

Visualization for Frequent Pattern Mining

by

Christopher Lee Carmichael

A thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada

March 2013

Copyright © 2013 by Christopher Lee Carmichael

Thesis advisor

Author

Dr. Carson K. Leung

Christopher Lee Carmichael

Visualization for Frequent Pattern Mining

Abstract

Data mining algorithms analyze and mine databases for discovering implicit, previously unknown and potentially useful knowledge. Frequent pattern mining algorithms discover sets of database items that often occur together. Many of the frequent pattern mining algorithms represent the discovered knowledge in the form of a long textual list containing these sets of frequently co-occurring database items. As the amount of discovered knowledge can be large, it may not be easy for most users to examine and understand such a long textual list of knowledge. In my M.Sc. thesis, I represent both the original database and the discovered knowledge in pictorial form. Specifically, I design a new interactive visualization system for viewing the original transaction data (which are then fed into the frequent pattern mining engine) and for revealing the interesting knowledge discovered from the transaction data in the form of mined patterns.

Table of Contents

Abstract	ii
Table of Contents	iv
List of Figures	v
List of Tables	vii
Acknowledgements	viii
1 Introduction	1
1.1 Thesis Statement	5
1.2 Contributions	6
1.3 Thesis Organization	6
2 Background and Related Work	8
2.1 Data Mining	8
2.1.1 Transactions	9
2.1.2 Itemsets	10
2.1.3 Association Rules	11
2.2 Data and Result Visualization	12
2.2.1 Data Visualization	12
2.2.2 Result Visualization	16
2.3 Summary	18
3 Transaction and Itemset Visualization	20
3.1 Basic Representation	21
3.1.1 Visualizing Transactions	22
3.1.2 Visualizing Mined Itemsets	23
3.1.3 Our System for Visualizing Transactions and Mined Itemsets	25
3.2 Frequency Graph (FGraph)	25
3.2.1 Item Order	29
3.2.2 Prefix/Extension Paradigm	31
3.3 Existence Graph (EGraph)	33
3.4 Interactive Features	34

3.4.1	Itemset Selection	34
3.4.2	Itemset Group Editor	37
3.4.3	Data Mining	39
3.4.4	Item Editor	39
3.5	Summary	43
4	Case Studies	55
4.1	Mushroom Data	56
4.2	Wine Data	74
4.3	Coauthorship Data	84
4.3.1	Overview of the Coauthorship Data	85
4.3.2	Q1: Who has collaborated most with Dr. Domaratzki?	87
4.3.3	Q2: Who has collaborated with Dr. Leung?	88
4.3.4	Q3: Which individual researcher has collaborated with Dr. Leung?	88
4.3.5	Q4: Compare the results to Q2 and Q3.	93
4.3.6	Q5: What size of group does Dr. Irani usually participate in?	93
4.3.7	Q6: Of the authors from Q5, who did Dr. Irani collaborate with and how often did he do it?	95
4.4	Summary	98
5	Conclusions and Future Work	101
5.1	Conclusions	101
5.2	Future Work	104
5.2.1	Aggregate Item Automatic Mining and Grouping	105
5.2.2	Transaction Counts	106
5.2.3	Itemset Search	106
5.2.4	Generalization of the Visualization for Presenting Other Graphs	107
	Bibliography	109

List of Figures

2.1	Mackinlay's ranked perceptual tasks.	14
2.2	A parallel coordinate plot of datum (4,-2,1,3,0).	15
2.3	Yang's system displays itemsets and association rules on parallel coordinates.	16
3.1	Basic glyph representation of transactions.	23
3.2	Basic glyph representation of itemsets.	24
3.3	Itemsets from Table 3.2 and from UC Irvine Mushroom Data.	26
3.4	FGraph can imply association rule support and confidence.	28
3.5	Corollary support information	30
3.6	FGraph contrasting edible 2-sets with different item orders.	45
3.7	Comparing subset/superset and prefix/extension relationships.	46
3.8	FGraph displaying supersets and subsets.	47
3.9	FGraph displaying supersets and subsets with ordered items.	48
3.10	The EGraph is used display individual highlighted itemsets	49
3.11	The extension interface adds extensions that have already been mined.	50
3.12	Selection filtering dialogue.	51
3.13	Three itemset groups coloured red green and blue.	52
3.14	Constraint based data mining interface.	53
3.15	Item editor interface.	54
4.1	Raw mushroom transactions.	57
4.2	Frequent singletons mined from mushroom.	58
4.3	One-item extensions of both edible and poisonous mushrooms.	60
4.4	Item editor.	61
4.5	Groups of edible and poisonous mushroom.	62
4.6	Poisonous mushrooms plus aggregate items P1 and !P1.	63
4.7	Mushroom data: poisonous and edible groups P2 and E2	65
4.8	Mushroom data: poisonous group P3 and edible group E3.	66
4.9	Mushroom data: summary	68
4.10	Mushroom data summary: poisonous group P1	70

4.11 Mushroom data summary: edible group E1	71
4.12 Mushroom data summary: poisonous group P2 and edible group E2	72
4.13 Mushroom data summary: poisonous group P3 and edible group E3	73
4.14 Wine data: class 1, 2, 3, extensions	75
4.15 Wine data: class 1 and non-class 1 extensions.	77
4.16 Wine data: malic acid wine class distribution.	78
4.17 Wine data: ash outlier extensions.	79
4.18 Wine data: class 1 aggregate ranges added	82
4.19 Wine data: refined order, class 1 aggregate ranges	83
4.20 Coauthorship data overview.	86
4.21 Q1: Who have collaborated most with with Dr. Domaratzki with?	89
4.22 Salomaa's publications.	90
4.23 Q2: Who (one or multiple authors) have collaborated with Dr. Leung?	91
4.24 Q3: Which individual researcher has collaborated with Dr. Leung?	92
4.25 Q4: Compare the results to Q2 and Q3	94
4.26 DBLP: Compare Q1 and Q2	96
4.27 Q6: Of the authors from Q5, who did Dr. Irani collaborate with and how often did he do it?	97
4.28 Dr. Irani 3-set collaborations	99
5.1 Possible graph visualization	108

List of Tables

3.1	Raw transaction database before data mining	22
3.2	Mined itemsets from Table3.1	23

Acknowledgements

I thank my research supervisor, Dr. Carson K. Leung, for his encouragement and academic & financial support. This thesis would not have been possible without him. Back when I was an undergraduate student, I took all three database and data mining courses as well as the Undergraduate Honours project course with Dr. Leung. His enthusiasm and inspiration make me interested in the research area of data mining. After I finished my undergraduate degree, I decided to stay in this research area, which gave me an opportunity to research on this M.Sc. thesis.

I would thank Dr. Qingjin Peng, who is currently on sabbatical, for his feedback on my thesis proposal. I would also thank my fellow members, especially Juan J. Cameron, in the Database and Data Mining Laboratory for their support, help and assistance.

Moreover, I would like to thank my thesis examination committee members—Dr. Xikui Wang and Dr. Pourang P. Irani—for their precious time to read and examine my thesis. I would also like to thank Dr. Yang Wang for chairing my thesis defense.

CHRISTOPHER LEE CARMICHAEL
B.Sc.(Honours), The University of Manitoba, Canada, 2007

The University of Manitoba
March 2013

Chapter 1

Introduction

Data mining algorithms mine databases to discover implicit, previously unknown and potentially useful knowledge. Frequent pattern mining [AIS93, AS94, LCH07] is one of the important data mining tasks. Frequent pattern mining algorithms discover sets of database items that often occur together. These sets of items are also known as *itemsets*. An example of an itemsets is {apples, bananas}, which represents that apples and bananas are being purchased together by customers in a grocery store. Another example of itemsets is {poisonous, smells foul}, which represents some co-occurring characteristics of some objects such as mushrooms (i.e., a poisonous mushroom with foul smell).

In addition to itemsets, frequent pattern mining algorithms also return the frequency of these itemsets. Comparing frequencies of related itemsets provides associate information between them. For example, the knowledge about a collection of itemsets can help determine the cause and effect relationships between sets of items. Given some event A has occurred, itemsets can help determine how likely event B will also

occur. Studying itemsets can reveal complex knowledge that may otherwise have been missed or not known at all. For instance, mining a general survey with hundreds of questions, one may discover a collection of itemsets that reveal how many people had Crohn's disease (a form of inflammatory bowel disease), used Isotretinoin (Accutane) acne medication, and have had some form of cancer.

Many of the frequent pattern mining algorithms represent the discovered knowledge in the form of a long textual list containing these itemsets. When only a few itemsets are considered, a textual list is capable of showing which items occurred together and how often. However, a few itemsets are not common in data mining. Since every possible combination of items can be counted, the number of discovered itemsets—even from a small database—can quickly grow beyond the ability to comprehend with text. For example, a database composed of n items can lead to about 2^n itemsets (i.e., 2^n possible combinations of the n items). Consider a situation, in which each participant answers 10 true-or-false questions (i.e., 10 items) in a survey. There are 1,024 possible combinations of true or false answers (i.e., 1,024 itemsets). So, for humans to see the information contained, it is necessary to organize and present this data in a way that can be easily understood.

As the amount of discovered knowledge can be large (and is usually much larger than just 10 items), it may not be easy for most users to examine and understand such a long textual list of knowledge. Here, in my M.Sc. thesis, I represent the discovered knowledge in pictorial form. Specifically, I design a new interactive system for itemset visualization. This system is capable of visualizing the raw data of the database transactions (e.g., the true-or-false answers in each survey), since a transaction is also

an itemset. For example, each set of attributes that describe the different characteristics of a mushroom make up a transaction. These sets of attributes are themselves sets of data base items or itemsets. In addition, my interactive system is also capable of visualizing the discovered knowledge (e.g., popular true-or-false combinations in the survey, some common characteristics of mushrooms). Visualizing itemsets can be useful because it helps users (e.g., physicians) to find out the number of patients suffering from Crohn's disease for the above real-life application. Moreover, itemset visualization also helps users to easily compare the number of patients who did not use Accutane with the number of people who used Accutane but did not get Crohn's disease or cancer.

There are a few existing visualization systems [Yan05, HC99, HAC00, HC00]. Although they can be used for visualizing itemsets, many of them were not designed for this purpose. They were designed to visualize other information such as association rules. Among the systems that were designed to visualize itemsets [LJI11, LJ12], some focused on visualizing closed itemsets [CL10], visualizing social networks [LC10], contrasting patterns [CHL11]. Some others [Kon06] are not easy to compare frequencies and/or which items were contained. For instance, some of these systems use colour and some use the thickness of a line or a curve to represent frequency. One can easily imagine that it is not easy to compare two similar colours or compare the thickness of two lines/curves. As such, it is not easy for users to compare two itemsets. As a preview, I solve this problem by using the screen positioning (instead of colour or line/curve thickness). Some follow-up questions include: "How can screen position be used to display both items contained in each itemset and their frequencies?" If

the itemset frequency was mapped to the y -position, then one could easily get a sense of which itemset is more frequent by looking at the height. Moreover, one could also easily get a sense of how much more frequent the itemset is. The same is true for the items. If the x -position was used to display which items are contained in each itemset, then their related itemsets could easily be found just by looking at the correct vertical column.

The problem of visualizing itemsets is more complicated than just finding a way to represent a million itemsets on the screen at one time. The number of times a set of items occurred together is not the only thing that needs to be considered. This information has to be given context. If 100 mushrooms are found to be poisonous and smelling foul, what does that mean? One cannot conclude that all mushrooms are poisonous, nor can one conclude that all poisonous mushrooms smell foul. To determine if all mushrooms are poisonous, one needs to know how many mushrooms were counted. To determine if all poisonous mushrooms smell foul, one needs to know how many poisonous mushrooms were counted.

Another challenge is how to present the itemsets in the context that the user has in mind. A dialogue is necessary for the user to describe which itemsets are to be mined because there are far too many combinations to mine all. A dialogue is necessary to determine which items to include, and a dialogue is necessary to compare and contrast the itemsets being presented.

The third challenge is frequent pattern mining algorithms usually produce a large number of itemsets, which may overwhelm the physical memory, the available hard drive space, and/or the user. For example, the mushroom data from UC Irvine,

which is a typical benchmark dataset, contains more than 8000 mushroom records with more than 100 distinct characteristics describing the mushroom. In other words, potentially $2^{100} \approx 1.2 \times 10^{30}$ different itemsets can be discovered. Users can reduce the number of itemsets by applying a minimum support constraint. This limits the number of itemsets produced by only counting those itemsets that had a frequency greater than or equal to the user-specific threshold. This was a good idea because it is often, but not always, the case that itemsets that occur most frequently are the most interesting. Depending on the context of what one is looking for, it may be interesting to know whether or not two items occur together at all (e.g., “Can a mushroom smell foul and smell fishy?”).

The fourth challenge is how to compare and contrast two or more discovered itemsets. My designed itemset visualization system is capable of highlighting the similarities or differences between different data items (e.g., “How many mushrooms are poisonous compared to how many are edible?”, “Show me the attributes that only occur on poisonous mushrooms”, “Which of those attributes are most frequent?”). My system provides a dialogue so the user can assign colours, shapes, sizes, and other visual tasks to compare and contrast one group of data to another.

1.1 Thesis Statement

Motivated by the above challenges, my **M.Sc. thesis statement** is as follows:

Develop an easy-to-understand, interactive, visual system to display and explore raw transactions and itemsets mined from a database.

1.2 Contributions

The key contribution of this work is a novel interactive and scalable itemset visual system. This research provides users with effective visual support for data analysis and knowledge discovery. Users can derive insight from raw data, guide a constraint based data mining algorithm in a focused manner to explore the data depending on whatever context they have in mind. Using many interactive features this system will help users compare, contrast and present their findings.

1.3 Thesis Organization

This thesis is organized as follows. Next chapter provides background and related work, followed by Chapter 3 describing my system. To demonstrate the effectiveness of my visualization system, Chapter 4 is dedicated to three case studies that provide varying types of data and real world scenarios that can be solved using my system. Conclusion and future work is given in Chapter 5.

The background and related work on Chapter 2 reviews work from both fields of data mining and data visualization. I give a detailed look at the limitations of data stored in a transaction database. I discuss basic algorithms used to mine itemsets from transaction database. I also discuss association rules, their importance, and some of the systems that visualize them. A conscious decision was made to visualize itemsets over association rules. The reasons are given, and how the results justifying that decision are considered. The graphs used to represent itemsets were inspired by the parallel coordinate plot because it fits so nicely with Machinlay's ranked perceptual

tasks. However, some modifications were necessary to display itemsets. These are also discussed at the end of the chapter.

In Chapter 3, my visual system is presented. It is divided into sections that cover the general glyph/graph itemset representation, the two graphs that are used to display itemsets, the prefix/extension paradigm that my system uses, and the interactive features that provide a dialogue for the user to express exactly what context they are interested in. The design of these two graphs are inspired by FIsViz [LIC08a], WiFIsViz [LIC08b], and FpVAT [LC09a] (which comprises FpViz [LC09b]).

Chapter 4 demonstrates the effectiveness of the system as a whole. Chapter 3 describes all of the individual interactive features, and Chapter 4 shows the practical use of these features in real world scenarios such as coauthorship information [LC11, LCT11].

Chapter 5 provides a conclusion of this thesis and path for future research.

Chapter 2

Background and Related Work

As my M.Sc. thesis work involves both data mining and data visualization, I provide in this chapter background and related work in these two areas. I first review the process of mining transactions in databases to discover itemsets, which can then be used to form association rules revealing the relationships between related itemsets. As the number of discovered itemsets and association rules can be large, it leads to some challenges in representing itemsets. We then review related work on the visual representation of itemsets. We also discuss different aspects of visualization—say, perceptive tasks such as colour, size and position in the parallel coordinate system or other related representation.

2.1 Data Mining

The formal definition of data mining states “data mining is the non-trivial extraction of implicit, previously unknown, and potentially useful information from

data” [FPM91]. Data mining discovers the useful relationships and patterns within data. Common data mining tasks include (1) classification [WAT11], (2) clustering [FTM⁺11], and (3) the topic of this thesis—association rule mining [CF11]. Association rules are formed by putting frequent itemsets in the antecedents and consequences of the rules. Frequent itemsets are discovered in *transactions* in the databases.

2.1.1 Transactions

A *transaction* is a set of database items that has occurred together at some event or on some object. A simple example is the stored record of a customer exchange at a grocery store. If a customer purchases apples, bananas and carrots together, then the transaction {apples, bananas, carrots} is entered into the database and given some unique identification number (i.e., transaction ID) that can be referenced later. A transaction usually stores a limited amount of data. For simplicity, we do not keep the quantitative information (e.g., “how many pounds of apples, bunches of bananas or baskets of carrots were purchased together at one time?”). The information stored in each transaction is limited to which items have occurred together.

Each transaction in a database consists of one or more items. In other words, each transaction can be considered as a set of items. Consider a database consisting of three publications: {Alice, Bob}, {Alice, Bob}, and {Alice, Bob, Carl}. Here, the first and second papers are coauthored by Alice and Bob. The third paper is coauthored by Alice, Bob and Carl. Here, the first two transactions reveal that the number of papers authored by Alice and Bob *alone* is 2. In contrast, the itemset {Alice, Bob} discovered from frequent pattern mining reveals that the number of

papers coauthored by at least Alice and Bob (and may include other coauthors) is 3.

An item in an itemset may represent a distinct value (e.g., the universal product code of a merchandise product) or a discretized value in a continuous range. The wine dataset used in a case study in Chapter 4 is an example for the latter. In the original database, each analogue wine constituent is converted to a discrete range of values. Specifically, the alcohol content of the wines range from 11.0 to 14.8% was discretized into eight ranges of 0.5 increment starting at 11%: (1) Alcohol:[11,11.5), (2) Alcohol:[11.5,12), ..., (8) Alcohol:[14.5,15]. Each individual wine is represented by a transaction, and the appropriate alcohol range for that wine is included in that transaction.

2.1.2 Itemsets

An *itemset* is a set of items. There are many algorithms designed to discover itemsets from transactions. These include the Apriori algorithm [AS94] and the FP-growth algorithm [HPYM04]. As mentioned, for each transaction of length n , potentially $2^n - 1$ non-empty itemsets can be extracted. To reduce the total number of itemsets produced, most frequent pattern mining algorithms only find frequent itemsets. An itemset is frequent if its frequency \geq user-specified frequency threshold *minsup*. Note that, if an itemset is frequent, then all its subsets are frequent. Equivalently, if an itemset is infrequent, then all its supersets are guaranteed to be infrequent and thus can be pruned.

In some real-life applications, user may want to limit the number of itemsets. He can do so by specifying some constraints. Many itemset mining algorithms have

been introduced that have further constraints [SVA97, BJ01, AR11]. Leung et al. [LJSW12] provides a good summary of the different overlapping classifications of constraints: succinct, anti-monotone, monotone, convertible anti-monotone, convertible monotone constraints. These constraints apply to individual item properties like the price of bananas. For example, the succinct constraint “ $\max(X.\text{price}) > \$30$ ” mines only those itemsets that contain an item with a price greater than \$30. The system presented in this thesis also offers a user-guided constrained mining algorithm. It uses the FP-growth with constraints algorithm [LLN03]. However, in some applications, users simply want to express their preference of including or excluding certain items not by their properties but ID. As a preview, my system developed in this M.Sc. thesis provides the users with a few constraint options including one allowing users to express their preference of including or excluding certain items by item IDs.

