset is the explicit purpose of [22]. Discretization is done by overexpression: if a gene's expression level in some situation is over a set threshold, the relationship is 1, 0 otherwise. The extraction of concepts is done with D-Miner [21], an algorithm designed to extract transcription factors, which works through local pattern discovery. The intent of these extracted concepts are then examined to find the synexpression groups.

The effects of various binarization techniques on discovered synexpression groups are explored by [57, 53, 56]. In the first study [57], a binarization method called *conceptual scaling*, which converts the value of each expression level to an interval, is proposed. If an expression level does correspond to an interval, it is said to possess that attribute. This is a better method than Potter's interordinal scaling, because if two expression levels fulfill a certain inequality, there is a better chance that attributes do not overlap. An example of Table 4.2 converted to binary data with conceptual scaling is shown in Table 4.4. A potential downside to this method is that the number and size of intervals should be determined by an expert to avoid losing too much precision.

Table 4.4: Gene expression matrix discretized with conceptual scaling proposed by Kaytoue-Uberall et al.

	$s_1, (4, \infty)$	$s_1, (3, 3.99)$	$s_1, (2, 2, 99)$	$s_1, (0, 0.99)$	$s_2, (1, 1.99)$	$s_2, (0.5, 0.99)$	$s_2, (0, 0.499)$
g_1	X					X	
g_2		X			X		
g_3			X				X
g_4				X		X	

In the second study, the effects of binarization on algorithms Norris, CbO, and NextClosure are examined. In particular, two methods of adapting real-valued data to a form FCA can use are inspected. These two methods, interordinal scaling and pattern structures, are used to reduce the size and complexity of the concept lattice in experiments with *Laccaria bicolor*, a fungus that is found on tree roots [53]. With data in a gene expression matrix, the numerical data is represented as as attributes in terms of scales. For example, if a gene g_i has value w_j in situation s_k, the attribute for g_i is represented as the one-valued attribute $s_k \leq w_j$. This is done for all values each gene may have in any situation, which accordingly leads to a very large context. To restrict attributes, the inequalities may be restricted (i.e., not included in the context) to satisfy some maximum value. As this method can result in an explosion of attributes, interval pattern structures are also explored as an alternative to binarization, reducing the risk of data loss. The experiments on the tree fungus dataset show the role and function of genes that are expressed similarly. Concept extraction with interval pattern structures is further explored in [56]. By modifying FCA to allow for pattern structures, concept extraction from such a context can show the gene synexpression.

An algorithm BiFCA+ was developed [46] to cluster gene expression data into biclusters based on FCA, which corresponds to synexpression groups. The algorithm is based on biclustering data that is binarized to a three-state matrix based on expression variance in conditions before transforming into a two-state matrix based on the average value per gene. The set of biclusters are reduced by the Bond correlation if there is significant overlap between samples.

A variation on finding synexpression groups is done by [40], who identifies candidate genes to include in the gene regulatory network, a group of genes that interact to perform and regulate cell function. Binarization is done by matching a gene to a template of a statistical measure. The method is tested on a tuberculosis dataset. While this application is not exactly the same as other synexpression experiments, it is related because some genes may interact positively (expression behaves similarly for both genes in a situation) or negatively (while one gene's expression increases, another's decreases). This negative interaction is also called negatively correlated genes.

Negative correlation between genes is examined in [85] and [47]. In [85], the algorithm NCFCA find the negatively correlated genes in cell cycle time series data. Discretization of the matrix is done by transforming the expression value to 1, 0, and -1 if the value in the time series data increased, not changed, or decreased, respectively. Then the matrix is binarized based on the row's average value deviation. The concept lattice is filtered by limiting the minimum number of genes in each subset.

Another algorithm, NBF, was proposed to find negative correlations, which the author refers to by negative biclusters [47]. Discretization is similar to [85], except when going from the 3-state to the 2-state matrix, two binary matrices are produced, one representing a positive average change per row, and another representing a negative change. Experimental results are obtained from yeast cell-cycle, human b-cell lymphoma, and Alzheimer gene expression databases.

Negative gene synexpression is an interesting application with FCA and should be explored further.

Using FCA to extract which genes perform biological functions is a natural extension of determining positive or negative synexpression groups, as the functions generally involve groups of genes expressed in similar situations or negatively correlated. Using a dataset of genes involved in colorectal cancer (context objects) and the expression profiles, gene ontology terms, and knowledge bases (context attributes), the concepts produced are then analyzed with enrichment analysis to determine the biological reasons the genes are grouped together in the concepts [19].

In order to select biclusters of genes that work together to perform biological functions, a variation on FCA, $\mathcal{K}$-FCA is developed [42], along with a web tool WebGeneKFCA [18] that provides a variety of visualization techniques to aid the user in determining gene under- or over-expression, as well as provide an interface with gene ontology. $\mathcal{K}$-FCA provides an analysis method similar to Fuzzy FCA that reduces the need to binarize the entire dataset. Lattice reduction is also addressed; a proposed solution is to "filter" concepts by judging if an intent contains too many or too few genes, according to some threshold which is again determined by an expert. This allows the lattice to be somewhat simplified, and allows the "critical" concepts to be generated and displayed. Further research in this area could attempt to reconcile Fuzzy FCA with $\mathcal{K}$-FCA.

Specifically using developing mouse embryo tissue gene expression data like [26], the makeup of genes in components of tissues is found [10, 11, 12]. The algorithm In-Close 2 was developed [10] and tested on the mouse tissue data to see which concepts were detected. Binarization in all three studies was done by allowing any expression level in a gene. The components of mouse tissue were visualized with the acquired knowledge of the genes expressed in each component. These studies could be further explored with a method of binarization that does not reduce the strength of each gene's expression.

4.2.3 Disease Applications

FCA can also be used to determine disease similarity [58], as well as discover which genes correspond to specific subtypes of diseases [69, 8, 9, 7].

Using FCA to test for disease similarity is done by [58]. Of particular interest is the graph theory formulation of FCA, which uses subgraphs and spanning trees to discover concepts. Analyzing the data of up- or down-regulated genes in renal disease biopsy tissue, the concept lattice is inspected to determine the relationship between the various renal diseases.

Determining a set of genes that can distinguish between two classes of a disease is done by [69]. Specifically, the goal is to distinguish which genes are associated with tumor-based breast cancer, and which are associated with metastatic breast cancer. Discretization is done by setting a threshold for each gene and having one attribute represent the experimental value below or equal to the threshold, and another represent above. Noise is reduced by setting a minimum and maximum number of genes allowed in an intent.

Another group of studies that examines the genetic data of diseases with FCA is [8, 9, 7]. FCA is used to find genes that have inhibited [8] or uninhibited [9] expression in breast cancer subtypes. Given a context where the objects are the inhibited genes and the attributes are whether that individual gene corresponds to a breast cancer subtype, the extracted concepts reveal which groups of inhibited genes correspond to cancer subtypes. To find which genes are expressed in tissues with lung cancer tumors as opposed to normal tissues, knowledge discovery is used by first statistically selecting features from lung cancer gene expression data, then examining the samples associated with the feature genes [62].

These methods can provide an opportunity to examine the roles of genes in diseases. For a study that uses gene expression data to detect cancer subtypes, we refer the reader to [87]. A future area of this field could lead to detection and classification of disease similarity and subtypes. This concludes our survey of the methods.

4.3 Hierarchical Capability of FCA

As discussed in Section 3.1, FCA, which is based in partial orders, has natural hierarchical capabilities. These include the concept lattice, which for sufficiently small datasets, can allow for visual inspection, thus improving the potential helpfulness of the analysis. Some of the studies previously mentioned do utilize the concept lattice [75, 26, 12]. If the data is prohibitively complex, examining the structure and similarity between multiple concepts can provide a similar benefit [8, 9, 7, 12, 57]. Many studies may benefit from using techniques described in Section 3.3. Most studies, however, merely use FCA techniques to generate synexpression groups without doing further analysis on the concepts generated. In these studies, FCA is reduced to a biclustering method [66]. This may still be an effective way to computationally examine and derive meaning from the dataset, but it may not be as efficient as other gene clustering methods, such as K-Means or other hierarchical clustering methods, such as Optimal Leaf Ordering. Moreover, the hierarchical capability does not obviously appear to be useful in the basic problem of finding synexpression groups. However, in more recent studies, especially those investigating disease, the hierarchical capability of FCA can be used (for example, [8, 9]). Discovering the solutions to more complex problems than finding synexpression groups could make FCA's cost worthwhile. The concept lattice structure may be useful in showing the hierarchy of genes that play a role in cancer, perhaps revealing previously unknown genes that impact the progression of the disease. Further investigations of this may also attempt to classify whether the disease is present or not based on which concepts a gene or genes belong to. Using FCA for disease classification is an area for a future feasibility study.

Chapter 5

Conclusion and Future Work

This thesis describes how Formal Concept Analysis (FCA) provides a framework for analysis of gene expression data, which measures how the cell genetically responds to biological signals. In particular, the concepts generated by FCA from the data can provide biologically relevant information in the form of synexpression groups, or clusters of co-expressed genes. How the data is discretized or how the concepts are generated may provide additional insight depending on the experiment's desired outcome. For example, binarizing the data by overexpression will result in discovering synexpression groups of only genes that are overexpressed. Similarly, choosing one algorithm to generate concepts over another may result in a finer collection of concepts, with little overlap between concepts, or a coarser collection with more overlap.

Recent state-of-the-art advances in this field include analyzing data from multiple experiments [48], using FCA to analyze similarities between diseases [58] or disease subtypes [7], and determining synexpression groups of genes that are negatively correlated [85, 47]. These advances in the field also indicate areas of future interest, which are described as follows:

- **Examining the detection and classification of diseases.** Some work has been done to determine whether diseases are similar to one another, as well as detecting subtypes or two classes of diseases. This can be further expanded to different types of genetically-related diseases, including Alzheimer's and other types of cancer.

- **A user-friendly tool to integrate gene expression data analysis using Fuzzy FCA, Pattern Structures, or Rough Set Theory.** While many studies address the benefits to using these variations, including the minimization of data loss, few user-friendly implementations exist. Effort should be made to explain how data is affected and provide an interface for biological analysis. Integration of these variations, such as $\mathcal{K}$-FCA and Fuzzy FCA, may also be an area of interest.

- **Negative gene synexpression.** While much work has been done in the past fifteen years to identify synexpression groups, the identification of groups of genes that are negatively co-expressed is a promising area. Negative synexpression groups may provide a fuller explanation of gene expression in situations, especially in disease applications such as cancer. This area could also be expanded by incorporating a variation of FCA to hold positive and negative data.