2.1.3 Association Rules

An *association rule* is generally of the form $A \Rightarrow C$, where A and C are frequent itemsets, (A being the antecedent and $A \cup C$ is the consequence). Since the antecedent is always assumed to be a subset of the consequence, it is generally left out of the expression [AIS93]. Support and confidence are two common measures used to express “*interestingness*” of the rule, but there are some others: lift, coverage, odds ratio, conviction, etc. [GH06]. The *support* of a rule, $\text{sup}(A \Rightarrow C)$, is the number of times itemset $A \cup C$ occurs in the database. The *confidence* of a rule is the conditional probability of the occurrence of C among the occurrences of A . It is the ratio of the $A \cup C$ occurrences to the occurrences of A alone: $\frac{\text{sup}(A \cup C)}{\text{sup}(A)}$.

If there are n items, $2^{(n-1)} - 1$ non-empty antecedents are possible. For each antecedent of length k , there are $2^{(n-k-1)} - 1$ possible consequences. Association rules describe how an itemset is related with another. As a preview, although association rules are not displayed explicitly in our visualization system, it does provide users with ways to imply the support and confidence of association rules based on the positions of the two associated itemsets. The *support* measures the frequency of itemsets, and the *confidence* measures the conditional probability of having the consequence given the antecedent. Since our itemset visualization system uses vertical positioning to display frequency, users can get a sense of the proportional difference between one itemset and another.

2.2 Data and Result Visualization

Developing effective visualization tools for frequent pattern mining has been the subject of many studies. This line of research can be sub-classified into two general categories: systems for visualizing data and those for visualizing the mining results.

2.2.1 Data Visualization

It is not surprising that data visualization has a long history [Fri05]. Many visualization techniques have been discovered. For instance, iconic displays [PG88] map values of multidimensional data to different features of an icon. Dense Pixel [Kei00] displays map data to a coloured pixel or group of pixels on the screen. Stacked displays [LWW90] present data that has a hierarchical structure. Geometrically transformed displays [Kei02] find interesting transformations for mapping multi-dimensional data

to some position on the screen space. Examples of geometrically transformed displays include scatterplot matrices, projection views, and the parallel coordinate graph. Our display is also inspired by geometrically transformed displays.

One common form of visualization is the parallel coordinates [ID90], which is quite effective in visualizing high-dimensional data. In the parallel coordinate system, the displaying axes of variables are rearranged such that they are parallel (instead of orthogonal) to each other. This is depicted in Figure 2.1. In the figure, three objects are mapped along two dimensions: the cost of the object (x -direction) and the quantity purchased (y -direction). The parallel coordinate display is constructed by rearranging the two axes in parallel, and projecting the objects onto each axis. Each object is then connected to itself along each of the axes. Hence, each line corresponds to one object. Parallel coordinates have been applied to many contexts [FWR99, HLD02] and constitute the base representation for many existing practical visualization tools.

Research done by Mackinlay [Mac86] in Figure 2.1 shows perceptual tasks ranked based on the type of data being displayed. The low ranking tasks (in grey) are not relevant to those data types. In the case of itemsets, the most important information to display is items contained and the frequency. Since screen position is ranked best for all types of data, a mapping should be found to map both the nominal data of items contained and the quantitative data of frequency to screen position. This fits nicely with the two dimensions that make up the screen space.

The parallel coordinates graph [ID90] is an example for visualizing multidimensional data. Typically, this graph represents each dimension by a vertical axis and

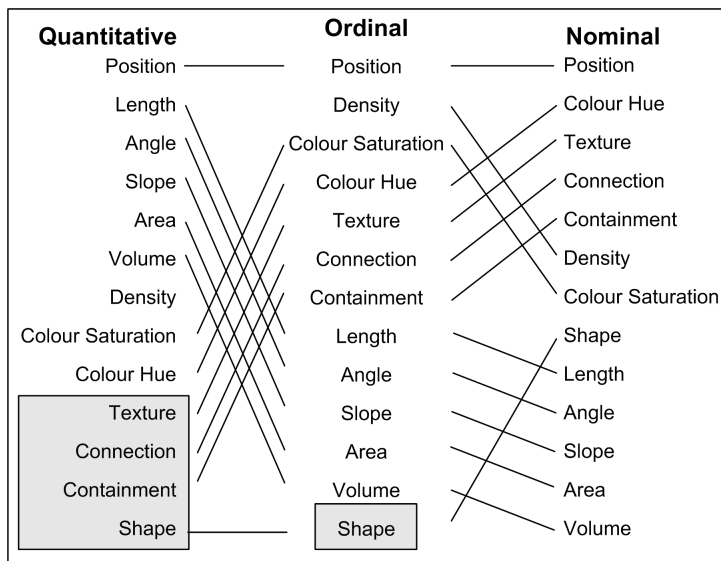


Figure 2.1: Mackinlay's ranked perceptual tasks.

each data element by a polyline. The point at which the polyline crosses each axis determines the value of that data dimension. Figure 2.2 shows an example. In this figure a single data element is displayed. This data element is composed of five dimensions all having values that range from -4 to 4 . Reading the positions that intersect with each axis, marked with red circles, from left to right we can determine the exact values contained in the element, i.e., $(D1:4, D2:-2, D3:1, D4:3, D5:0)$. This technique is quite flexible as each dimension can have independent ranges. If $D2$ ranged from 0 to 100 , its range could be changed while the scale displayed on all the other dimensional axes remained the same. This graph is designed to display data elements that have a value for each dimension. For example, if the data element described above was missing a value for $D3$, how would it be displayed? This is one of the problems that had to be overcome when displaying itemsets since they vary in length.

Examples of some specific data visualization systems include VisDB [KK94], Spot-

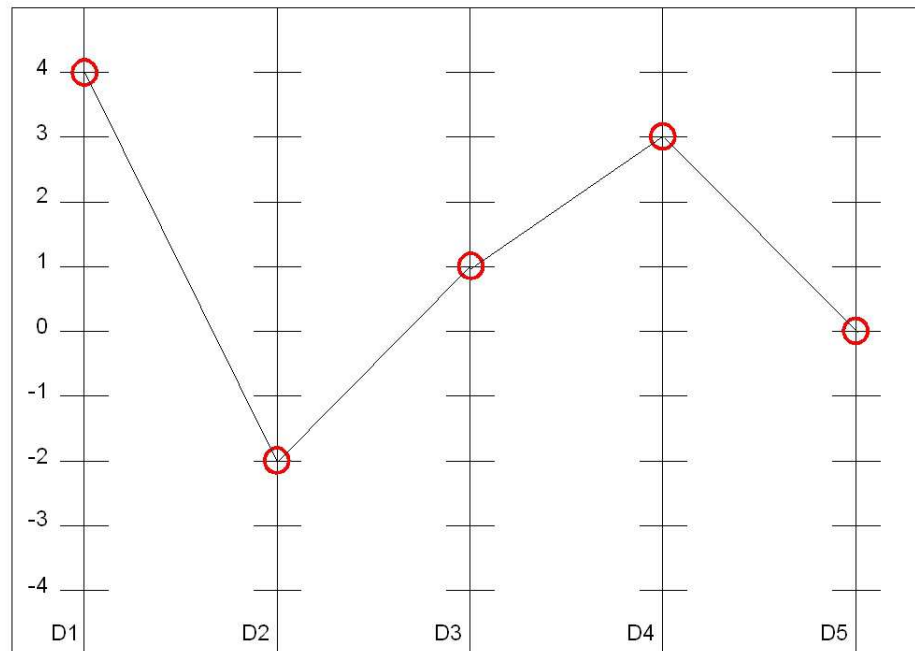


Figure 2.2: A parallel coordinate plot of datum $(4,-2,1,3,0)$.

fire [Ahl96], independence diagrams [BJR00], and Polaris [STH02]. These systems provide features to arrange and display data in various forms. For example, VisDB provides *parallel coordinates*, pixel-oriented techniques, and stick figures to users for exploring large datasets; Polaris provides a visual interface to help users formulate complex queries against a multi-dimensional data cube. However, these systems are not connected to any data mining engine, nor are they designed to display data mining results.

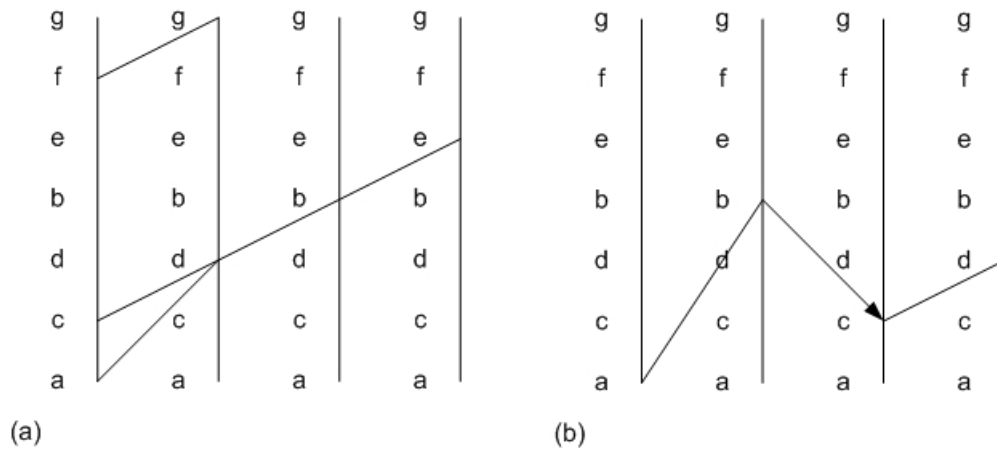


Figure 2.3: Yang's system displays itemsets and association rules on parallel coordinates.

2.2.2 Result Visualization

Yang [Yan05] created a system that uses the parallel coordinates to display itemsets and association rules. In this system he considers cardinality and items contained as the most important information to display. The vertical increments on each axis represent each item in the set. The length of the line indicates the cardinality of the set. Figure 2.3 shows a simplification of this system. On the left itemsets $\{f, g\}$, $\{c, d, b, e\}$ and $\{a, c\}$ are displayed. Displayed on the right is the association rule $\{a, b, c\} \Rightarrow \{d\}$. In this system, the thickness of the line is used to determine the itemset frequency. In case of association rules, thickness is used to indicate confidence. Line thickness may be adequate for determining which itemset occurs more frequently provided there is a big enough difference between the two, but it is often difficult to determine how much more frequent one itemset is from the other.

In Wong's system [WCF⁺00], another parallel coordinated system for visualizing sequential frequent patterns, line colour is used to show support of sequential patterns

over time. Here, we have the same problem. Again the perceptual task of comparing the colour of two lines is good enough to inform the user on which frequency is greater than the other. However, it is poor when determining the exact frequency or the difference between two frequencies.

Munzner et al. [MKN⁺05] presented a visualizer called PowerSetViewer (PSV), which provides users with guaranteed visibility of frequent patterns in the sense that the pixel representing a frequent pattern is guaranteed to be visible by highlighting such a pixel. However, PSV does not explicitly show the relationship between related patterns (e.g., patterns {apples, bananas} and {apples, bananas, cherries} are related—the former is a subset of the latter). Moreover, multiple frequent patterns may be represented by the same pixel in PSV. It is not easy to distinguish one from others.

As a preview, in the next chapter, we will modify the parallel coordinate graph for the specific purpose of displaying items contained and frequency. Different from these other two systems, in my system items run along the horizontal and the frequency is displayed along the vertical. One item is assigned to each vertical axis and the height of the data displayed determines the frequency. The problem of varying dimensional lengths was overcome by the introduction of glyphs. If a data element (itemset) contains an item, a glyph is put on the graph where that data element crosses the item axis. If that itemset does not contain that item, no glyph is used.

Displaying data elements of variable length also leads to the other problem of determining exactly which itemsets are displayed. In Figure 2.3, the two overlapping itemsets displayed are {*c, d, b, e*} and {*a, c*}. However, if they were {*c, d*} and

$\{a, c, b, e\}$, then the image would look exactly the same. Yang addressed this problem by introducing curved polynomial lines that are always perpendicular to the axis when they cross it. Our system also had to overcome this problem. We solved the problem with the introduction of a second modification of the parallelized coordinated graph. It focuses on displaying exactly which itemsets are displayed in some limited context like a mouse selection. This frees the first graph to handle the overview of a million or more itemset that can be visually compared by frequency, while the second graph can provide details as the user demands them.

2.3 Summary

The visualization system presented in this thesis is used to display both raw transaction data and itemsets. Transactions are the sets of items that occur in the original database. Their count describes the number of times that set of items has occurred together with no other items. Given a transaction database, data mining algorithms efficiently count how many times a combination of items have occurred together. Since the number of possible combinations can be very large, constraints are provided to restrict the number of itemsets produced. The basic constraint of minimum support was provided right from the beginning, but is not sufficient for some tasks as users cannot use the constraint to limit the cardinality of the resulting itemsets.

Using Mackinlay's ranked perceptual tasks, finding the right visualization for itemsets was achieved by mapping the most important data presented to position. Yang's system also uses the parallel coordinate system. However, frequency is mapped to line

thickness, which makes it difficult to compare how much more frequent one itemset is over the other. The parallel coordinate system was designed to display multidimensional data, but length of each transaction may vary. It is not easy to determine the maximum dimensions of transactions. To overcome this problem, glyphs are introduced to convey that an itemset data element contains a value for that vertical axis. If no glyph is contained on an item axis, then that item is assumed not to be in the set.

Moreover, when data elements overlap, it is difficult to tell where one stops and the other one continues. This is another consequence of variable length data being displayed on the parallel coordinate graph. The solution is to display itemsets with two graphs. One concentrates on displaying an overview of millions of itemsets while the other displays exactly which itemsets exist within a given detail on demand context like a mouse selection.

Chapter 3

Transaction and Itemset

Visualization





In this chapter, we present our system for visualizing transactions and itemsets. The design was inspired by the parallel coordinate graph. To show data of multiple dimensions, we first introduce glyphs, which are simple images that convey information on (a) where itemsets start or end, and (b) which items are contained.

We then discuss two graphs used in our visualization system: Frequency graph (FGraph) and Existence graph (EGraph). The FGraph displays millions of itemsets, their frequencies, and their prefixes as well as superset extension. As a complement, the EGraph also uses the glyph representation to show exactly which itemsets exist within a limited context.

Finally, we discuss interactive features that provide dialogue between the user and the display.

3.1 Basic Representation

A glyph representation displays itemsets as a path on a parallel coordinate graph. Nodes represent the first, intermediate, and last items within an itemset and/or in a transaction. Edges between nodes indicate those items are contained within the same itemset or transaction. Each node consists of one or more glyphs placed on an x -axis. The x -position of the node indicates which item is contained. The type of glyph indicates whether that node represents the first, the intermediate, or the last item within an itemset or transaction. These glyphs are shown below:

1.  indicates the first (leftmost) item in an itemset,
2.  indicates an intermediate item contained in an itemset, and
3.  indicates the last (rightmost) item in an itemset.
4.  indicates the last (rightmost) item in a transaction.





A combination of nodes and edges represents an itemset. A *singleton itemset* is represented by unconnected left and right glyphs, , as it starts and ends with the same item. A *2-itemset* (i.e., itemset consisting of 2 items) is represented by connected left and right glyphs . A *3-itemset* (i.e., itemset consisting of 3 items) is represented by a left, an intermediate, and a right glyph . Any itemset that ends with a  indicates it is the last item within an itemset and that itemset is also a transaction in the database.

Table 3.1: Raw transaction database before data mining

Transaction ID	Content	Transaction ID	Content
1	{a}	7	{a}
2	{b,c}	8	{a, b, c}
3	{a,c}	9	{b,c}
4	{a,b,c}	10	{b,c}
5	{b}	11	{a,b}
6	{a,c}	12	{a, b, c}

3.1.1 Visualizing Transactions

Visualizing transactions allow users to (a) visualize a raw transaction database before any mining takes place and (b) consider special association rules such as $\{a,b\} \Rightarrow \{a,b,\emptyset\}$. This rule compares the likelihood of $\{a,b\}$ occurring alone, given that items a and b have occurred together.

Internally, displaying transaction information adds some complexity to the underlying data structure. Specifically, we store the transaction data in a tree structure, in which each tree path represents a transaction. Each tree node represents an item in a transaction. We also associate the transaction count with each node.

Table 3.1 shows a simple database of 12 transactions. Our visualization system displays the contents of these transactions. Figure 3.1 shows the glyph representation of the contents of these transactions using three different visual compression techniques. Figure 3.1(a) shows one transaction per line with no visual compression. Each line clearly indicates each transaction and all items within it. Figure 3.1(b) shows an extreme vertical compression, where all transactions are compressed horizontally into one line. The final compression technique, as shown in Figure 3.1(c), enable users to vertically compress these transactions. Note that there is only one interpretation as to which transactions are represented.

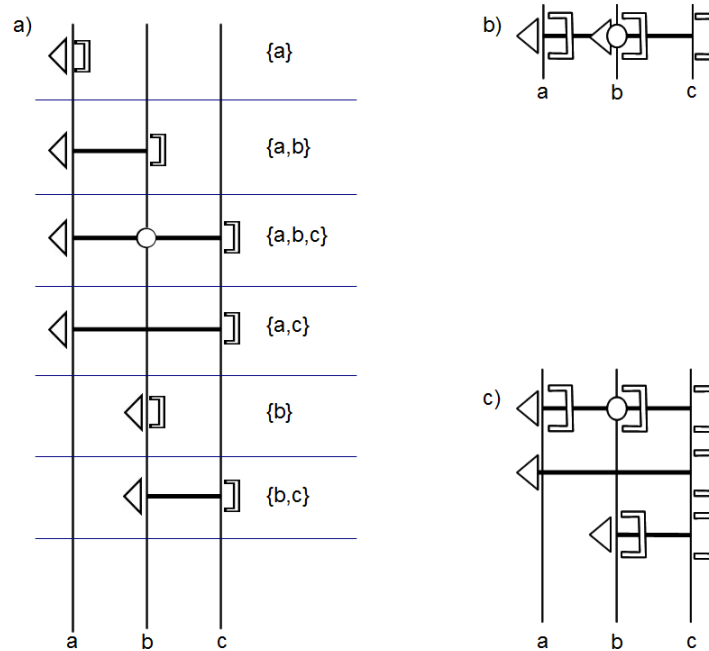


Figure 3.1: Basic glyph representation of transactions.

3.1.2 Visualizing Mined Itemsets

Seven itemsets mined from the database of 12 transactions in Table 3.1 are shown in Table 3.2. Figure 3.2 shows the glyph representation of these itemsets using three different visual compression techniques. Figure 3.2(a) shows one itemset per line with no visual compression. Each line clearly indicates each itemset and all items within it.

Table 3.2: Mined itemsets from Table3.1

Itemset in the form of $\{items\}:frequency$
{a}:8, {a, b}:4, {a, b, c}:3
{a, c}:5
{b}:8, {b, c}:6
{c}:8

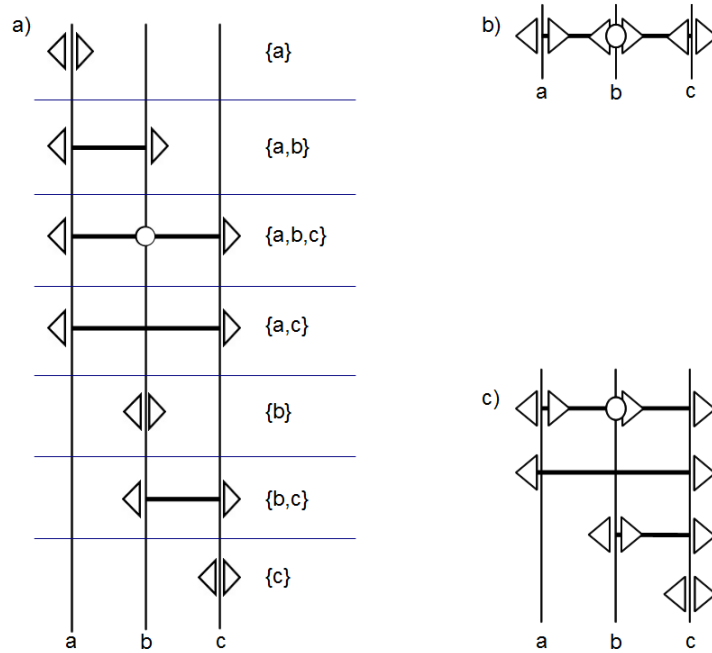


Figure 3.2: Basic glyph representation of itemsets.