- **Naive binarization data loss.** Depending on how binarization of gene expression data is done, the risk of data loss is very high. Incorporating the strength of expression without an explosion of attributes and maintaining the core definitions of FCA would allow data to be analyzed on existing FCA tools. Additionally, great care should be taken as to how data is discretized depending on the application.

- **Use of the concept lattice in analysis.** Most studies explained in this paper address only the composition of concepts. If sufficient purpose is chosen that can benefit the application, the concept lattice may be useful and the relationships between concepts can be analyzed. FCA concept generation used solely as a clustering method appears to be inefficient, and another, more efficient, clustering method is advised.

5.1 Future Doctoral Work

In addition to the areas mentioned above, there are several key areas of opportunity in other areas of FCA for future research; we focus on two here: FCA for classification of diseases with biological data, and formulation of FCA as a bipartite graph.

5.1.1 FCA Classification of Diseases with Biological Data

Classifying diseases from biological data is an area of bioinformatics with a great amount of potential to help people in the real world. There is existing work addressing FCA's use for classification, mostly for knowledge discovery in databases [74, 13, 33]. Our survey revealed that FCA has been used in two-class classification of breast cancer [69]. However, this is not a diagnostic classification problem, but between two types of the existing disease. Other studies using biological data examines aspects of the disease but do not classify whether the disease is present or not. An FCA algorithm has been designed by Gotoor [43], a member of our research group, that uses an autoencoder to classify images of tissues with colon cancer. The algorithm is based on Fuzzy FCA, and classification is performed based on concept similarity when all concepts have been generated. However, the concepts are not examined due to the abstract nature of the autoencoder-generated data. This shows that FCA classification of whether a disease is present or not in the data given remains an open problem. Our goal is to adapt this existing system to use with gene expression data.

Additionally, our goal is to:

- **Investigate feasibility.** We aim to test the system on a common numerical dataset. This will provide insight as to whether classification is a feasible problem for FCA, and this system in particular.

- **Perform classification with gene expression data.** Since the algorithm already takes as input numerical data, we aim to use this algorithm with an open-source cancer gene expression dataset to classify whether the cancer is present or not, as well as what type of cancer is present.

- **Investigate concepts once generated.** While this may not be feasible for the current algorithm due to the abstract nature of the autoencoder's parameters, using the system with gene expression data instead of the encoded image data will provide the opportunity to investigate the concepts generated, as that would be an efficient use of FCA's hierarchical capabilities.

- **Develop a baseline for discovering classes.** The existing algorithm classifies objects in the same class if they appear in many of the same concepts together. However, naive judgements as based on the concept lattice may be incorrect. For example, the "Bodies of Water" context (Figure 5.1) and accompanying lattice (Figure 5.1) from Wille's 1984 paper shows some potential issues with determining classes solely on the visual results [90]. Two obvious classes appear to be "floating" (containing objects "river", "brook", "canal", and "ditch") and "stagnant" (containing objects "lake", "pond", "basin", and "pool"). However, if classes "large" and "small" are desired instead, the presentation of the lattice may be misleading, giving the false impression that "basin" is "small" since it is on the right side of the diagram, or correspondingly that "ditch" is "large". Similar objects, such as a river and lake, both natural, large bodies of water, may be on opposite sides of the lattice. Depending on the desired group of classes, adding distance functions or other measurements may avoid such confusions.

Table 5.1: Context for the "waters" lexical field as given in [90]

$\mathbb{K}_{\text{waters}}$	floating	stagnant	natural	artificial	large	small
river	X		X		X	
brook	X		X			X
canal	X			X	X	
ditch	X			X		X
lake		X	X		X	
pond		X	X		X	
basin		X		X	X	
pool		X		X		X

This proposed area of research into disease classification with FCA has the potential for real-world benefit. Our discussion above shows the areas for immediate research.

5.1.2 Bipartite FCA

Now we address the formulation of FCA as a bipartite graph. In our survey of the literature, several authors discussed the definition of a context $\mathbb{K} = (G, M, I)$ as a graph $G = (G \cup M, I)$, where G and M are independent sets, and an edge between an object $g \in G$ and attribute $m \in M$ occurs if and only if gIm [4, 26]. A visual representation as shown in [4] is in Figure 5.2.

In this framework, concepts are maximal bicliques, induced complete subgraphs where inclusion of any edge will result in a non-complete subgraph. Some work has been done to link these two fields already [39], including computing clusters with fuzzy measurement-based random walks. However, not much else has been done to

Figure 5.1: Concept lattice for the “waters” lexical field as given in [90]

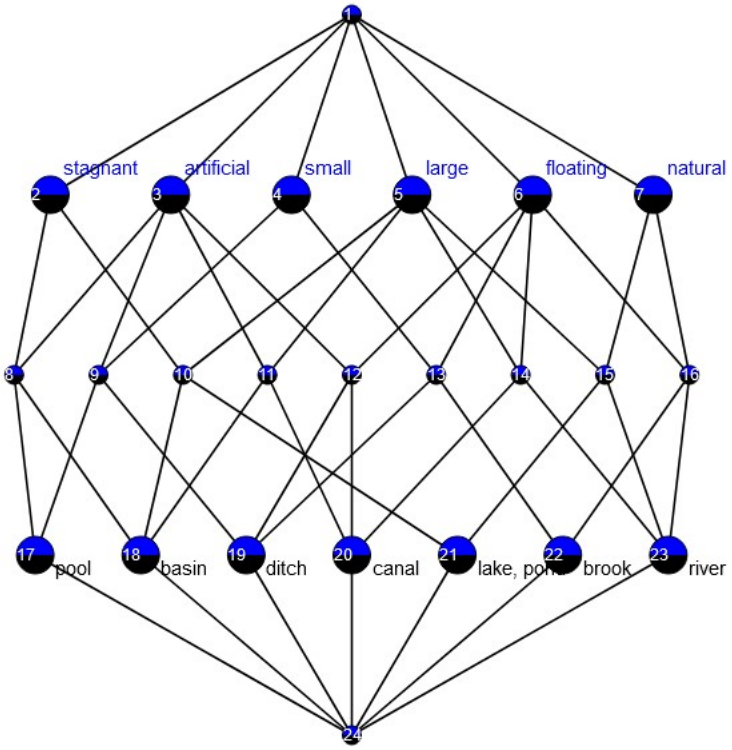
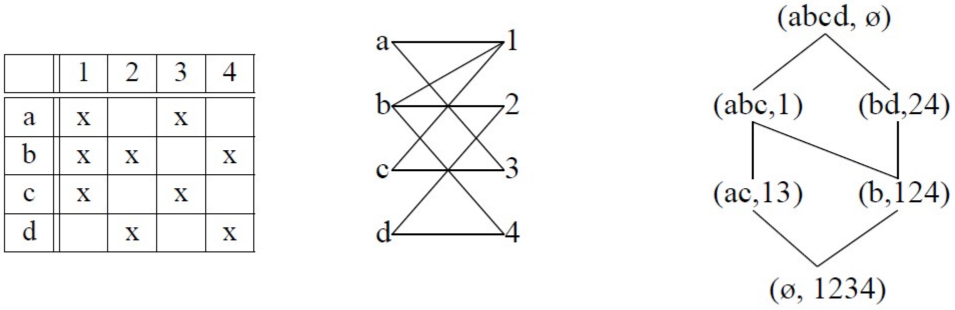


Figure 5.2: Representation of FCA as a bipartite graph, as seen in [26].



	1	2	3	4
a	x		x	
b	x	x		x
c	x		x	
d		x		x

establish useful FCA formalism in a bipartite graph format. Notation in this area can also be standardized based on Wille and Ganter’s FCA notation. Additionally, while some work has been done to link Fuzzy FCA with graph theoretic concepts in [39], this can be further expanded and also be developed to use other extensions such as probabilistic FCA. Once the formalism is developed, common graph algorithms could be adapted to generate concepts and provide other useful FCA features.

Specifically, our goal in this area concerns the following:

- **Formalize notation.** Formalization of notation between bipartite graph theory and Wille and Ganter's standard notation for FCA will benefit future researchers in this area.

- **Develop or adapt an efficient algorithm for graph-based concept generation**. Develop an efficient algorithm or adapt a current graph algorithm to find maximal cliques from the bipartite graph. This will allow for traditional graph-based algorithms to perform functions such as concept generation, lattice formulation, or classification, thus improving the intersection of these two fields

- **Support common FCA variations**. Formulate variations (such as fuzzy or probabilistic) of FCA in detailed theoretic format.

- **Classification**. Once the above tasks have been done, we can develop an incremental graph algorithm to predict new data's biclique membership. This could be done with existing graph algorithms such as strongly connected components, or a variation on other common algorithms.

In this section, we have presented several core problems that will form the basis of our PhD research. These areas for future doctoral work represent the state-of-the-art areas of research for Formal Concept Analysis. We will begin working on these problems as soon as this thesis is submitted. We expect that this will lead to pioneering results in both theory and applications.

Bibliography

[1] A Brief Guide to Genomics. https://www.genome.gov/about-genomics/fact-sheets/A-Brief-Guide-to-Genomics.

[2] Gene Expression Is Analyzed by Tracking RNA — Learn Science at Scitable. https://www.nature.com/scitable/topicpage/gene-expression-is-analyzed-by-tracking-rna-6525038/.

[3] WHO — WHO definitions of genetics and genomics. http://www.who.int/genomics/geneticsVSgenomics/en/.

[4] J. Abello, A. J. Pogel, and L. Miller. Breadth first search graph partitions and concept lattices. *Journal of Universal Computer Science*, 10(8):934–954, Dec. 2004.

[5] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, 96(12):6745–6750, June 1999.

[6] Amedeo Napoli and Thi Nhu Nguyen Le. Lattice editor. https://latviz.loria.fr/about.html.

[7] I. I. Amin, A. E. Hassanien, S. K. Kassim, and H. A. Hefny. Big DNA Methylation Data Analysis and Visualizing in a Common Form of Breast Cancer. In

A. E. Hassanien, A. T. Azar, V. Snasael, J. Kacprzyk, and J. H. Abawajy, editors, *Big Data in Complex Systems: Challenges and Opportunities*, Studies in Big Data, pages 375–392. Springer International Publishing, Cham, 2015.

[8] I. I. Amin, S. K. Kassim, A. e Hassanien, and H. A. Hefny. Formal concept analysis for mining hypermethylated genes in breast cancer tumor subtypes. In *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 764–769, Nov. 2012.