Figure 3.2(b) shows an extreme vertical compression, where all itemsets are displayed on the same line. This graph captures many different combinations of itemsets by compressing them horizontally into one line. The final compression technique, as shown in Figure 3.2(c), enable users to vertically compress these same itemsets. Note that there is only one interpretation as to which itemsets are represented.

3.1.3 Our System for Visualizing Transactions and Mined Itemsets

Observant readers may notice that each transaction (i.e., input for the data mining engine) consists of one or more items and each mined itemset (i.e., output from the data mining engine) also consists of one or more items. The problem of visualizing transactions and the problem of visualizing itemsets can be very similar. As such, we develop a visualization system that can display both transactions and itemsets.

Our system uses two graphs. Both of them use the glyph representation as described above. The first graph, the *Frequency Graph (FGraph)*, provides an overview and is capable of displaying millions of itemsets using a visual compression technique based on itemset frequency. This visual compression technique can be ambiguous as lines may overlap. To solve this problem, we use the second graph, the *Existence Graph (EGraph)*, as an unambiguous extension to describe exactly which itemsets exist within a given context.

3.2 Frequency Graph (FGraph)

The FGraph uses the same representation shown in Figure 3.1 (c) and 3.2 (c), except nodes are placed vertically depending on the frequency of the itemset or transaction they represent. This visual compression technique can display any mined database composed of n items and m transaction in an $n \times m$ visual space. Figure 3.3(a) shows how the itemsets in Table 3.2 are displayed in the FGraph. For each of the seven displayed itemsets, their frequency can be determined by the height of

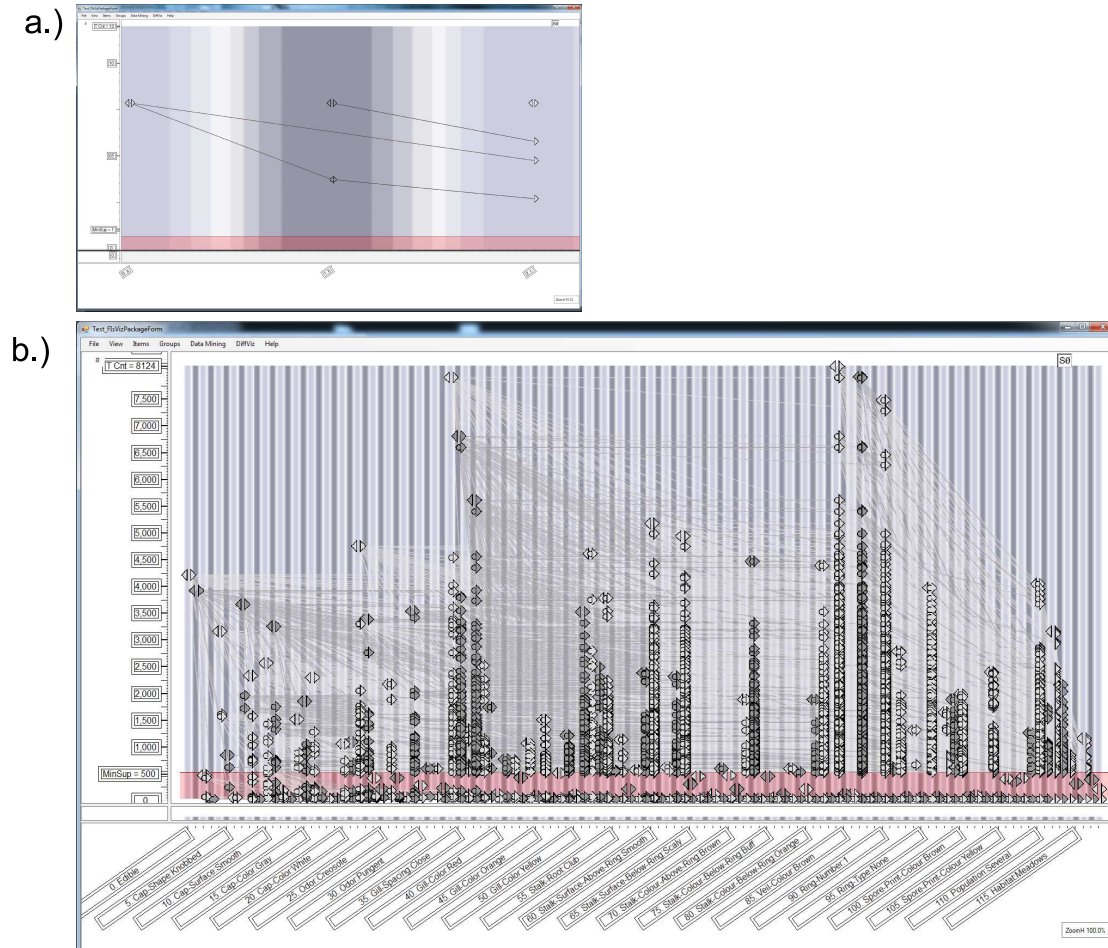


Figure 3.3: Itemsets from Table 3.2 and from UC Irvine Mushroom Data.

the node and the items contained in each can be determined by the connected edges. This graph is also scalable. Figure 3.3(b) shows an image of 1.4 million itemsets mined from the UC Irvine mushroom dataset [FA13] mined with a minimum support of 500.

Recall that itemsets are the focus of our visualization system because they contain

all of the association rule information. By using position to show frequency, users can look at any two nodes to determine their frequencies as well as the proportional difference between them. If one of the nodes represents a superset of the other, this proportional difference allows the user to extract the confidence and support of the association rule that links them. It is also interesting to note that, by the nature of this representation, a superset is always on the right or below its subset. Hence, nodes on the left logically represent the antecedent and nodes on the right represent the consequence of the connecting rule. Figure 3.4 shows that, by comparing the heights of two itemsets, the users get a sense of the confidence of the association rule that connects them. Figure 3.4(a) shows that by comparing the heights of the nodes one can determine that $\{\text{Apples}\} \Rightarrow \{\text{Cherries}\}$ has a confidence of $7/8$ or 87.5%. Figure 3.4(b) shows $\{\text{Apples}\} \Rightarrow \{\text{Bananas, Cherries}\}$ has a confidence of $4/8$ or 50%. The figure also shows that if both nodes have the same frequency (y position) the confidence of the rule they form will be 100%, Figure 3.4(c).

In addition to specific itemset support information, the FGGraph also provides some corollary support information. When considering how many people who bought apples also bought bananas, $\{a\} \Rightarrow \{a, b\}$, many other questions may come to mind. Examples include “how many people bought apples?”, “how many people bought bananas?”, “out of the people who bought apples, how many did not buy bananas?”, and “out of the people that did not buy apples, how many bought bananas?”. We use the programming “not” symbol (i.e., “!”) to indicate those items that are *not* contained in a transaction. When considering the rule $\{a\} \Rightarrow \{a, b\}$, the following are seven related rules:

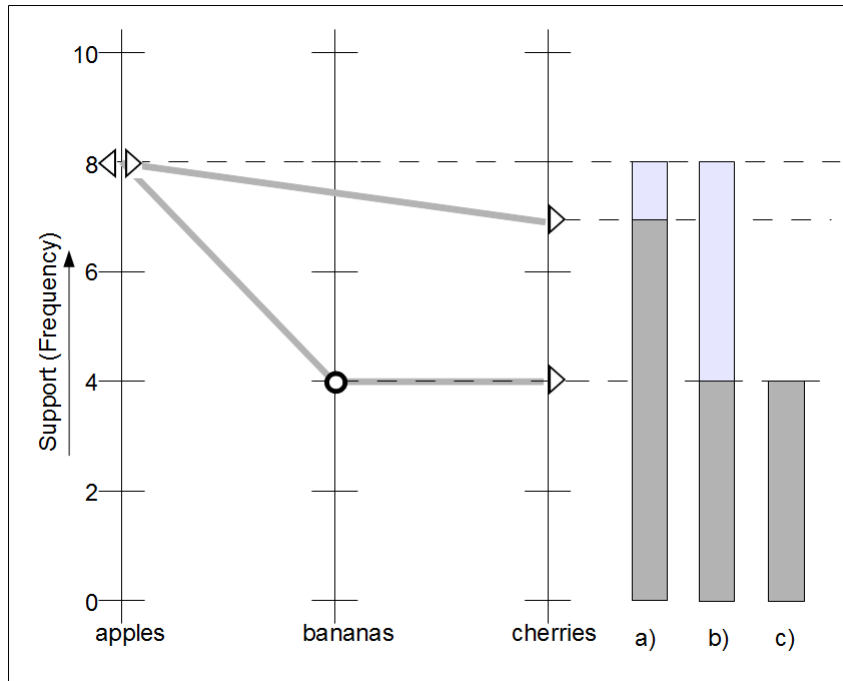


Figure 3.4: FGraph can imply association rule support and confidence.

1. $\{a\} \Rightarrow \{a, !b\}$,
2. $\{!a\} \Rightarrow \{!a, b\}$,
3. $\{!a\} \Rightarrow \{!a, !b\}$,
4. $\{b\} \Rightarrow \{a, b\}$,
5. $\{b\} \Rightarrow \{!a, b\}$,
6. $\{!b\} \Rightarrow \{a, !b\}$, and
7. $\{!b\} \Rightarrow \{!a, !b\}$.

To form these rules, the supports of the following itemsets are needed and they are listed as follows:

1. $\{a\}$,
2. $\{a, b\}$,
3. $\{a, !b\}$,
4. $\{!a\}$,
5. $\{!a, b\}$,
6. $\{!a, !b\}$,
7. $\{b\}$, and
8. $\{!b\}$.

The FGraph displays the support information of seven itemsets in Figure 3.5.

3.2.1 Item Order

Having the ability to order items helps the user to examine the mined itemsets. Moving an item to the left reduces the number of nodes on that column. This makes it easier to see how that item interacts with the remaining items. Figure 3.6 shows how different item order changes the context of the display. Both images in this figure show only 2-itemsets about an edible red mushroom (i.e., with the edible attribute and coloured red). In Figure 3.6(a), the edible attribute is on the left. In Figure 3.6(b), it is moved to the right. Between them, Figure 3.6(a) gives clear indication of both

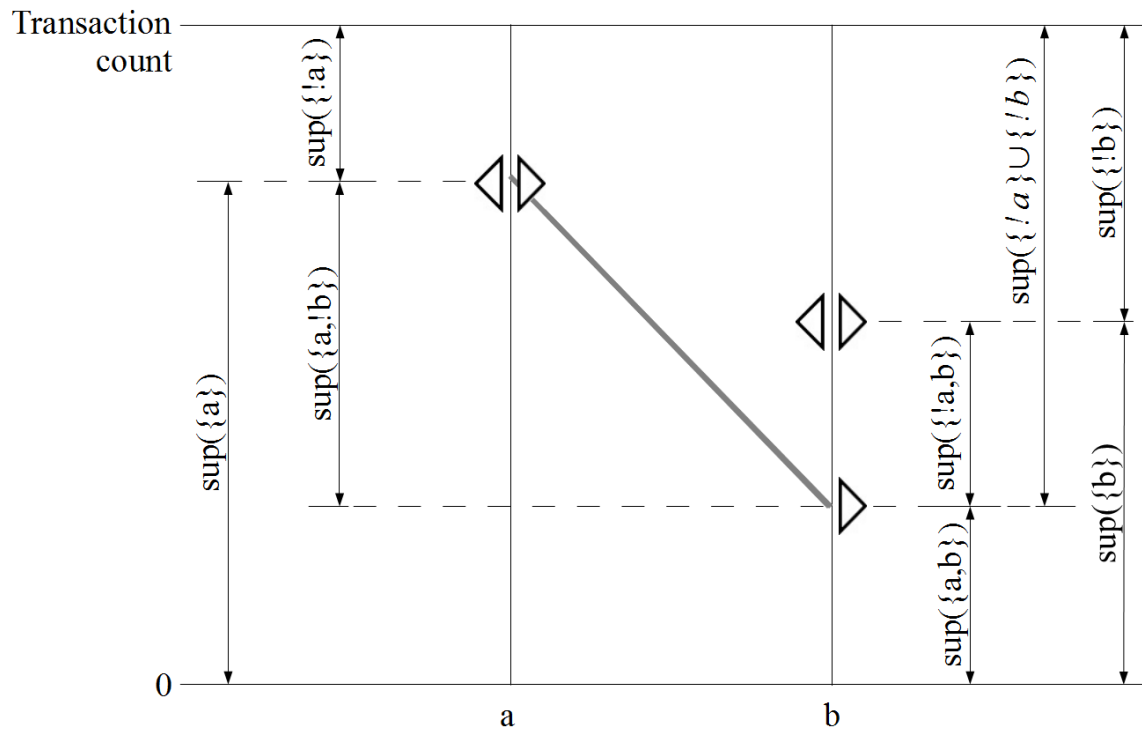


Figure 3.5: Corollary support information

the items that occur with the edible attribute and their frequencies. In Figure 3.6(b), the items that occur with the edible attribute are still clearly indicated, but their frequencies are more difficult to read. To determine the frequency of any 2-itemsets containing edible in this figure, the user needs to follow the edge from the start glyph to the edible column. These two images display the same itemsets but in two different contexts. Having the ability to order items helps the user to pick the desirable one between the two orderings. This allows the user to explore the data and/or the mined results.

3.2.2 Prefix/Extension Paradigm

Our visualization system is based on the prefix/extension relationship. This relationship helps to limit the number of connections between itemsets depending on the item order. For example, on the one hand, if the user is interested in all extensions of a , he can order item a before b . In this case, itemsets $\{a\}$ and $\{a, b\}$ are related as $\{a\}$ is a prefix of $\{a, b\}$. On the other hand, if the user is interested in all extensions of b , he can order item b before a .

Our system simplifies both the relationship between itemsets and the underlying data structures. Specifically, many superset-subset relationships are reduced into a smaller number of prefix-extension relationships.

The simpler extension-prefix relationship is defined as: Given items $i \in I$, order R and itemsets $P \subset E \subseteq I$, E is an extension of prefix P if and only if $E = P \cup i$, where \forall items $p \in P$, $\text{order}(p) < \text{order}(i)$. For example, in a domain of four items (a, b, c, d and e), given a 3-itemset $\{a, c, e\}$, there are many 2-item subsets of $\{a, c, e\}$, namely $\{a, c\}$, $\{a, e\}$ and $\{c, e\}$. Similarly, there are many 4-item supersets of $\{a, b, c, e\}$ and $\{a, c, d, e\}$. On the other hand, there is only one 2-item prefix of $\{a, c, e\}$, namely $\{a, c\}$. Note that both $\{a, e\}$ and $\{c, e\}$ are subset but not prefix of $\{a, c, e\}$. There is also only one 4-item extension of $\{a, c, e\}$, namely $\{a, c, e, d\}$.

Figure 3.7 shows a comparison between supersets and extensions. The number of itemsets remains the same. The relationship between itemsets has changed.

The user is not limited to looking only at prefixes and extensions. Figure 3.8 shows an example of how the subsets and supersets are displayed. The itemset $\{1, 22, 26, 34, 37, 51\}$ was chosen at random and coloured blue. The subsets of this

itemset are coloured green. Notice all of their end glyphs are to the left (or on the same item) and above (or at the same frequency) of the blue end-glyph item 51 with the label “Stalk Shape:Enlarging”. All of the supersets are coloured red. Due to the Apriori property [AS94], the frequencies of all supersets/extensions of a pattern X must not be higher than the frequency of X . Equivalently, the frequencies of all subsets/prefix of a pattern X must not be lower than the frequency of X . So, all supersets are shown on the same level or below X in the y -direction.

The prefix/extension itemset relationship is directly dependent on the order chosen. To better display the subsets and supersets related to the itemset $\{1, 22, 26, 34, 37, 51\}$ a specific order for this task can be chosen. Compare Figure 3.8 to Figure 3.9 where itemsets are ordered with all $\{1, 22, 26, 34, 37, 51\}$ items to the left hand side. Notice how all subsets and supersets grouped appropriately. In this ordered image, it is much easier to see which items occur with itemset $\{1, 22, 26, 34, 37, 51\}$ and which do not.

The FGraph is good for providing an overview as it can display millions of transactions and itemsets at once. When displaying millions of transactions or itemsets, users cannot be expected to trace edges and understand exactly which items belong to which itemsets, see Figure 3.3(b) for example. The FGraph was not designed to be stand-alone nor is it designed to be static. It was designed to work with interactive features to provide a general overview and to display the details of cases where itemsets are not excessively overlapping. In the cases where overlapping is a problem second graph is used.

3.3 Existence Graph (EGraph)

The Existence Graph complements the FGraph by providing a representation showing that itemsets exist within a given context (e.g., itemsets selected with a mouse). The EGraph only shows whether or not an itemset exists. It does not display itemset frequency information.

The EGraph uses the same glyph representation as shown in Figure 3.1(c) and Figure 3.2(c) with the addition of vertical lines that help the user to identify itemsets having the same prefix. A node on the FGraph may represent an item that is contained in a thousand different itemsets all ending with that item and all having the same frequency. Interactive features together with the EGraph allow the user to brush, filter, highlight and examine exactly which itemsets pass through a given node. Users can examine any of the nodes displayed on the FGraph to see exactly which itemsets are represented. Figure 3.10 shows how itemsets highlighted in yellow on the FGraph are also highlighted in yellow on the EGraph. In addition, when the users move over any node in the EGraph, that node's path to the start item is highlighted in red on both graphs. These interactive highlighting features are designed to link the two graphs in the users' mind. The yellow highlighted itemsets in the FGraph are the same yellow itemsets displayed in the EGraph. The red highlighted itemset in the FGraph is the same red itemset in the EGraph. The highlighted portions of both graphs give a complete picture of itemsets (and their frequencies) of user interest. Typically, the user selects one or more nodes on the FGraph, he then moves the mouse to the EGraph to highlight interesting individual itemsets. Item labels and columns are also highlighted with the same colour to help the user to identify items

and frequencies.

3.4 Interactive Features

Interactive features provide a communication channel between the user and the display. The user is given several options to describe what they would like to see. These include four main sub-options: (1) itemset selection, (2) item editing, (3) itemset group editing and (4) data mining. In this section, we look at the different features to get an understanding of what is available to the user when exploring itemsets. The usefulness of these features is evaluated in Chapter 4 where we apply our system to three specific cases with the real-life databases and scenarios.

3.4.1 Itemset Selection

Itemset selection is a basic command for brushing, filtering, and finding details on demand. The user can select one or more itemsets (or transactions) of interest.

Specifically, users can select items by a mouse drawn rectangle on the FGraph. The itemsets and transactions represented by the end nodes contained within that rectangle are then highlighted. If a second selection is made by holding the control key down, the user has the option of applying one of the set operators: Union, intersection, difference, or complement to the two selections.

The extension interface is shown in Figure 3.11. This dialogue gives the user an option to add previously mined extensions to their current selection. The interface presents the user with a single line glyph representation of all the extensions of a

mined itemset.

The filtering dialogue is shown in Figure 3.12. From top to bottom, users are presented with the options to filter by the following:

1. cardinality (e.g., only display 2-itemsets or 3-itemsets),
2. minimum support (e.g., only display itemsets occur at least 10 times), and
3. user-specified constraints (e.g., only display some specific item ID numbers, frequency ranges, as well as prefix frequencies).