[9] I. I. Amin, S. K. Kassim, A. e Hassanien, and H. A. Hefny. Using formal concept analysis for mining hyomethylated genes among breast cancer tumors subtypes. In *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 521–526, Aug. 2013.

[10] S. Andrews. In-Close2, a High Performance Formal Concept Miner. In S. Andrews, S. Polovina, R. Hill, and B. Akhgar, editors, *Conceptual Structures for Discovering Knowledge*, Lecture Notes in Computer Science, pages 50–62. Springer Berlin Heidelberg, 2011.

[11] S. Andrews and K. McLeod. Gene Co-Expression in Mouse Embryo Tissues. *IJIIT*, 9(4):55–68, Oct. 2013.

[12] S. Andrews and K. McLeod. A Visual Analytics Technique for Exploring Gene Expression in the Developing Mouse Embryo. In P. Chapman, D. Endres, and N. Pernelle, editors, *Graph-Based Representation and Reasoning*, Lecture Notes in Computer Science, pages 137–151. Springer International Publishing, 2018.

[13] Z. Azmeh, M. Huchard, C. Tibermacine, C. Urtado, and S. Vauttier. WSPAB: A Tool for Automatic Classification Selection of Web Services Using Formal

Concept Analysis. In *2008 Sixth European Conference on Web Services*, pages 31–40, Nov. 2008.

[14] J. Baixeries, M. Kaytoue, and A. Napoli. Characterizing functional dependencies in formal concept analysis with pattern structures. *Annals of Mathematics and Artificial Intelligence*, 72(1-2):129–149, Oct. 2014.

[15] E. Bartl and J. Konecny. L-concept analysis with positive and negative attributes. *Information Sciences*, 360:96–111, Sept. 2016.

[16] E. Bartl and J. Konecny. L-Concept lattices with positive and negative attributes: Modeling uncertainty and reduction of size. *Information Sciences*, 472:163–179, Jan. 2019.

[17] R. Belohlavek and V. Vychodil. What is a Fuzzy Concept Lattice? *Proc. Concept Lattices and their Applications*, page 12, 2005.

[18] R. Belohlavek, S. B. Yahia, J. Diatta, P. Eklund, S. O. Kuznetsov, M. Liquière, E. M. Nguifo, U. Priss, S. B. Yahia, K. Bertet, F. Brucker, C. Carpineto, P. Cordero, F. Distel, F. Domenach, M. Ducassé, R. Fuentes-gonzalez, C. V. Glodeanu, M. Huchard, V. G. Kaburlasos, M. Kaytoue, S. Krajci, M. Kryszkiewicz, S. O. Kuznetsov, J. Medina-moreno, and R. Missaoui. *Program Chairs Laszlo Szathmary*. 2012.

[19] S. Benabderrahmane. Formal Concept Analysis and Knowledge Integration for Highlighting Statistically Enriched Functions from Microarrays Data. In *International Work-Conference on Bioinformatics and Biomedical Engineering, IWBBIO 2014, Granada, Spain,*, page 1, Granada, Spain, Apr. 2014.

[20] A. Bertaux, F. Le Ber, A. Braud, and M. Trémolières. Identifying Ecological Traits: A Concrete FCA-Based Approach. In S. Ferré and S. Rudolph, editors, *Formal Concept Analysis*, Lecture Notes in Computer Science, pages 224–236. Springer Berlin Heidelberg, 2009.

[21] J. Besson, C. Robardet, J.-F. Boulicaut, and S. Rome. Constraint-based concept mining and its application to microarray data analysis. *Intelligent Data Analysis*, 9(1):59–82, Jan. 2005.

[22] S. Blachon, R. G. Pensa, J. Besson, C. Robardet, J.-F. Boulicaut, and O. Gandrillon. Clustering Formal Concepts to Discover Biologically Relevant Knowledge from Gene Expression Data. *In Silico Biology*, 7(4,5):467–483, Jan. 2007.

[23] V. G. Blinova, D. A. Dobrynin, V. K. Finn, S. O. Kuznetsov, and E. S. Pankratova. Toxicology analysis by means of the JSM-method. *Bioinformatics*, 19(10):1201–1207, July 2003.

[24] A. Buzmakov and A. Napoli. How Fuzzy FCA and Pattern Structures are connected? In *5th Workshop "What Can FCA Do for Artificial Intelligence?" (FCA4AI'2016)*, Aug. 2016.

[25] J. Chen, J. Mi, B. Xie, and Y. Lin. A fast attribute reduction method for large formal decision contexts. *International Journal of Approximate Reasoning*, 106:1–17, Mar. 2019.

[26] V. Choi, Y. Huang, V. Lam, D. Potter, R. Laubenbacher, and K. Duca. Using formal concept analysis for microarray data comparison. *J. Bioinform. Comput. Biol.*, 06(01):65–75, Feb. 2008.

[27] F. Crick. Central Dogma of Molecular Biology. page 3, 1970.

[28] A. Demin, D. Ponomaryov, and E. Vityaev. Probabilistic Concepts in Formal Contexts. In E. Clarke, I. Virbitskaite, and A. Voronkov, editors, *Perspectives of Systems Informatics*, Lecture Notes in Computer Science, pages 394–410, Berlin, Heidelberg, 2012. Springer.

[29] J. Deogun and L. Jiang. SARM — Succinct Association Rule Mining: An Approach to Enhance Association Mining. In M.-S. Hacid, N. V. Murray, Z. W. Raś, and S. Tsumoto, editors, *Foundations of Intelligent Systems*, Lecture Notes in Computer Science, pages 121–130, Berlin, Heidelberg, 2005. Springer.

[30] J. Deogun, L. Jiang, and V. V. Raghavan. Discovering Maximal Potentially Useful Association Rules Based on Probability Logic. In S. Tsumoto, R. Słowiński, J. Komorowski, and J. W. Grzymała-Busse, editors, *Rough Sets and Current Trends in Computing*, Lecture Notes in Computer Science, pages 274–284, Berlin, Heidelberg, 2004. Springer.

[31] J. Deogun, L. Jiang, Y. Xie, and V. Raghavan. Probability Logic Modeling of Knowledge Discovery in Databases. In N. Zhong, Z. W. Raś, S. Tsumoto, and E. Suzuki, editors, *Foundations of Intelligent Systems*, Lecture Notes in Computer Science, pages 402–407, Berlin, Heidelberg, 2003. Springer.

[32] S. M. Dias and N. J. Vieira. Concept lattices reduction: Definition, analysis and classification. *Expert Systems with Applications*, 42(20):7084–7097, Nov. 2015.

[33] B. Díaz-Agudo and P. A. González-Calero. Classification Based Retrieval Using Formal Concept Analysis. In D. W. Aha and I. Watson, editors, *Case-Based Reasoning Research and Development*, Lecture Notes in Computer Science, pages 173–188, Berlin, Heidelberg, 2001. Springer.

[34] G. Dong, C. Jiang, J. Pei, J. Li, and L. Wong. Mining Succinct Systems of Minimal Generators of Formal Concepts. In L. Zhou, B. C. Ooi, and X. Meng, editors, *Database Systems for Advanced Applications*, Lecture Notes in Computer Science, pages 175–187. Springer Berlin Heidelberg, 2005.

[35] A. Formica. Similarity reasoning in formal concept analysis: From one- to many-valued contexts. *Knowl Inf Syst*, 60(2):715–739, Aug. 2019.

[36] B. Ganter and S. O. Kuznetsov. Pattern Structures and Their Projections. In H. S. Delugach and G. Stumme, editors, *Conceptual Structures: Broadening the Base*, Lecture Notes in Computer Science, pages 129–142, Berlin, Heidelberg, 2001. Springer.

[37] B. Ganter, S. Rudolph, and G. Stumme. Explaining Data with Formal Concept Analysis. In M. Krötzsch and D. Stepanova, editors, *Reasoning Web. Explainable Artificial Intelligence: 15th International Summer School 2019, Bolzano, Italy, September 20–24, 2019, Tutorial Lectures*, Lecture Notes in Computer Science, pages 153–195. Springer International Publishing, Cham, 2019.

[38] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer Science & Business Media, Dec. 2012.

[39] B. Gaume, E. Navarro, and H. Prade. A Parallel between Extended Formal Concept Analysis and Bipartite Graphs Analysis. In E. Hüllermeier, R. Kruse, and F. Hoffmann, editors, *Computational Intelligence for Knowledge-Based Systems Design*, Lecture Notes in Computer Science, pages 270–280, Berlin, Heidelberg, 2010. Springer.

[40] J. Gebert, S. Motameny, U. Faigle, C. V. Forst, and R. Schrader. Identifying Genes of Gene Regulatory Networks Using Formal Concept Analysis. *Journal of Computational Biology*, 15(2):185–194, Mar. 2008.

[41] A. Gély, M. Couceiro, and A. Napoli. Steps Towards Achieving Distributivity in Formal Concept Analysis. June 2018.

[42] J. M. González-Calabozo, F. J. Valverde-Albacete, and C. Peláez-Moreno. Interactive knowledge discovery and data mining on genomic expression data with numeric formal concept analysis. *BMC Bioinformatics*, 17(1):374, Sept. 2016.

[43] N. R. Gotoor. Image Classification using Fuzzy FCA. *Embargoed Master's Theses*, Dec. 2019.

[44] D. Grissa, B. Comte, M. Pétéra, E. Pujos-Guillot, and A. Napoli. A hybrid and exploratory approach to knowledge discovery in metabolomic data. *Discrete Applied Mathematics*, Jan. 2019.

[45] R. Henriques, C. Antunes, and S. C. Madeira. A structured view on pattern mining-based biclustering. *Pattern Recognition*, 48(12):3941–3958, Dec. 2015.

[46] A. Houari, W. Ayadi, and S. Ben Yahia. A new FCA-based method for identifying biclusters in gene expression data. *Int. J. Mach. Learn. & Cyber.*, 9(11):1879–1893, Nov. 2018.

[47] A. Houari, W. Ayadi, and S. B. Yahia. NBF: An FCA-Based Algorithm to Identify Negative Correlation Biclusters of DNA Microarray Data. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pages 1003–1010, May 2018.

[48] A. Hristoskova, V. Boeva, and E. Tsiporkova. A formal concept analysis approach to consensus clustering of multi-experiment expression data. *BMC Bioinformatics*, 15(1):151, May 2014.