Moreover, the user is given the option to define a filtering string for some more complex criteria. Our visualization system also provides a list of functions for users to choose. The following are some examples:

1. If the user specifies the equation “ $\text{ifreq}() \geq 1000$ ”, then our system displays all the itemsets with a frequency greater than or equal to 1000.
2. If the user specifies the function “ $\text{pif}(1) \geq 1200$ ”, then our system goes through each k -itemset X , find all its $(k - 1)$ -item prefix of X , and displays such a $(k - 1)$ -itemset if its frequency is at least 1200.
3. If the user specifies the equation “ $\text{card}() == 10$ ”, then our systems displays all the 10-itemsets.
4. If the user specifies the function “ $\text{hall}(a,b,c)$ ”, then our systems displays all itemsets that contain a , b and c .
5. If the user specifies the function “ $\text{hany}(a,b,c)$ ”, then our systems displays all itemsets that contain a , b or c .

6. If the user specifies the equation “ $\text{hmany}(a,b,c) == 2$ ”, then our systems displays all itemsets that contain 2 of the specified items (i.e., any two of a, b and c , in other words, all supersets of $\{a, b\}$, $\{a, c\}$ and $\{b, c\}$).

While the above functions deal with itemset visualization, the following functions deal with transaction visualization:

1. If the user specifies the equation “ $\text{tcnt}() \geq 10$ ”, then our system displays all the transactions that appear repeatedly at least 10 times.
2. If the user specifies the function “ $\text{isTrans}()$ ”, then our system displays all transactions that appear at least once (i.e., all transactions in the database).
3. If the user specifies the function “ $\text{ptc}(1) \geq 12$ ”, then our system goes through each transactions t_i consisting of k items, find all transactions t_j containing only the first $k - 1$ items in t_i , and displays t_j if its transaction count is at least 12 (i.e., if t_j is repeatedly appear at least 12 times).
4. If the user specifies the equation “ $\text{isTrans}() \ \&\& \ \text{card}() == 10$ ”, then our system displays all the transactions consisting of exactly 10 items.
5. If the user specifies the function “ $\text{isTrans}() \ \&\& \ \text{hall}(a,b,c)$ ”, then our system displays all transactions that contain a, b and c .
6. If the user specifies the function “ $\text{isTrans}() \ \&\& \ \text{hany}(a,b,c)$ ”, then our system displays all transactions that contain a, b or c .
7. If the user specifies the equation “ $\text{isTrans}() \ \&\& \ \text{hmany}(a,b,c) == 2$ ”, then our systems displays all transactions that contain 2 of the specified items (i.e., any

two of a, b and c).

Furthermore, users can express any logical combinations of these functions and/or equations. An example is the equation “!(ifreq() \geq 1000 && pif(1) == ifreq())”, which displays all itemsets but those with a frequency greater than or equal to 1000 and has a first prefix with the same frequency.

3.4.2 Itemset Group Editor

Once the user has selected a collection of transactions or a set of itemsets (i.e., itemset group), he can take a union of two selections, create a new selection, or replace an existing selection. The purpose of the itemset group is to have a subset of itemsets stand out by adjusting their visual properties such as node size, node colour, edge size, column colour, and colour weights.

Our visualization system displays the groups using colour blending that is similar to a Venn diagram. It does so with an “alpha red green blue” (ARGB) value, which is an RGB triple with an alpha transparency value, for the nodes, edges, columns representing the group. Here a blend value $\in [0,1]$ is used to set an arithmetic weight that blends colours as one group overlaps another. Inspired by the Venn diagram, which overlaps semi-transparent discs, representing different subsets, our system identifies members that belong to group A or group B or both. The blend colour works in the same fashion as these disks. For example, if an itemset is contained in both groups A and B, the final glyph colour is calculated by the following equation:

$$\begin{aligned}
EndGlyphColour &= (1 - GroupBWeight) \times & (3.1) \\
&(ItemNodeColour \times (1 - GroupAWeight) \\
&+(GroupAWeight \times GroupAColour)) \\
&+(GroupBWeight \times GroupBColour),
\end{aligned}$$

where each colour is a 4-dimensional ARGB value.

A similar result could be achieved simply by using the alpha value. However, using an external weight allows the user to override another overlapped group's colour with a completely transparent colour. For example, if the user wants to visualize only those itemsets that belong to group A but not group B, he can set the colour of group B to transparent and group B's colour weight to 1. All of group B's end-nodes then become transparent, leaving only those itemsets contained in group A and not contained in group B visible.

The end-glyph node was used in the above example because it is the simplest colour to set. The start and intermediate glyph nodes and their colours are more complicated because their colours depend on the group colour of every extension. Consider the two itemsets, $\{a, b\}$ which is in group A only and $\{a, c\}$ which is in group B only. What colour should the start glyph on item column a be? This glyph is a partial representation for both of these itemsets. So, its colour should reflect the colours of both groups. Assuming no other extension is coloured, its final colour will be calculated with the same equation as above.

The user also has the option of only showing those itemsets that are grouped. The purpose of doing this is that of brushing away all the itemsets that the user is not

interested in.

3.4.3 Data Mining

Quick-mine is a feature that mines all the immediate extensions of the currently selected itemsets by appending only one more item. If no selection is made, then Quick-mine mines the single items and appends each of them to current itemset collection. For more options the users can explicitly state which items to mine using the data mining interface shown in Figure 3.14.

3.4.4 Item Editor

The item editor is used to colour items, edit item strings, and set the way they are ordered. Eight columns are used to display all the information associated with each item. These are (1) Item ID, (2) Natural Index, (3) Name, (4) Description, (5) Frequency, (6) Colour, (7) Column Colour, whether or not the item is displayed, and (8) Aggregate String.

The third and fourth columns list the names and description of each item. Both of these are fully editable. The item name is used for the item labels at the bottom of every graph image. The description is used for user notes and possible future requirements like the addition of pop up information when the user hovers over an item label.

The item frequency column is provided for user convenience. This information helps the user to determine which items are more important than others, which items to display, and how they should be ordered.

Each item is assigned a colour so that it can be easily recognized. The user has the option of changing the default colour by using a few different methods. Individual item colours can be changed by clicking the colour boxes and making a selection, or several items can be coloured at the same time by highlighting them and then choosing a colour. The items within a category can be coloured the same. This helps the user recognize that they belong to the same category.

The next column displays the colour of the column in the graph. This is a useful tool for rearranging item orders based on coloured groups. The colour of the group affects the colour of the column, so in this form users can see exactly which items are contained in each group and make a decision on item order based on this information.

The next column determines whether or not the item will be displayed on the graph. This value can be changed independently by clicking individual display boxes or by highlighting several rows and hitting the space bar. If the item display value is turned off no itemset containing that item will be shown. The effect is the same as if that item was never included in the mining process. However, even though removing items from the display does not change any of the visible itemset frequencies, it may change some of the individual transaction counts. Consider the mined database results in Tables 3.2. If items b and c are no longer displayed, the only remaining itemset is $\{a\}:8$. However, the frequency of the remaining transaction has increased from $\{a\}:2$ to $\{a\}:8$ because the items that are removed are also removed from the mining process. Transactions like $\{a, b\}$ become $\{a\}$.

The item editor is also used to edit the item order. Order can be changed using two different methods. Clicking the top of any column will order the items based on

that value. The user can also select one or more rows and by using the up and down buttons, items are move up and down in the order.

This item editor also has the ability to add *Aggregate items*. These are items added to the transactions for user convenience. They are the aggregate inclusion or exclusion of items already contained in the database. For example, looking at a grocery store transaction data base, it may be interesting to examine the number of times milk, cheese, or sour cream has been purchased with beer. The user may want to organize taxonomy of items and consider the three items as dairy products. The question becomes, how many times has a dairy product been purchased with beer? The items milk, cheese, and sour cream would be aggregated into a single item called “dairy”.

For the mushroom data example, all mushrooms that have either a green or purple cap colour are edible. With that knowledge, the user can compare those attributes that occur with these coloured caps to those mushrooms that do not. To do this, the user creates two attribute items, a “purple/green cap colour” item and a “non purple/green cap colour” item. These items are added to the appropriate transactions allowing the user to explore and compare the extensions of each. This is also another example of why the extension mining algorithm allows the users to focus their mining efforts. Since every non-green/non-purple cap item will appear in every mushroom transaction that has a red cap, there is no need to mine these two items together. By using a limited number of extension items in mining algorithm, the user can choose not to mine into any of the cap colour items.

Once the mining is completed, the resulting itemsets naturally include a com-

bined frequency count of all the included cap colours. This is a very handy feature when trying to determine how many non-green/non-purple capped mushrooms are edible and have no odour. In this case the $\text{sup}(\{\text{edible, non-green/non-purple cap, no odour}\}) = \text{sup}(\{\text{edible, red cap, no odour}\}) + \text{sup}(\{\text{edible, cinnamon cap, no odour}\}) + \dots + \text{sup}(\{\text{edible, yellow cap, no odour}\})$ because no mushroom has more than one cap colour. However, if we look back at the dairy example, $\text{sup}(\{\text{beer, dairy}\}) \neq \text{sup}(\{\text{beer, milk}\}) + \text{sup}(\{\text{beer, cheese}\}) + \text{sup}(\{\text{beer, sour cream}\})$ since many transactions could contain both cheese and sour cream.

Our visualization system inserts the aggregate items into the database by looking at each transaction and determining if the new aggregate item should be added to it. A text space is provided to enter an equation string that chooses which transaction will receive the new aggregate item? For the above example, the transactions for the “purple/green cap colour” aggregate item are identified by the equation “ $\text{hany}(18, 19)$ ”, where 18 and 19 are the item numbers for green and purple caps respectively. This works because the $\text{hany}(18, 19)$ function returns true if a transaction “has any” of the item ID numbers provided in the arguments. The equation for the “non purple/green cap colour” aggregate item is $!\text{hany}(18, 19)$, where “!” is again the logical negation operator. These include $\text{hall}()$ —“has all” and $\text{hmany}()$ —“how many” which return the number of arguments contained in each itemset. These are the same functions used for itemset filtering. So, many many different equations are possible. They can even include transaction counts and cardinalities. Using the $\text{hmany}()$ function, a user can add an aggregate item to transactions that have any 5 out of 10 items. If a transaction has more than 5 sugar snacks it can be labelled as “unhealthy”. The

user would now be able to consider how many “unhealthy” transactions also contain beer.

3.5 Summary

In this chapter, we introduced a general itemset representation using a modified version of the parallel coordinate that includes glyphs. Four glyphs are used to determine which items start, are contained in, and end a transaction or itemset. Two graphs, both of which use this representation, are used together to provide a complete picture, where one graph provides an overview of millions of itemsets, and the other gives the details of which itemsets are displayed within a given context.

Our visualization system consists of two graphs. The frequency graph (FGraph) displays the frequency of (a) transactions (which represent the number of times a transaction appears repeatedly) or (b) itemsets (which represent the number of times an itemset is mined from the transaction dataset). Specifically, the FGraph uses polylines to represent the prefix-extension relationship of transactions and/or itemsets along the x -axis. Each polyline represents a transaction or an itemset. The y -position of nodes on the polyline shows the frequencies of the prefix of such a transaction or itemset. As polylines bend, two or more polylines may meet at the same node. To avoid potential ambiguity about the extension of a node, we complement the FGraph with an existence graph.

The existence graph (EGraph) uses a wiring-type diagram (i.e., orthogonal graphs with horizontal and vertical lines) to represent transactions and/or itemsets. The EGraph is designed to indicate whether or not an item is present in a transaction

and/or an itemset. It also indicates which transactions and itemsets are present within a specific context like a user selection. To focus on which items, transactions and itemsets exist, the transaction count of each transaction and the frequency count of each itemset is not displayed. Instead, we complement the EGraph with a frequency graph which reveals the count information.

An observant reader may notice that both FGraph and EGraph complement each other. They together are taking the benefits of both worlds: The FGraph shows the count information, whereas the EGraph shows the existence information and reveals the prefix-extension relationship. Moreover, we also provided several interactive features for our visualization system consisting of both FGraph and EGraph.

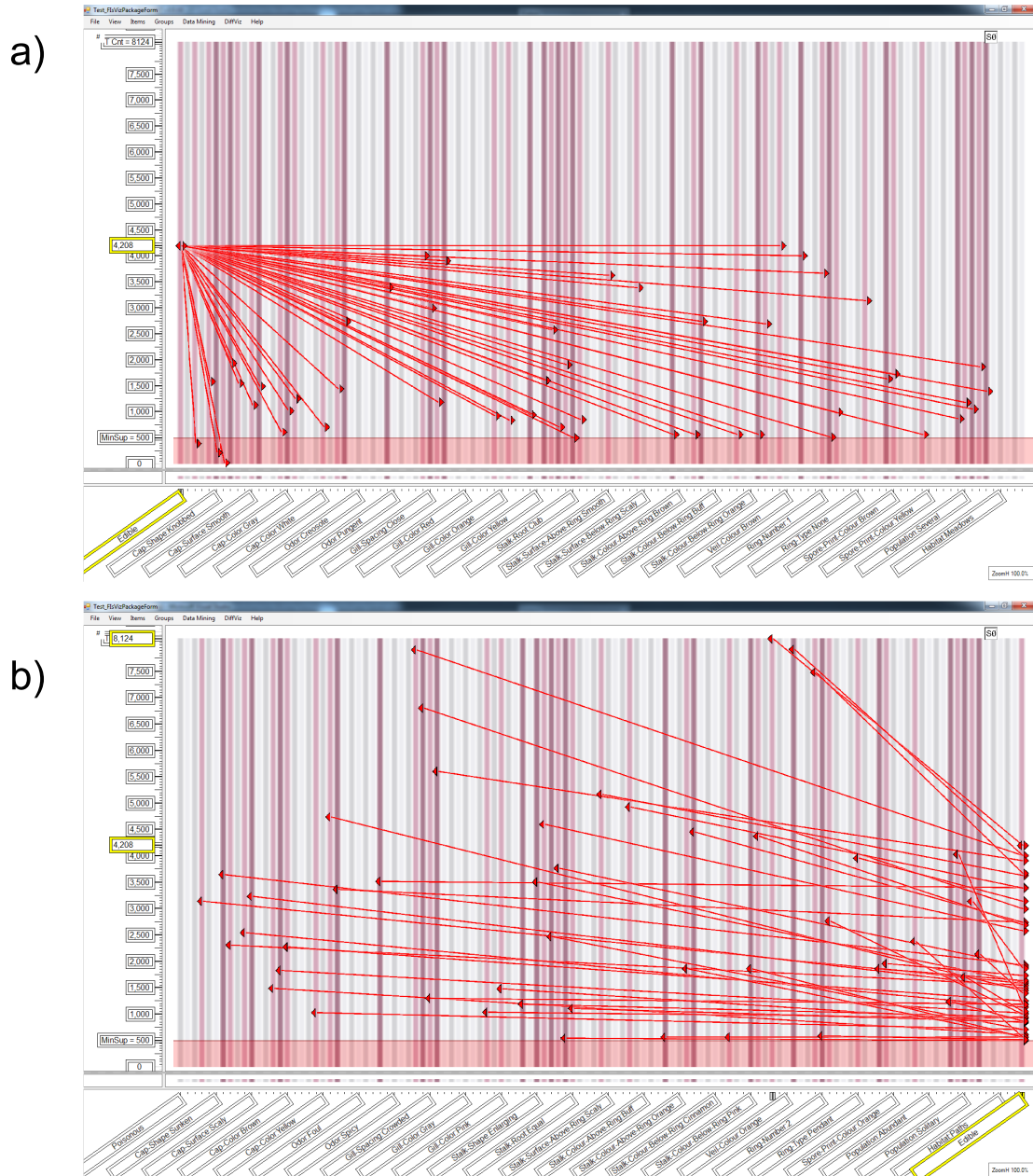


Figure 3.6: FGraph contrasting edible 2-sets with different item orders.

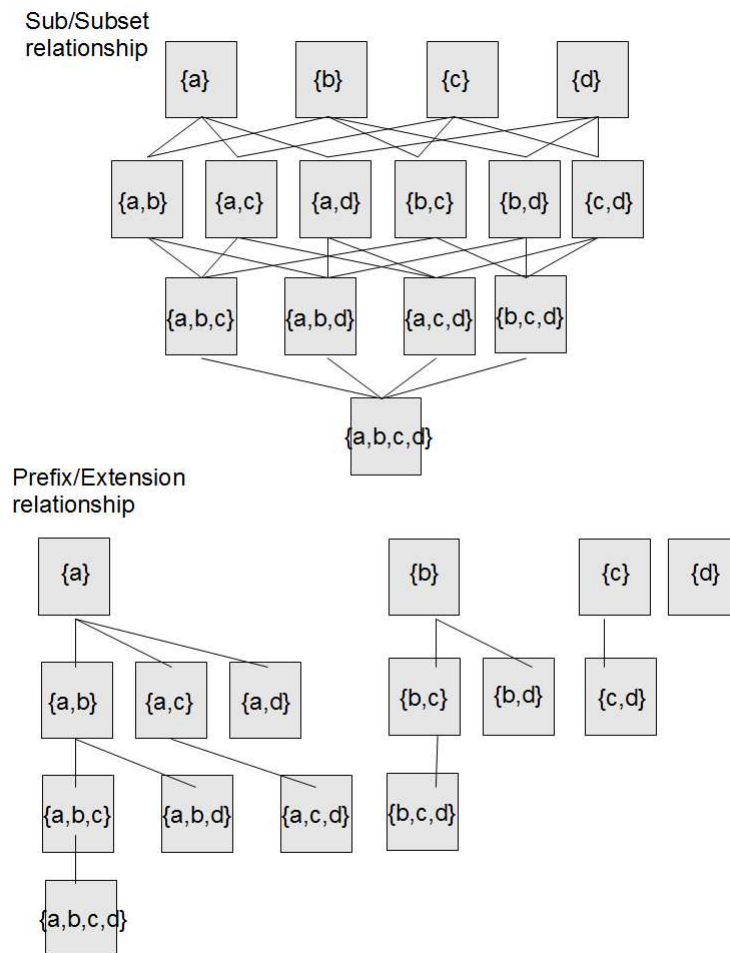


Figure 3.7: Comparing subset/superset and prefix/extension relationships.