[49] D. Jiang, C. Tang, and A. Zhang. Cluster Analysis for Gene Expression Data: A Survey. https://www.computer.org/csdl/journal/tk/2004/11/k1370/13rRUy2YLTf, Nov. 2004.

[50] L. Jiang. *New Data Mining Models Based on Formal Concept Analysis and Probability Logic.* Ph.D., The University of Nebraska - Lincoln, United States – Nebraska, 2006.

[51] N. Juniarta, M. Couceiro, and A. Napoli. A Unified Approach to Biclustering Based on Formal Concept Analysis and Interval Pattern Structure. Oct. 2019.

[52] J. B. Juvilla. *Lattice characterization of armstrong and symmetric dependencies.* http://purl.org/dc/dcmitype/Text, Universitat Politècnica de Catalunya (UPC), 2007.

[53] M. Kaytoue, S. Duplessis, S. O. Kuznetsov, and A. Napoli. Two FCA-Based Methods for Mining Gene Expression Data. In S. Ferré and S. Rudolph, editors, *Formal Concept Analysis*, Lecture Notes in Computer Science, pages 251–266. Springer Berlin Heidelberg, 2009.

[54] M. Kaytoue, S. O. Kuznetsov, J. Macko, and A. Napoli. Biclustering meets triadic concept analysis. *Ann Math Artif Intell*, 70(1):55–79, Feb. 2014.

[55] M. Kaytoue, S. O. Kuznetsov, and A. Napoli. Biclustering Numerical Data in Formal Concept Analysis. In P. Valtchev and R. Jäschke, editors, *Formal*

Concept Analysis, Lecture Notes in Computer Science, pages 135–150. Springer Berlin Heidelberg, 2011.

[56] M. Kaytoue, S. O. Kuznetsov, A. Napoli, and S. Duplessis. Mining gene expression data with pattern structures in formal concept analysis. *Information Sciences*, 181(10):1989–2001, May 2011.

[57] M. Kaytoue-Uberall, S. Duplessis, and A. Napoli. Using Formal Concept Analysis for the Extraction of Groups of Co-expressed Genes. In H. A. Le Thi, P. Bouvry, and T. Pham Dinh, editors, *Modelling, Computation and Optimization in Information Systems and Management Sciences*, Communications in Computer and Information Science, pages 439–449. Springer Berlin Heidelberg, 2008.

[58] B. J. Keller, F. Eichinger, and M. Kretzler. Formal Concept Analysis of Disease Similarity. *AMIA Jt Summits Transl Sci Proc*, 2012:42–51, Mar. 2012.

[59] E. W. Khandjian and C. Méric. A procedure for Northern blot analysis of native RNA. *Analytical Biochemistry*, 159(1):227–232, Nov. 1986.

[60] P. Krajca, J. Outrata, and Vilem Vychodil. Parallel Recursive Algorithm for FCA. In *CLA 2008, Proceedings of the Sixth International Conference on Concept Lattices and Their Applications*, pages 83–94, Olomouc, Czech Republic, Oct. 2008.

[61] P. Krajca, J. Outrata, and V. Vychodil. Advances in algorithms based on CbO. page 13, 2010.

[62] Y. Li, W. Hong, S. Li, J. Song, and X. Liu. Cancer Gene Expression Data Attribute Partial Ordered Representation and Knowledge Discovery. In *2015*

Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), pages 861–865, Sept. 2015.

[63] V. Liñeiro-Barea, J. Medina, and I. Medina-Bulo. Generating Fuzzy Attribute Rules Via Fuzzy Formal Concept Analysis. In L. T. Kóczy and J. Medina, editors, *Interactions Between Computational Intelligence and Mathematics*, Studies in Computational Intelligence, pages 105–119. Springer International Publishing, Cham, 2018.

[64] S. Lopes, J.-M. Petit, and L. Lakhal. Functional and approximate dependency mining: Database and FCA points of view. *Journal of Experimental & Theoretical Artificial Intelligence*, 14(2-3):93–114, Apr. 2002.

[65] J. Macko. User-Friendly Fuzzy FCA. In P. Cellier, F. Distel, and B. Ganter, editors, *Formal Concept Analysis*, Lecture Notes in Computer Science, pages 156–171, Berlin, Heidelberg, 2013. Springer.

[66] S. Madeira and A. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, Jan. 2004.

[67] R. Medina and L. Nourine. Conditional Functional Dependencies: An FCA Point of View. In L. Kwuida and B. Sertkaya, editors, *Formal Concept Analysis*, Lecture Notes in Computer Science, pages 161–176. Springer Berlin Heidelberg, 2010.

[68] N. Messai, M.-D. Devignes, A. Napoli, and M. Smail-Tabbone. Many-Valued Concept Lattices for Conceptual Clustering and Information Retrieval. In *Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on*

Artificial Intelligence, pages 127–131, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.

[69] S. Motameny, B. Versmold, and R. Schmutzler. Formal Concept Analysis for the Identification of Combinatorial Biomarkers in Breast Cancer. In R. Medina and S. Obiedkov, editors, *Formal Concept Analysis*, Lecture Notes in Computer Science, pages 229–240. Springer Berlin Heidelberg, 2008.

[70] A. Napoli. Chapter 41 - A SMOOTH INTRODUCTION TO SYMBOLIC METHODS FOR KNOWLEDGE DISCOVERY. In H. Cohen and C. Lefebvre, editors, *Handbook of Categorization in Cognitive Science*, pages 913–933. Elsevier Science Ltd, Oxford, Jan. 2005.

[71] G. S. Pall and A. J. Hamilton. Improved northern blot method for enhanced detection of small RNA. *Nature Protocols*, 3(6):1077–1084, June 2008.

[72] R. G. Pensa and J.-F. Boulicaut. Towards Fault-Tolerant Formal Concept Analysis. In S. Bandini and S. Manzoni, editors, *AI*IA 2005: Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages 212–223. Springer Berlin Heidelberg, 2005.

[73] J. L. Pfaltz and C. M. Taylo. Closed Set Mining of Biological Data. In *BIOKDD02*, Edmonton, Alberta, Canada, July 2002.

[74] J. Poelmans, D. I. Ignatov, S. O. Kuznetsov, and G. Dedene. Formal concept analysis in knowledge processing: A survey on applications. *Expert Systems with Applications*, 40(16):6538–6560, Nov. 2013.

[75] D. P. Potter. A combinatorial approach to scientific exploration of gene expression data: An integrative method using Formal Concept Analysis for the comparative analysis of microarray data. Aug. 2005.

[76] U. Priss. Representing Median Networks with Concept Lattices. In H. D. Pfeiffer, D. I. Ignatov, J. Poelmans, and N. Gadiraju, editors, *Conceptual Structures for STEM Research and Education*, Lecture Notes in Computer Science, pages 311–321. Springer Berlin Heidelberg, 2013.

[77] K. Qin, H. Lin, and Y. Jiang. Local attribute reductions of formal contexts. *Int. J. Mach. Learn. & Cyber.*, Apr. 2019.

[78] K. Raza. Formal Concept Analysis for Knowledge Discovery from Biological Data. *IJDMB*, 18(4):281, 2017.

[79] F. Rioult. Mining concepts from large SAGE gene expression matrices. In *KDID*, Sept. 2003.

[80] M. Rouane-Hacene, Y. Toussaint, and P. Valtchev. Mining Safety Signals in Spontaneous Reports Database Using Concept Analysis. In C. Combi, Y. Shahar, and A. Abu-Hanna, editors, *Artificial Intelligence in Medicine*, Lecture Notes in Computer Science, pages 285–294, Berlin, Heidelberg, 2009. Springer.

[81] K. Sato, Y. Okubo, M. Haraguchi, and S. Kunifuji. Data Mining of Time-Series Medical Data by Formal Concept Analysis. In B. Apolloni, R. J. Howlett, and L. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, Lecture Notes in Computer Science, pages 1214–1221, Berlin, Heidelberg, 2007. Springer.

[82] D. Slezak and J. Wroblewski. Rough Discretization of Gene Expression Data. In *2006 International Conference on Hybrid Information Technology*, volume 2, pages 265–267, Nov. 2006.

[83] D. Ślęzak and J. Wróblewski. Roughfication of Numeric Decision Tables: The Case Study of Gene Expression Data. In J. Yao, P. Lingras, W.-Z. Wu, M. Szczuka, N. J. Cercone, and D. Ślęzak, editors, *Rough Sets and Knowledge Technology*, Lecture Notes in Computer Science, pages 316–323. Springer Berlin Heidelberg, 2007.

[84] K. Sumangali and C. Aswani Kumar. Concept Lattice Simplification in Formal Concept Analysis Using Attribute Clustering. *J Ambient Intell Human Comput*, May 2018.

[85] X. Tu, Y. Wang, M. Zhang, and J. Wu. Using Formal Concept Analysis to Identify Negative Correlations in Gene Expression Data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13(2):380–391, Mar. 2016.

[86] D. van der Merwe, S. Obiedkov, and D. Kourie. AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. In P. Eklund, editor, *Concept Lattices*, Lecture Notes in Computer Science, pages 372–385, Berlin, Heidelberg, 2004. Springer.

[87] P. Vasudevan and T. Murugesan. Cancer Subtype Discovery Using Prognosis-Enhanced Neural Network Classifier in Multigenomic Data. *Technol Cancer Res Treat*, 17:1533033818790509, Jan. 2018.

[88] E. E. Vityaev, A. V. Demin, and D. K. Ponomaryov. Probabilistic generalization of formal concepts. *Program Comput Soft*, 38(5):219–230, Sept. 2012.

[89] Y. Wan and L. Zou. An Efficient Algorithm for Decreasing the Granularity Levels of Attributes in Formal Concept Analysis. *IEEE Access*, 7:11029–11040, 2019.

[90] R. Wille. Line diagrams of hierarchical concept systems. *KNOWLEDGE ORGANIZATION*, 11(2):77–86, 1984.

[91] R. Wille. RESTRUCTURING LATTICE THEORY: AN APPROACH BASED ON HIERARCHIES OF CONCEPTS. In S. Ferré and S. Rudolph, editors, *Formal Concept Analysis*, Lecture Notes in Computer Science, pages 314–339. Springer Berlin Heidelberg, 2009.

[92] S. A. YEVTUSHENKO. System of data analysis. *Proc. 7th National Conference on Artificial Intelligence (KII'00)*, pages 127–134, 2000.

[93] L. Zou, Z. Zhang, and J. Long. An efficient algorithm for increasing the granularity levels of attributes in formal concept analysis. *Expert Systems with Applications*, 46:224–235, Mar. 2016.