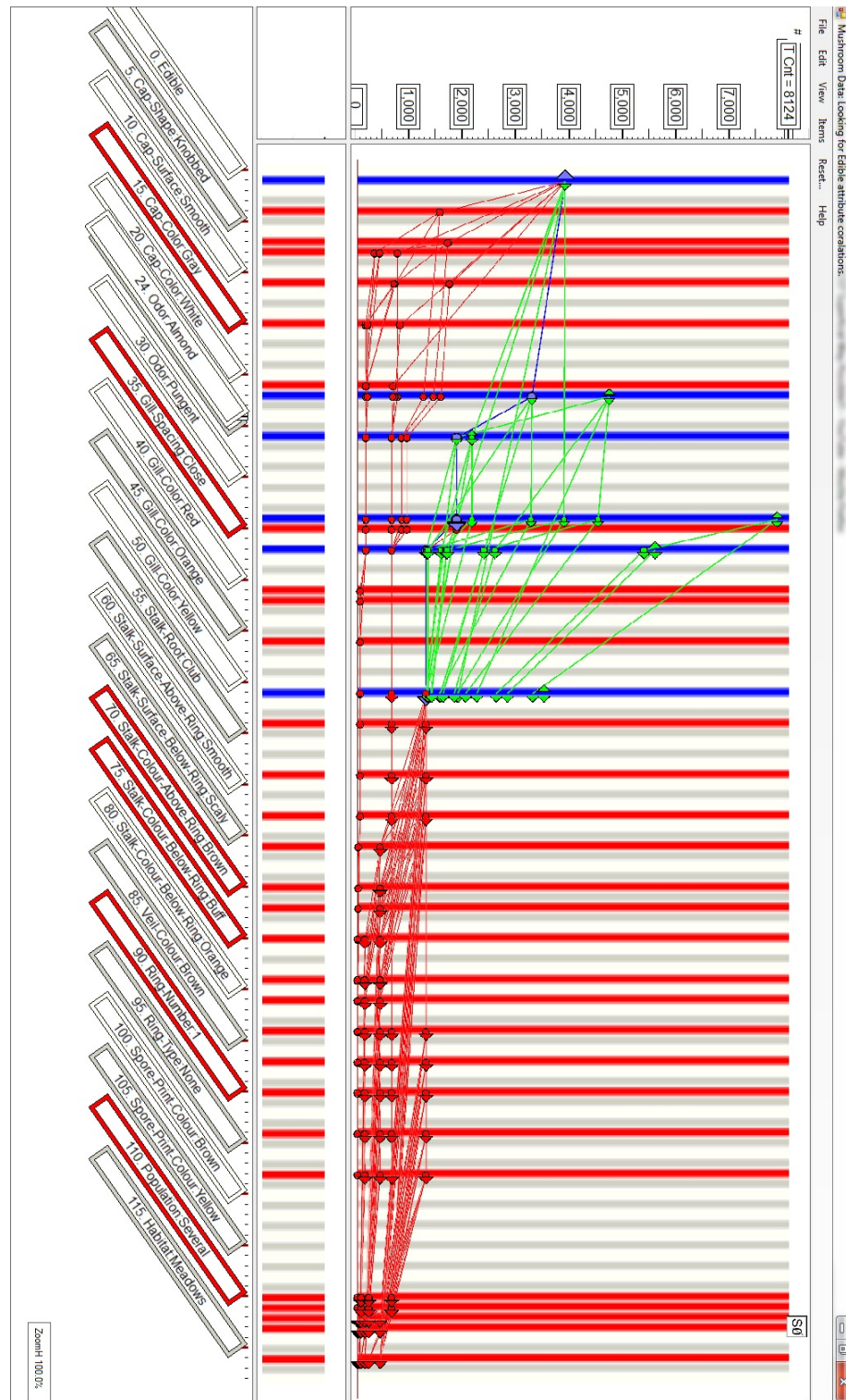


Figure 3.8: FGraph displaying supersets and subsets.

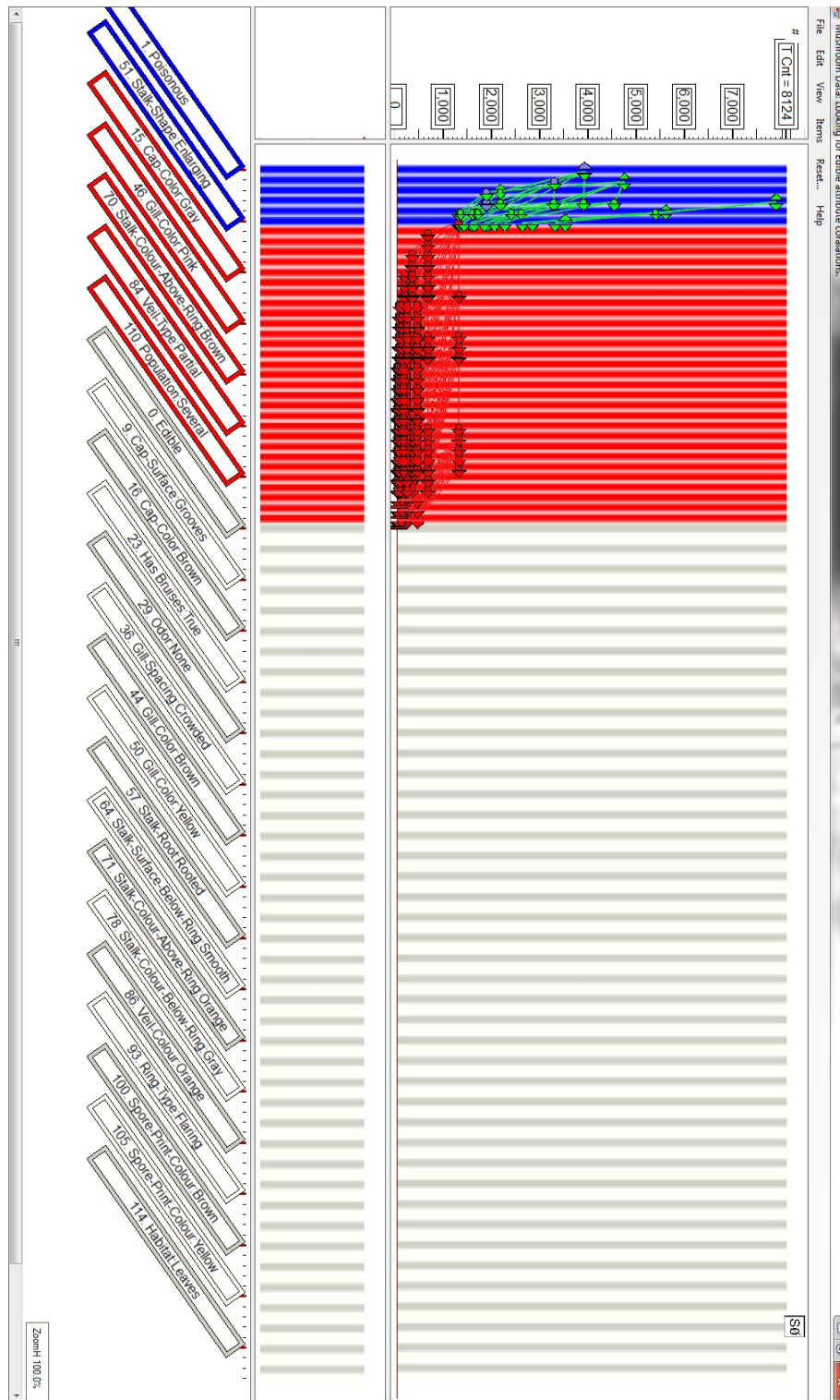


Figure 3.9: FGraph displaying supersets and subsets with ordered items.



Figure 3.10: The EGraph is used display individual highlighted itemsets

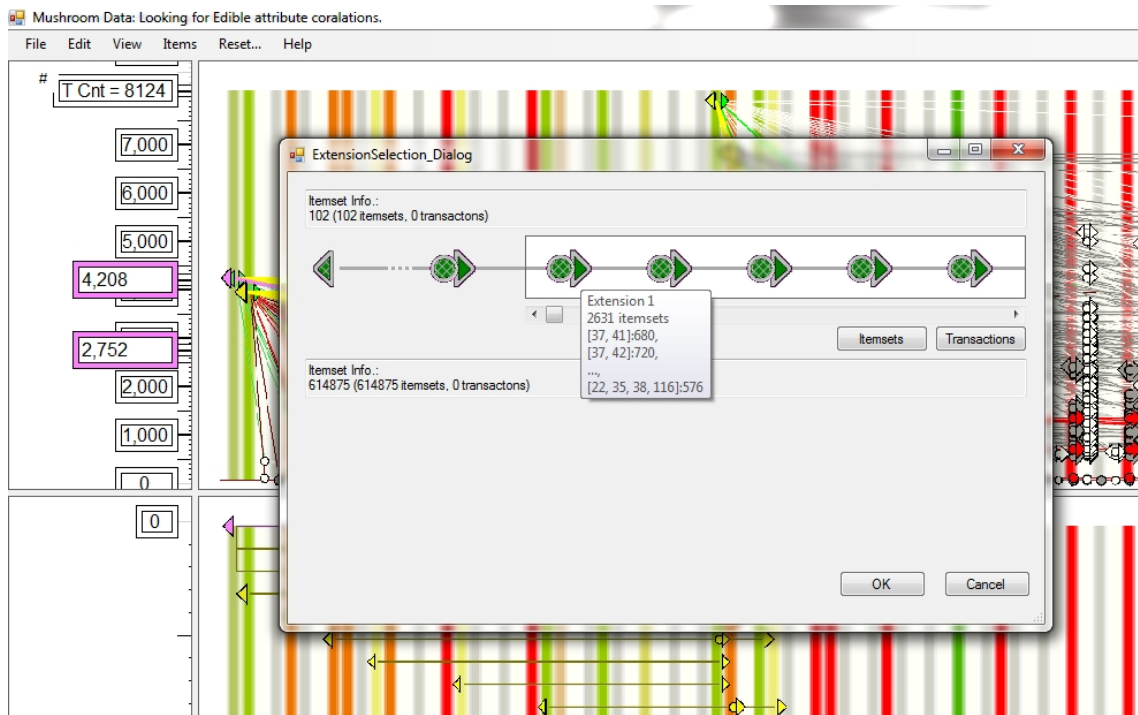


Figure 3.11: The extension interface adds extensions that have already been mined.

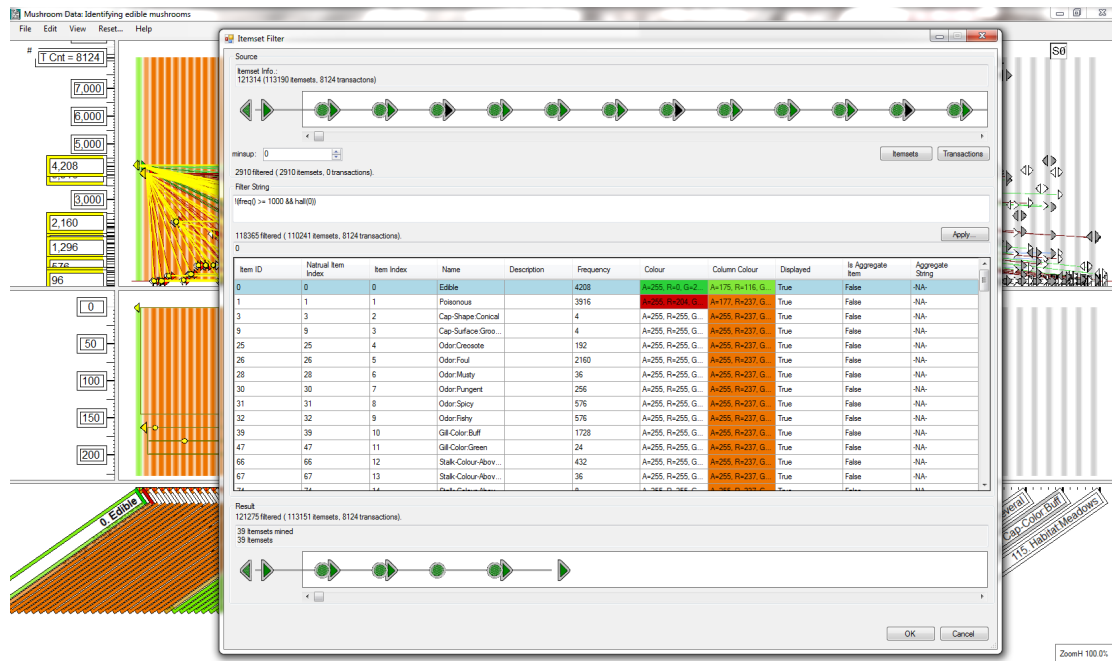


Figure 3.12: Selection filtering dialogue.

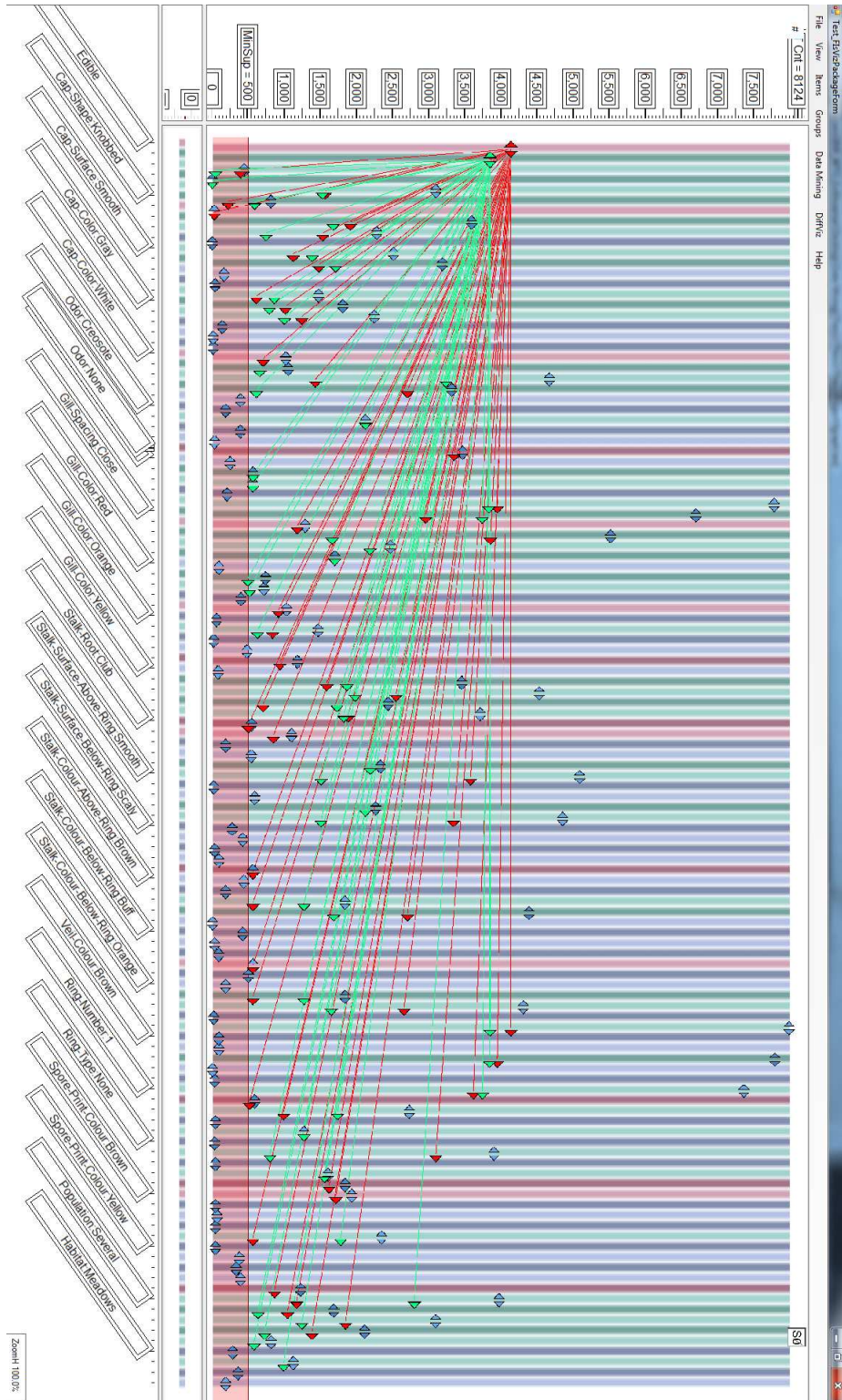


Figure 3.13: Three itemset groups coloured red green and blue.

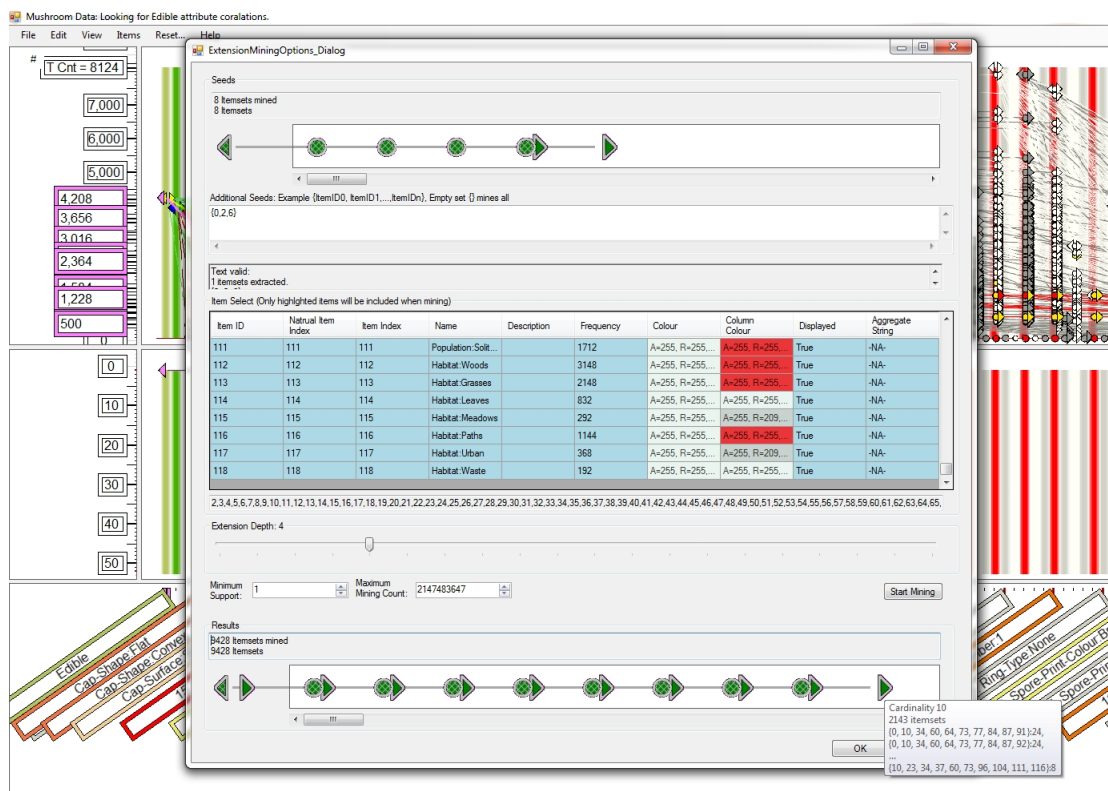


Figure 3.14: Constraint based data mining interface.

The screenshot displays the 'Item Editor' window within an 'Unnamed ItemsetCollection' application. The window features a menu bar (File, Edit, View, Reset, Tests, Help) and a sidebar on the left with a tree view showing a hierarchy of items. The main area contains a table with the following data:

Item ID	Natural Item Index	Name	Description	Frequency	Colour	Column Colour	Displayed	Aggregate String
0	0	Edible		4208	A=255, R=0, G=0	A=255, R=0, G=0	True	NA
1	1	Poisonous		3916	A=255, R=255, G=0	A=255, R=255, G=0	True	NA
2	2	Cap-Shape Bell		452	A=255, R=255, G=0	A=255, R=255, G=0	True	NA
3	3	Cap-Shape Conical		4	A=255, R=255, G=0	A=255, R=209, G=0	True	NA
4	4	Cap-Shape Flat		3152	A=255, R=255, G=0	A=255, R=255, G=0	True	NA
5	5	Cap-Shape Knobbed		328	A=255, R=255, G=0	A=255, R=209, G=0	True	NA
6	6	Cap-Shape Sunken		32	A=255, R=255, G=0	A=255, R=255, G=0	True	NA
7	7	Cap-Shape Convex		3656	A=255, R=255, G=0	A=255, R=209, G=0	True	NA
8	8	Cap-Surface Fibrous		2320	A=255, R=168, G=0	A=255, R=196, G=0	True	NA
9	9	Cap-Surface Groov...		4	A=255, R=168, G=0	A=255, R=138, G=0	True	NA
10	10	Cap-Surface Smooth		2556	A=255, R=168, G=0	A=255, R=196, G=0	True	NA
11	11	Cap-Surface Scaly		3244	A=255, R=168, G=0	A=255, R=138, G=0	True	NA
12	12	Cap-Color Buff		168	A=255, R=255, G=0	A=255, R=255, G=0	True	NA
13	13	Cap-Color Cinnamon		44	A=255, R=255, G=0	A=255, R=209, G=0	True	NA
14	14	Cap-Color Red		1500	A=255, R=255, G=0	A=255, R=255, G=0	True	NA
15	15	Cap-Color Gray		1840	A=255, R=255, G=0	A=255, R=209, G=0	True	NA
16	16	Cap-Color Brown		2294	A=255, R=255, G=0	A=255, R=255, G=0	True	NA
17	17	Cap-Color Pink		144	A=255, R=255, G=0	A=255, R=209, G=0	True	NA
18	18	Cap-Color Green		16	A=255, R=255, G=0	A=255, R=255, G=0	True	NA
19	19	Cap-Color Purple		16	A=255, R=255, G=0	A=255, R=209, G=0	True	NA
20	20	Cap-Color White		1040	A=255, R=255, G=0	A=255, R=255, G=0	True	NA

The interface also includes a sidebar on the left with a tree view showing a hierarchy of items, and a main window with a table and buttons for OK, Cancel, and Apply. The status bar at the bottom right indicates 'ZoomH 100.0%'.

Figure 3.15: Item editor interface.

Chapter 4

Case Studies

In the previous chapter, we described our visualization system. In this chapter, let us evaluate the practicality of the system on three case studies. They demonstrate the diversity of data mining problems that can be solved with this visualization.

First, we visualize the mushroom dataset as it is one of the benchmark databases used in the data mining community. Then, we visualize the wine dataset. It highlights how we mine analog data by discretizing the continuous values in the transaction data set. The system is used as a visual classifier so that users can answer questions like “What attributes determine a class 1 wine?”. Finally, we study a co-authorship database extracted from the Digital Bibliography & Library Project (DBLP) at <http://dblp.uni-trier.de/db/>. It illustrates the ability of the system to compare and contrast the difference between groups of authors who have collaborated together.

4.1 Mushroom Data

So far, we have used the mushroom dataset for a few examples in this thesis. This dataset is from UC Irvine [FA13], and was originally drawn from The Audubon Society Field Guide to North American Mushrooms. The dataset captures 119 attributes for 8124 instances of mushrooms. The task of this case study is to identify the attributes that indicate whether a mushroom is edible or poisonous. Similar to a real world problem, we find a list of item attributes that identify 100% of all edible and poisonous mushrooms.

After loading all 8124 mushroom transactions, we get Figure 4.1. The user is presented with a screen showing only raw transactions. The transaction count (T-Cnt in the top left corner) is 8124. Items are distributed along the x-axis at the bottom and are grouped by similar attributes, i.e., all cap colours are grouped together, all odours are grouped together. The attributes edible and poisonous are to the far left as indicated by the item labels along the bottom of the screen. The angle of the label text is set to 35 degree (adjustable) to make the text more readable and to reduce the vertical space it takes up.

Next, the singleton 1-itemsets are mined. Figure 4.2 shows the attributes alternately coloured blue and yellow. The edible and poisonous item attributes are coloured green and red, respectively. When the user is searching for a mushroom to eat, he would pick those highlighted in green (i.e., edible mushroom) to avoid consuming those poisonous ones (highlighted in red). The heights of the green and red nodes show that just over half of all mushrooms are edible. All the mined singleton itemsets are coloured orange, and all other itemsets and transactions have been re-

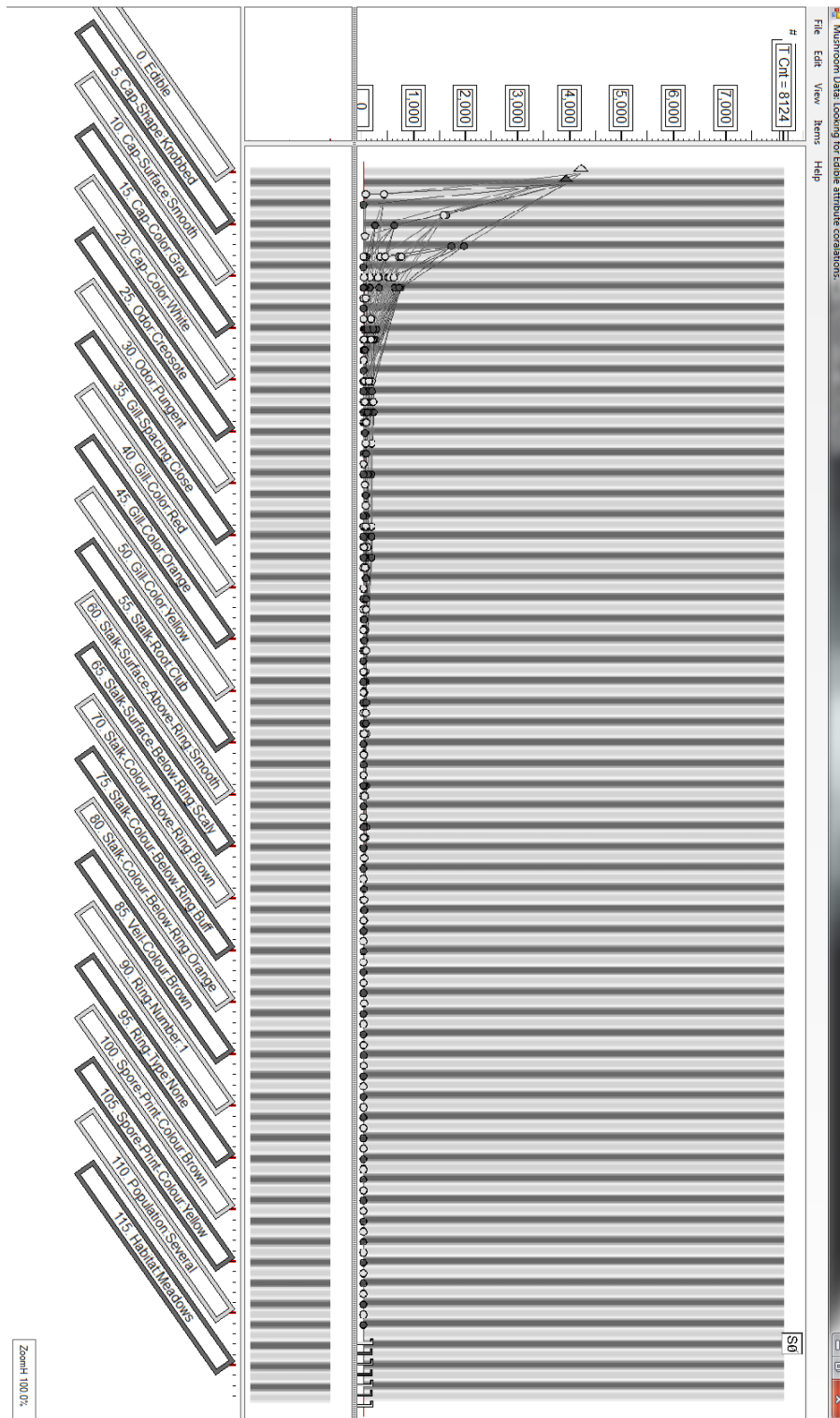


Figure 4.1: Raw mushroom transactions.

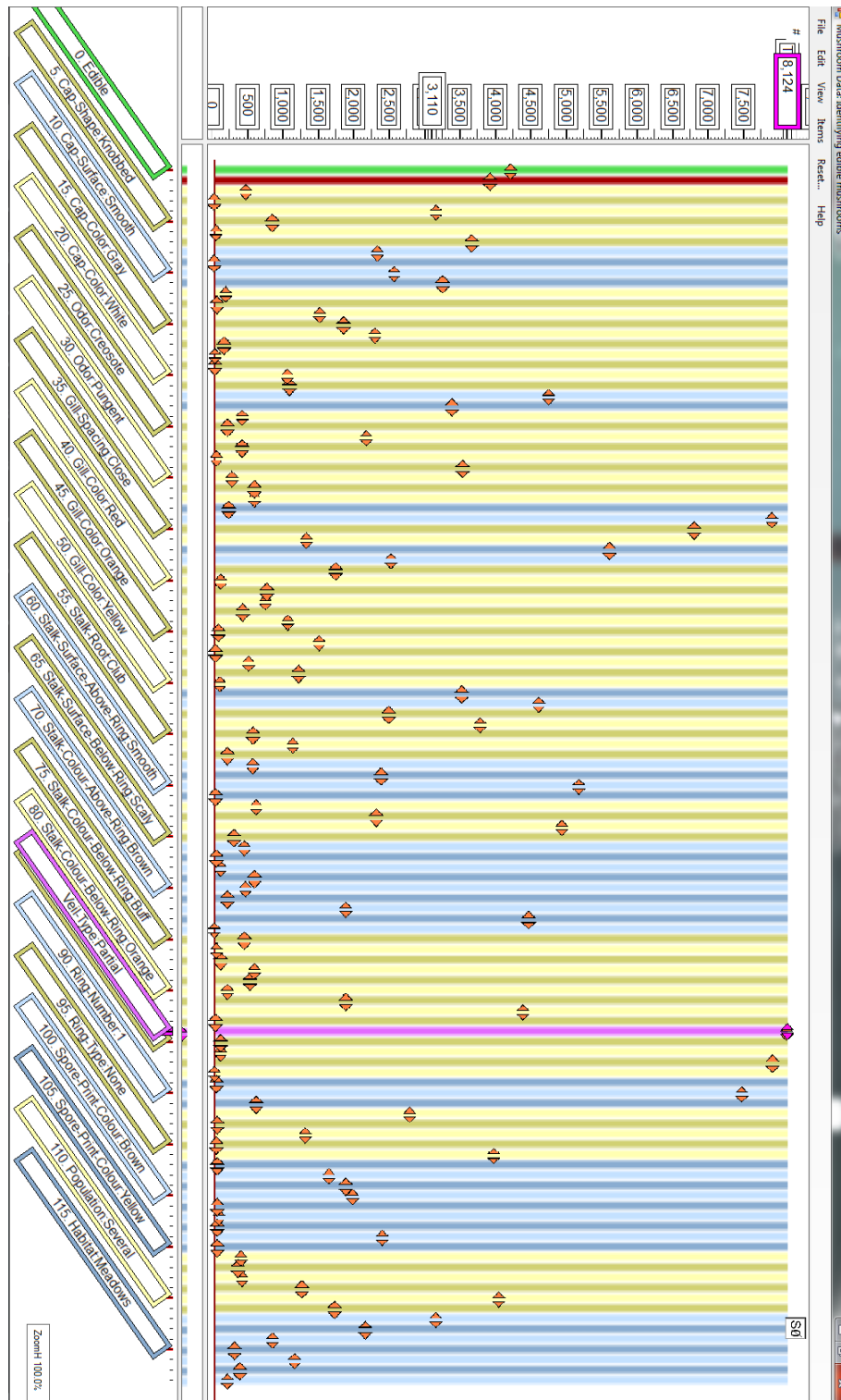


Figure 4.2: Frequent singletons mined from mushroom.

moved. This figure gives an overview of the database. It shows how often each item has occurred. Each yellow and blue attribute set of items can be examined, to get a sense of how often the individual attributes have occurred. Because of the nature of this data domain, the sum of all frequencies in a blue or yellow attribute set add up to 8124, which is the total transaction count. In other words, all mushrooms have been assigned one and only one odour item so the total frequency of all odour items is 8124.

In Figure 4.2, one itemset stands out because of its frequency. The singleton itemset {Veil Type:Partial} has a frequency of 8124, which is the same as the transaction count. This means every mushroom in the database has partial veil.

Next, we mine for one-item extensions of both edible and poisonous mushrooms and get Figure 4.3. Here, the edible extensions are highlighted in green and poisonous extensions are highlighted in red. Those item attributes that occur on both edible and poisonous mushrooms are coloured brown, which is a combination of red and green. By looking at item column colours, it is clear (a) which items only occur with the edible attribute, (b) which items only occur with the poisonous attribute, and (c) which ones occur on both.

Next, we isolate these three groups of item attributes. Since the column colour now represents which group they belong to, all the items that occur only on poisonous mushrooms can be moved to the left, followed by those items that only occur with edible mushrooms, followed by the rest. Figure 4.4 shows how the column colour in the Item Edit screen is used to reorder the items and Figure 4.5 shows the results.

Let us call the group of items that only occur on poisonous mushrooms P1. These

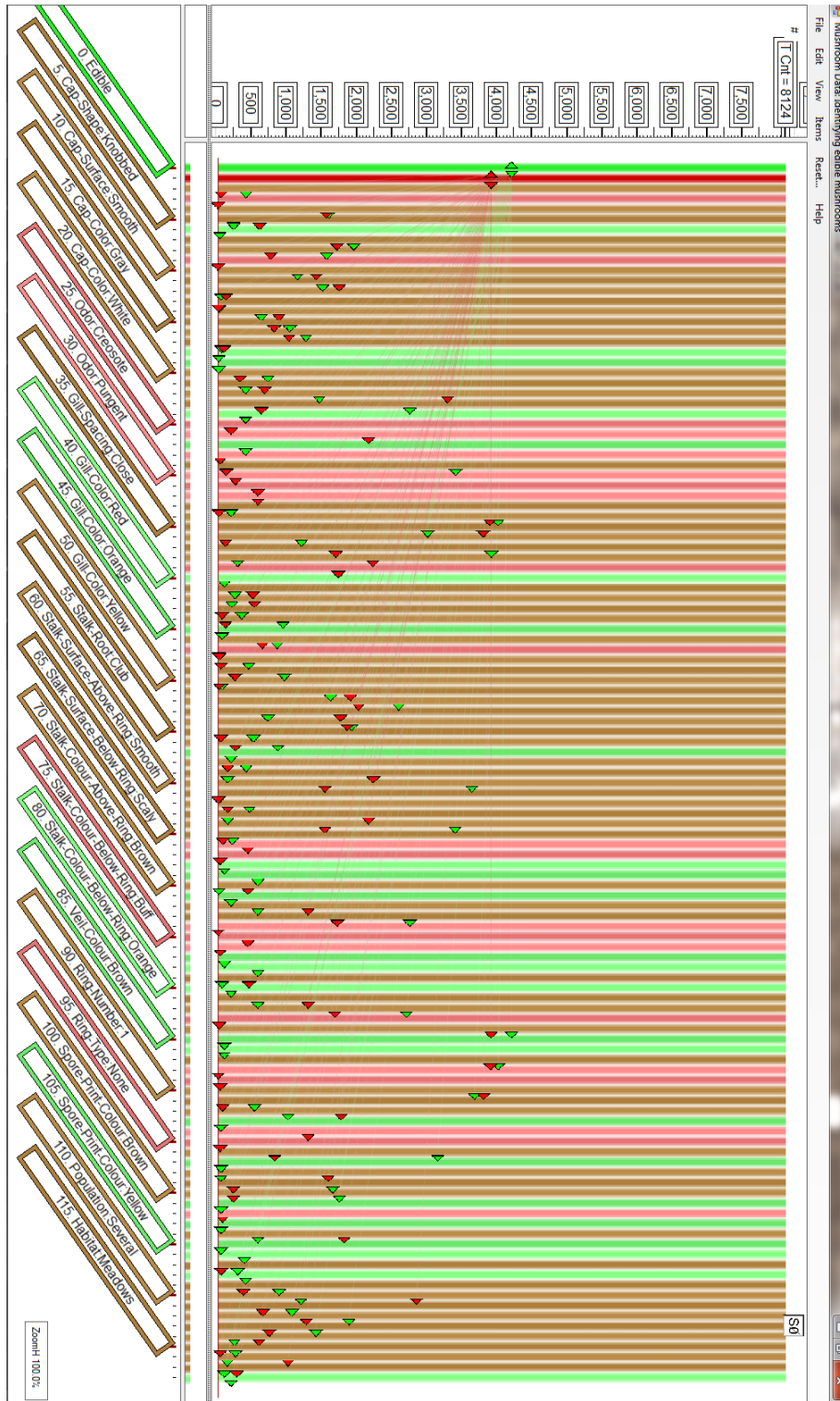


Figure 4.3: One-item extensions of both edible and poisonous mushrooms.

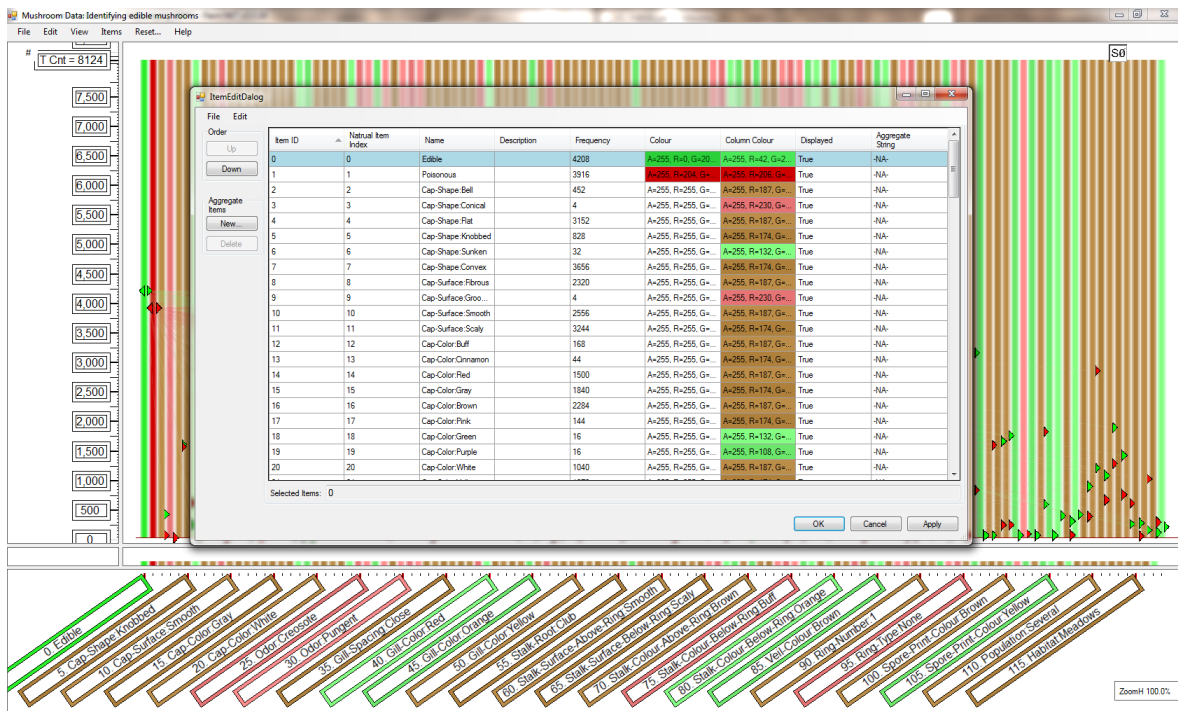


Figure 4.4: Item editor.

are items 3, 9, 25, 26, 28, 30, 31, 32, 39, 47, 66, 67, 74, 75, 76, 83, 88, 89, 94, 95, and 102. Their names can be read from Figure 4.6. With the programming symbol “!” to denote negation, the group !P1 is made up of those itemsets items not in P1. To specify these groups P1 and !P1, the user created functions $\text{hany}(3, 9, 25, 26, 28, 30, 31, 32, 39, 47, 66, 67, 74, 75, 76, 83, 88, 89, 94, 95, 102)$, and $!\text{hany}(3, 9, 25, 26, 28, 30, 31, 32, 39, 47, 66, 67, 74, 75, 76, 83, 88, 89, 94, 95, 102)$ are used, respectively. See Figure 4.6.

We use the same procedure to display those item attributes that only occur on edible mushrooms. The item groups E1 and !E1 are created. Specifically these item IDs are 6, 18, 19, 24, 27, 40, 45, 57, 68, 69, 71, 77, 78, 80, 85, 86, 93, 97, 101, 103,

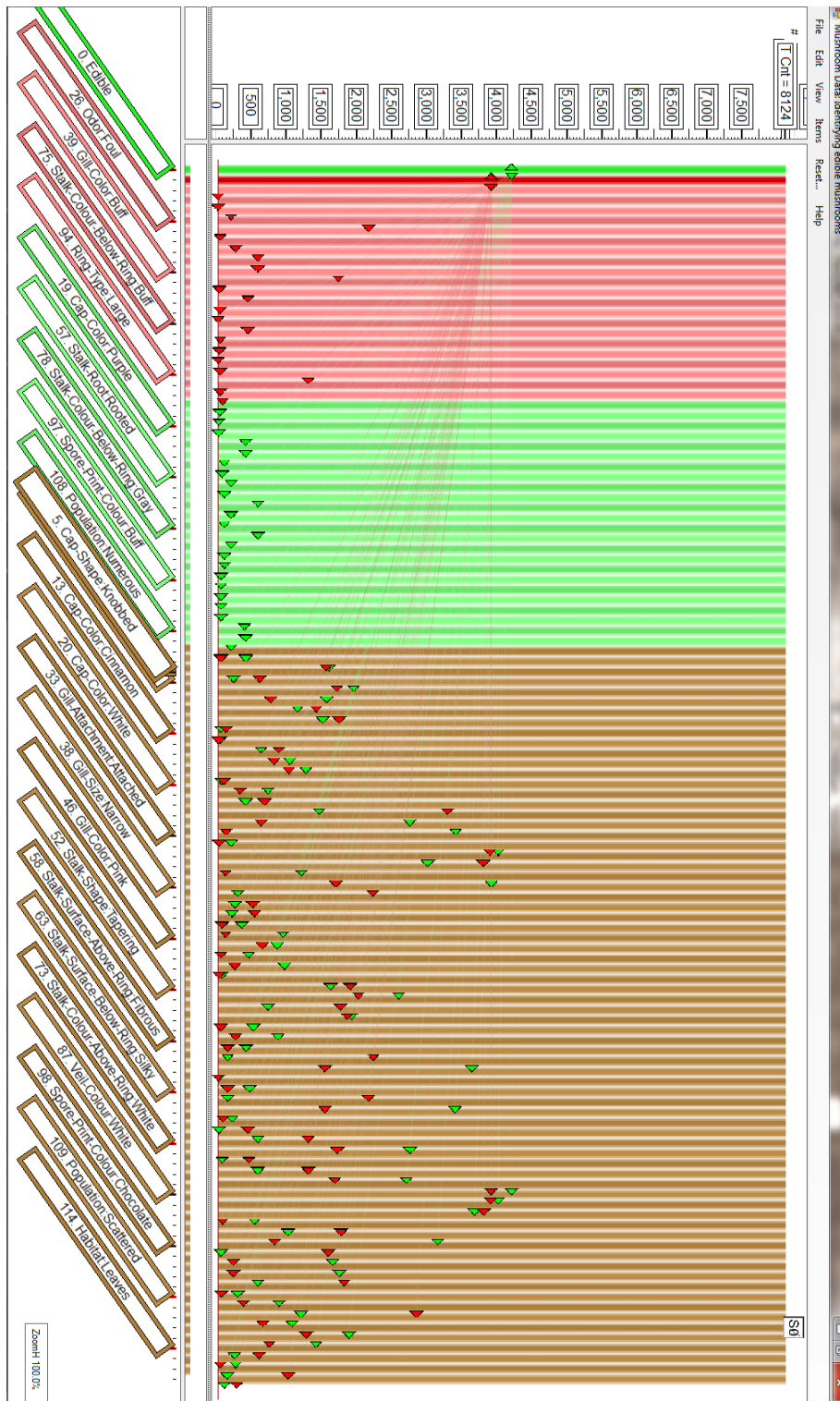


Figure 4.5: Groups of edible and poisonous mushroom.

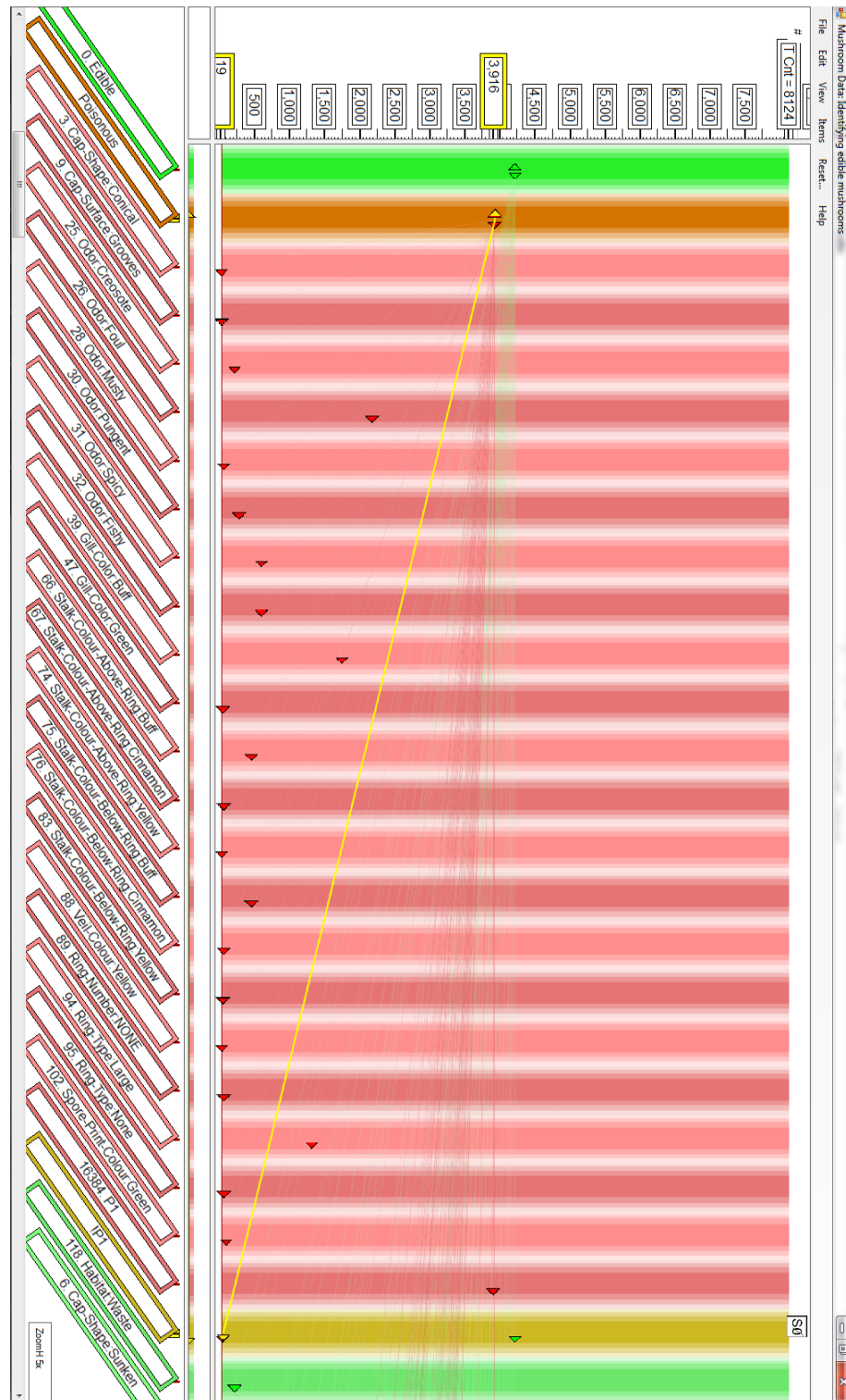


Figure 4.6: Poisonous mushrooms plus aggregate items P1 and !P1.

105, 106, 108, and 118. There are 2752 mushrooms contained in E1 and obviously all are edible. There are 5372 mushrooms that occur in !E1. Of these, 1456 are edible, 3916 are poisonous. However, all but 19 of these can be eliminated if we only consider mushrooms that have attributes from the !P1 item group. Having already identified a good portion of both poisonous and edible mushrooms, we continue the case study by looking at only those mushrooms that have at least one item from !P1 and at least one item from !E1. In other words, we continue by looking at only those mushrooms that do not have any poisonous only or edible only attribute items. All those mushrooms that contain at least one edible only or one poisonous only attribute item are removed from consideration.

In Figure 4.7, the extensions of itemsets $\{\text{edible}, !P1, !E1\}$ and $\{\text{poisonous}, !P1, !E1\}$ are coloured in the same manner as above. The colour of the column again allows for the arrangement of the extension items. Red columns are first. These items extend $\{\text{poisonous}, !P1, !E1\}$ and do not extend $\{\text{edible}, !P1, !E1\}$. Green columns are next. These items extend $\{\text{edible}, !P1, !E1\}$ and do not extend $\{\text{poisonous}, !P1, !E1\}$. Brown columns show the items that extend both itemsets $\{\text{poisonous}, !P1, !E1\}$ and $\{\text{edible}, !P1, !E1\}$. Finally, those columns that are left are coloured white and are items that do not extend either itemset. Items like these exist because they do not occur on any mushroom grouped in !E1 and !P1.

The extensions of $\{\text{Edible}, !P1, !E1, !P2, !E2\}$ and $\{\text{Poisonous}, !P1, !E1, !P2, !E2\}$ can now be coloured and examined. All poisonous and edible mushrooms are accounted for with the final definition of item groups P3 and E3 as shown in Figure 4.8. This figure zooms-in to the ordered extension of $\{\text{Edible}, !E1, !P1, !E2,$

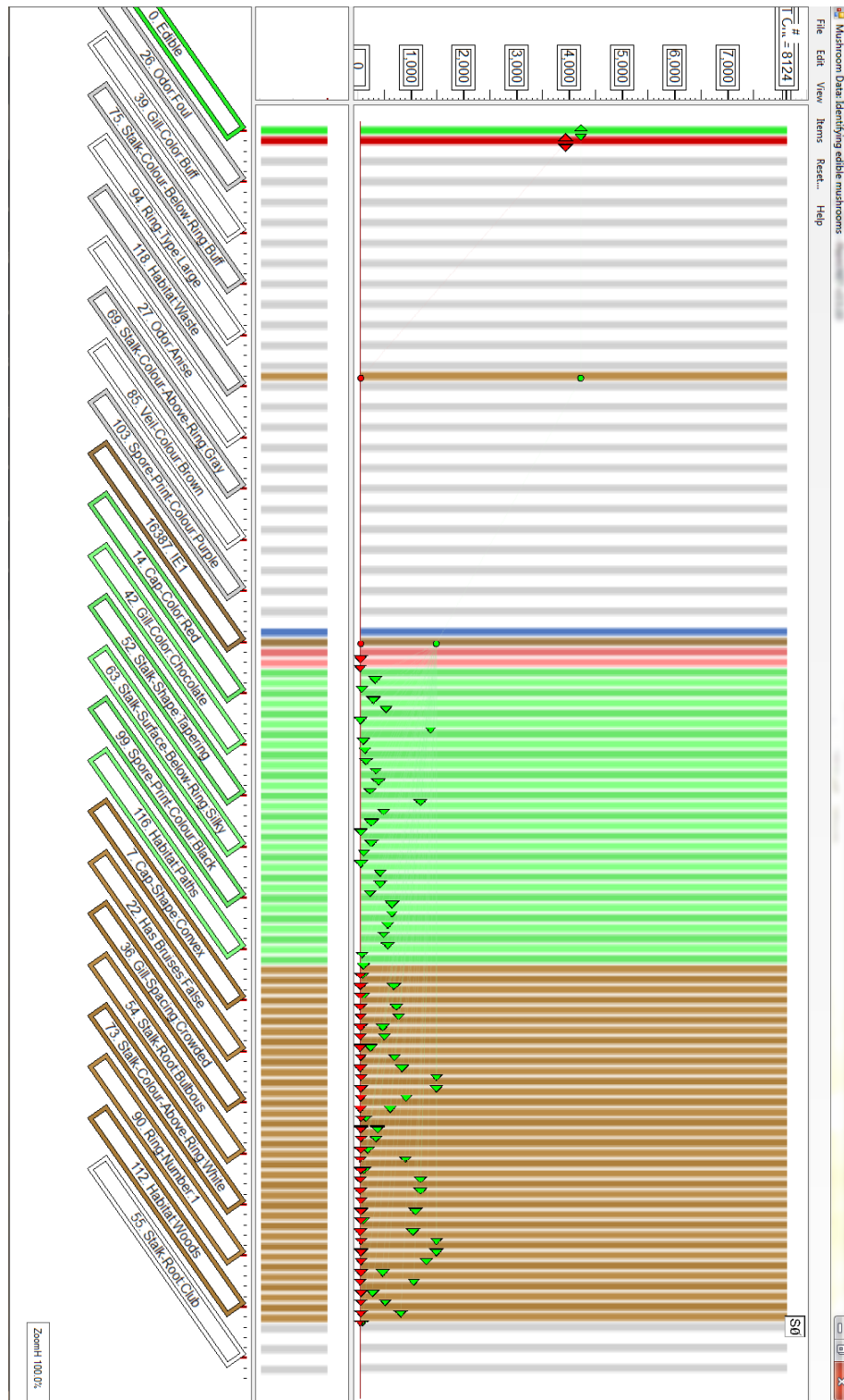


Figure 4.7: Mushroom data: poisonous and edible groups P2 and E2

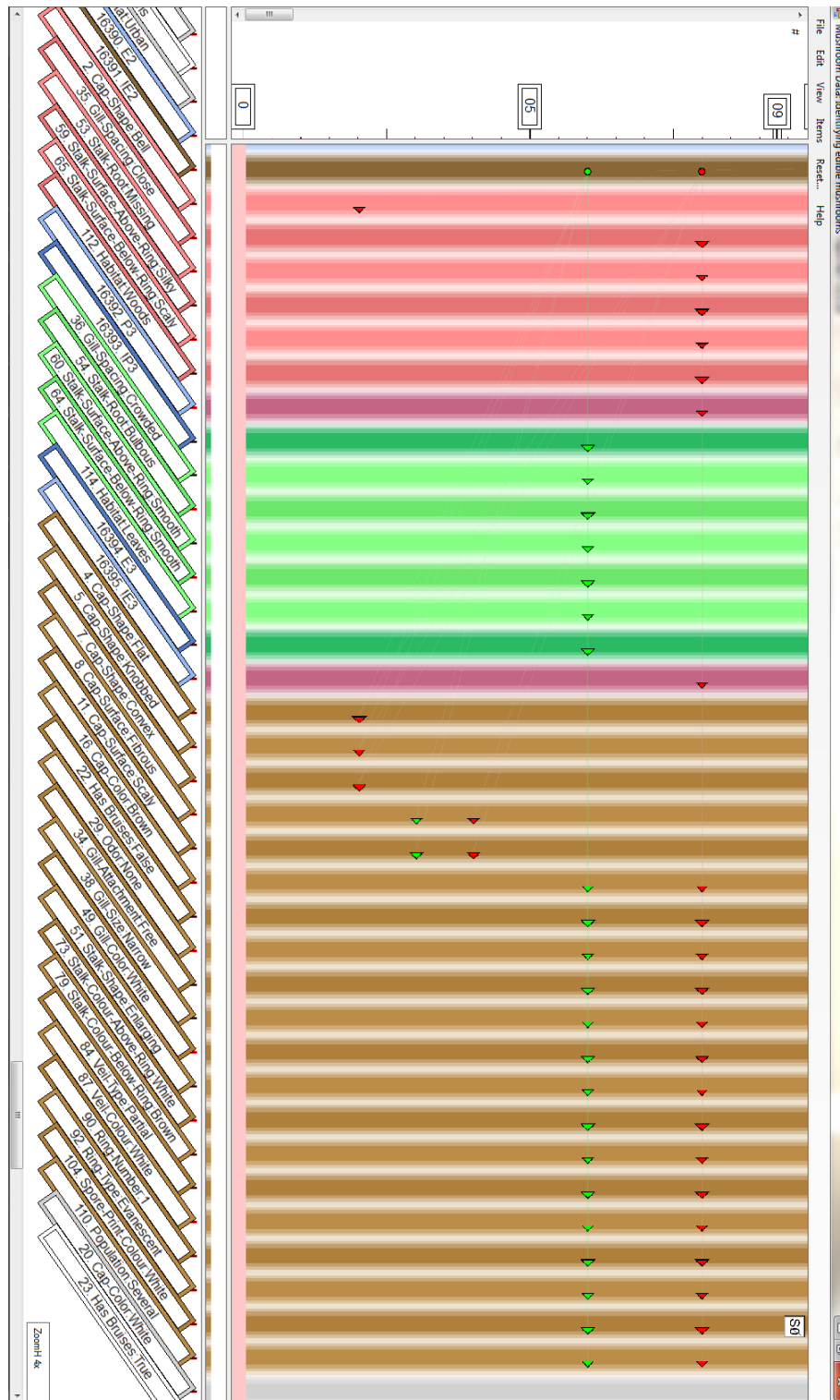


Figure 4.8: Mushroom data: poisonous group P3 and edible group E3.

$\{!P2\}$ and $\{\text{Poisonous}, !E1, !P1, !E2, !P2\}$. The itemsets $\{\text{Edible}, !P1, !E1, !P2, !E2\}$, $\{\text{Poisonous}, !P1, !E1, !P2, !E2\}$ enter the image frame from the left. These itemsets are represented by the first left green and red nodes respectively. Their frequencies are 6 and 8. Since there are several poisonous only attributes that account for all 8 of $\{\text{Poisonous}, !E1, !P1, !E2, !P2\}$ mushrooms and all poisonous only items that account for all 6 edible mushrooms, there is no need to continue after the final groups E3 and P3 are created.

With groups E1, E2 E3, P1, P2 and P3 all edible and poisonous mushrooms have been accounted for. Next, we provide a summary of our progress:

1. All mushrooms that have an attribute from E1 are edible.
2. All mushrooms that have an attribute from P1 are poisonous.

For the remaining mushrooms that do not have an attribute from E1 or P1:

1. All mushrooms that have an attribute from E2 are edible.
2. All mushrooms that have an attribute from P2 are poisonous.

For the remaining mushrooms that do not have an attribute from E1, P1, E2, or P2:

1. All mushrooms that have an attribute from E3 are edible.
2. All mushrooms that have an attribute from P3 are poisonous.

Figure 4.9 shows a summary of these rules. For each group the frequency of items in that group is shown. The itemset $\{\text{Edible}, !P1, !E1\}$ is highlighted to emphasize how many poisonous mushrooms are removed after Step 1 is complete.

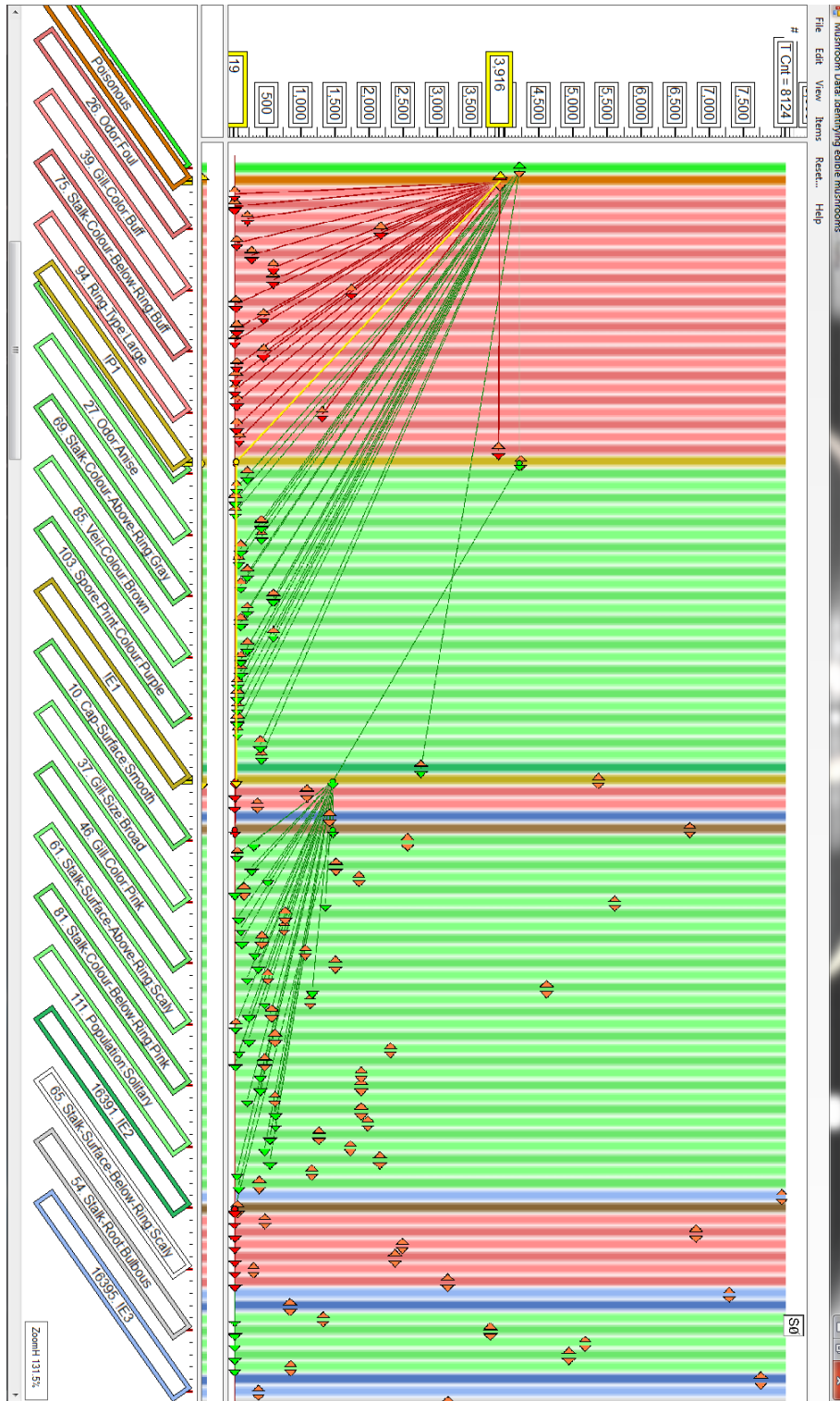


Figure 4.9: Mushroom data: summary

Figures 4.10, 4.11, 4.12 and 4.13 zoom in on each group to see the item labels and itemset frequencies.

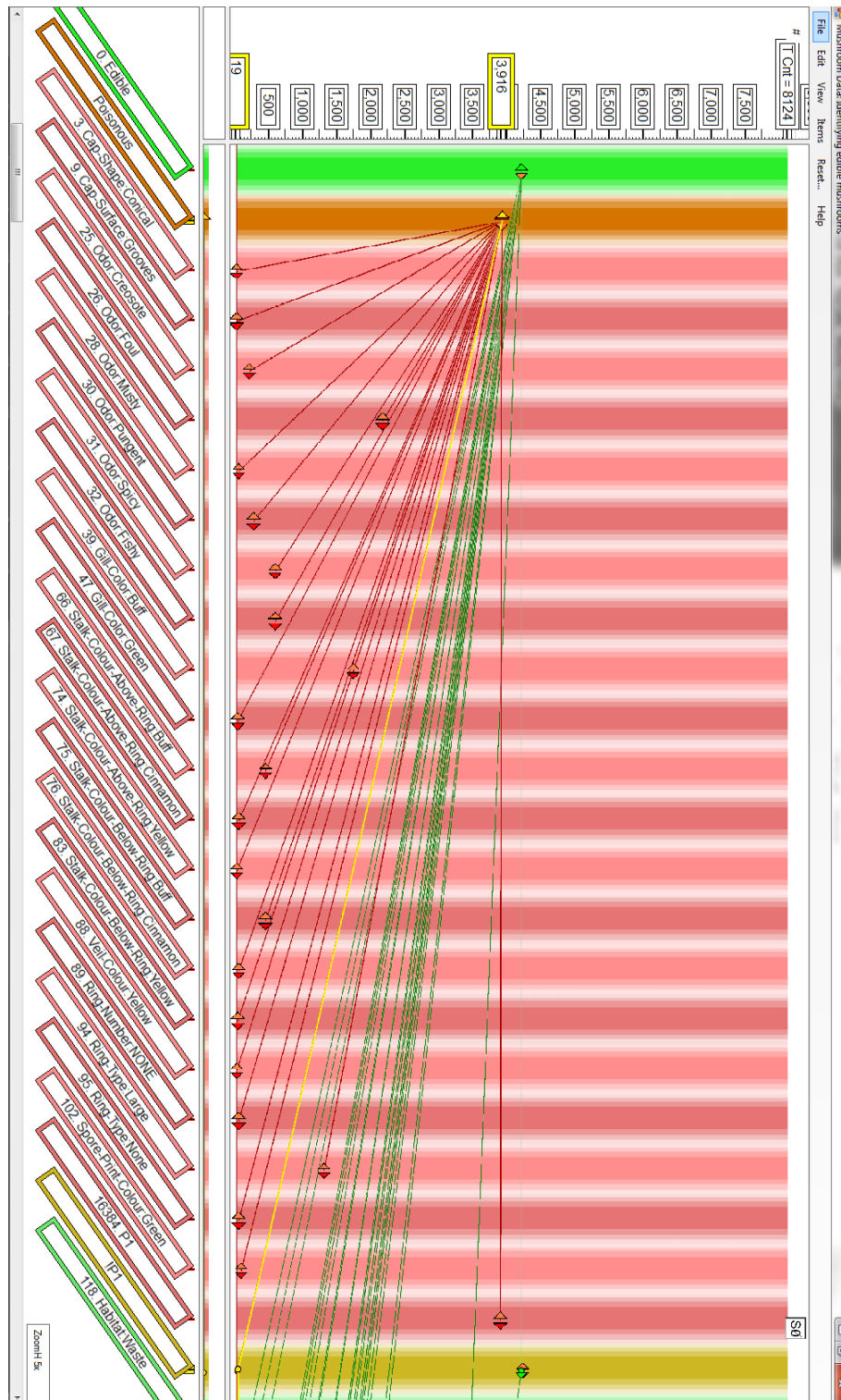


Figure 4.10: Mushroom data summary: poisonous group P1

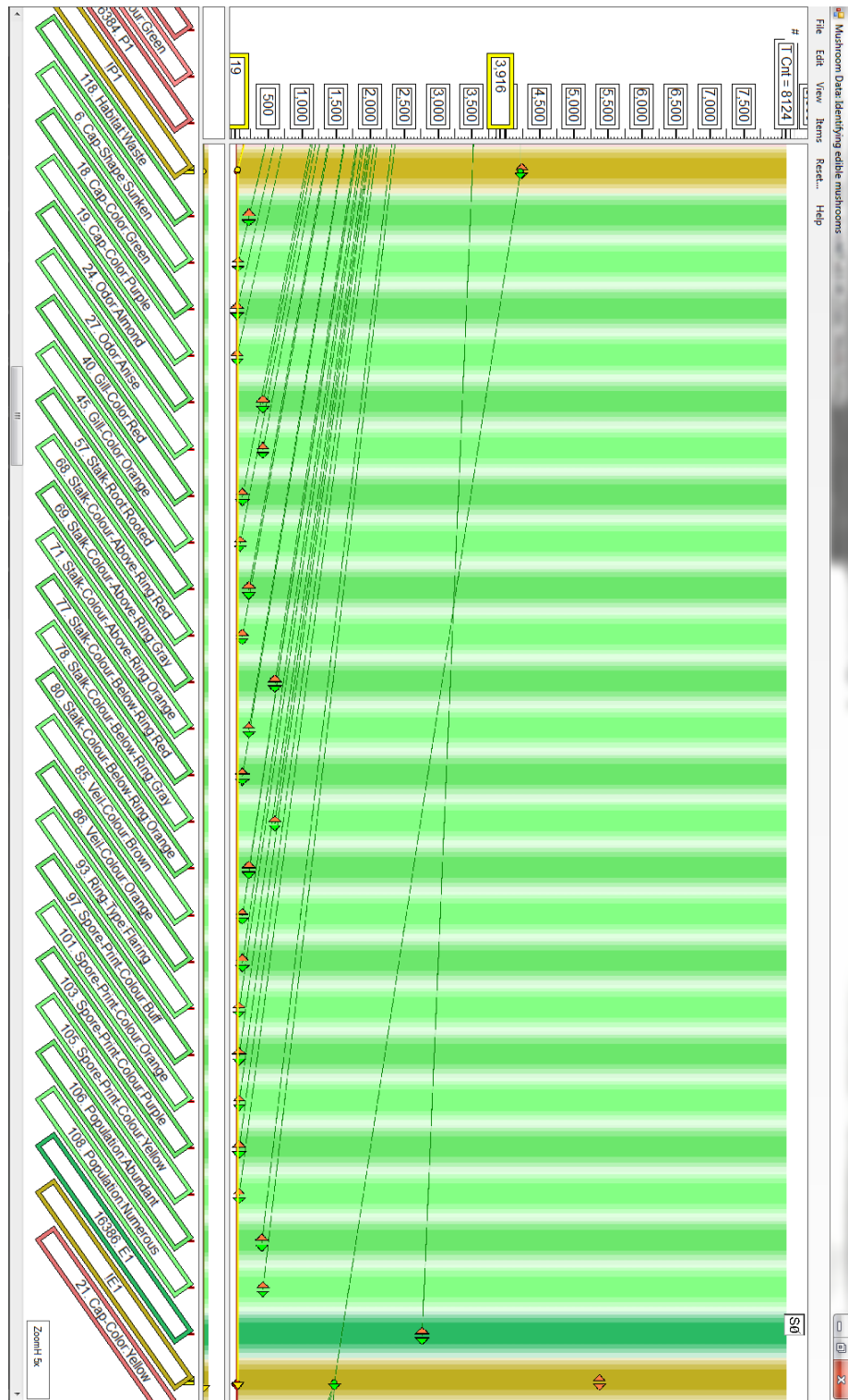


Figure 4.11: Mushroom data summary: edible group E1

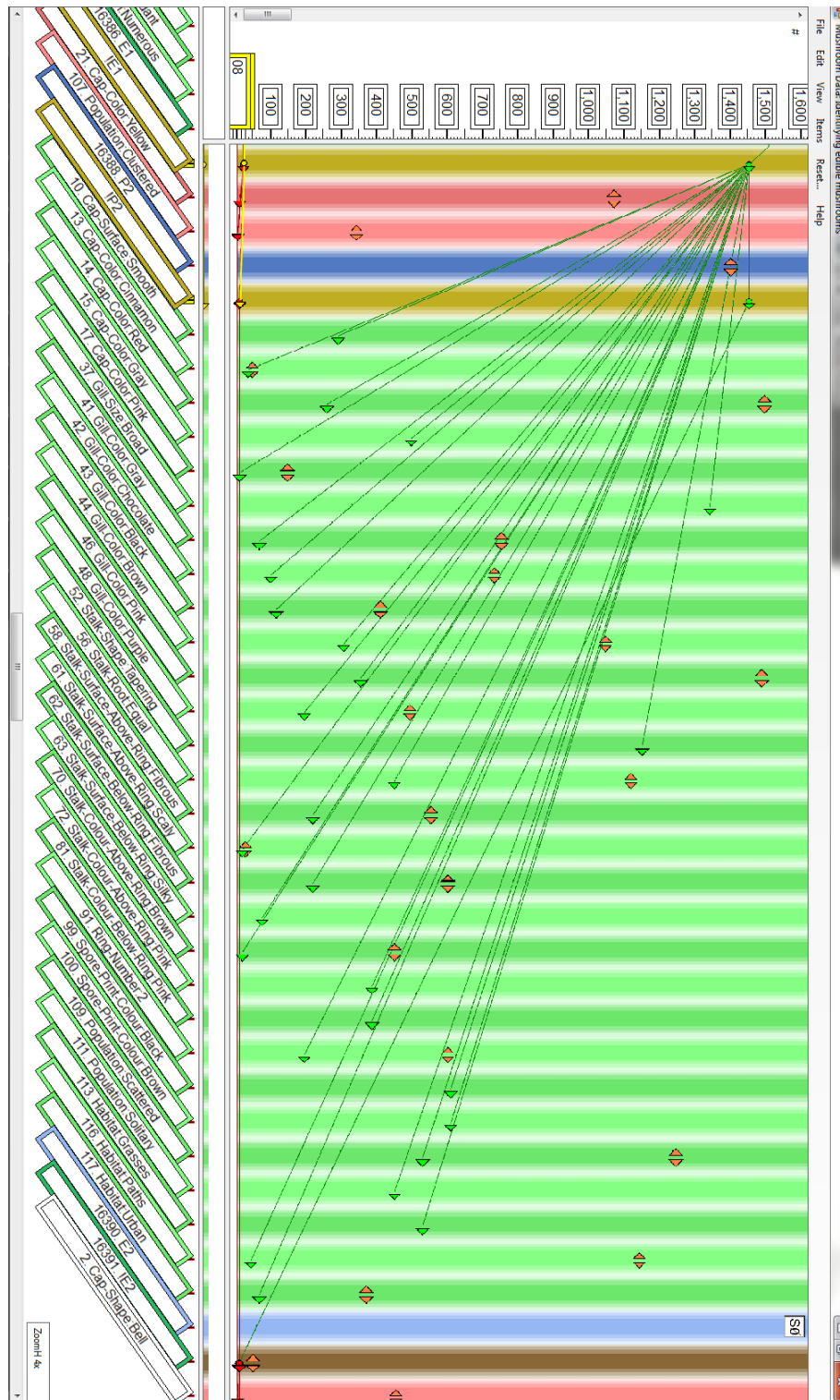


Figure 4.12: Mushroom data summary: poisonous group P2 and edible group E2

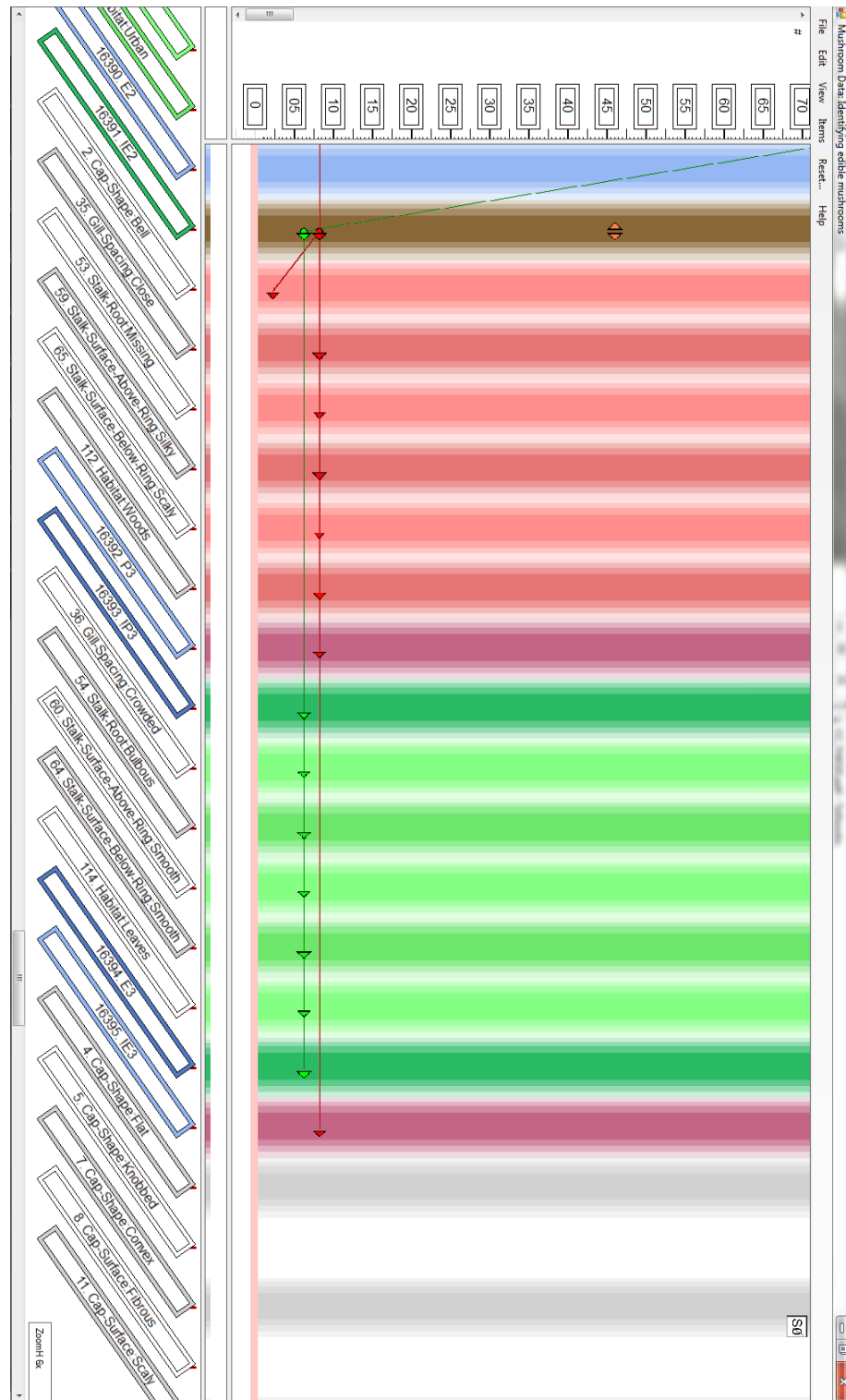


Figure 4.13: Mushroom data summary: poisonous group P3 and edible group E3

4.2 Wine Data

The wine data [FA13] is a result of a chemical analysis of 178 wines grown in the same Italian region, but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three classes of wines. The task is to classify class 1 wines. The result is a rule that distinguishes class 1 wines from the others. This is similar to the mushroom data task. However, as well as being an analog dataset, the wine data has three different types of class. The mushroom data had only two possible types, edible and poisonous.

Each transaction in the wine database contains analog data that must be discretized before we mine or display them. Each raw transaction has 13 analog values representing the amount of alcohol, malice acid, ash, etc. contained in the wine sampled. These values must be converted into discrete items before they can be displayed. For this case study, each of the 13 constituents has been broken up into approximately 10 intervals that depend on the range of the sample. For example, the alcohol content ranges from 11.03 to 14.83. To round off the values, this range has been extended to 11 to 15 and an interval range of 0.5 is used to produce 8 items, Alcohol:[11, 11.5), Alcohol:[11.5, 12), ..., Alcohol:[14.5, 15]. Each raw analog transaction is converted to discrete item transactions that can be mined and displayed. The resulting database has 142 items contained in 178 transactions.

Figure 4.14 gives an overview of the wine data. The three classes (1, 2, and 3) have been highlighted to show their frequencies. They have also been coloured green, blue, and red respectively. The 2-itemset extensions have been grouped and coloured as well. This figure shows (a) the frequency of each class and (b) how each of the

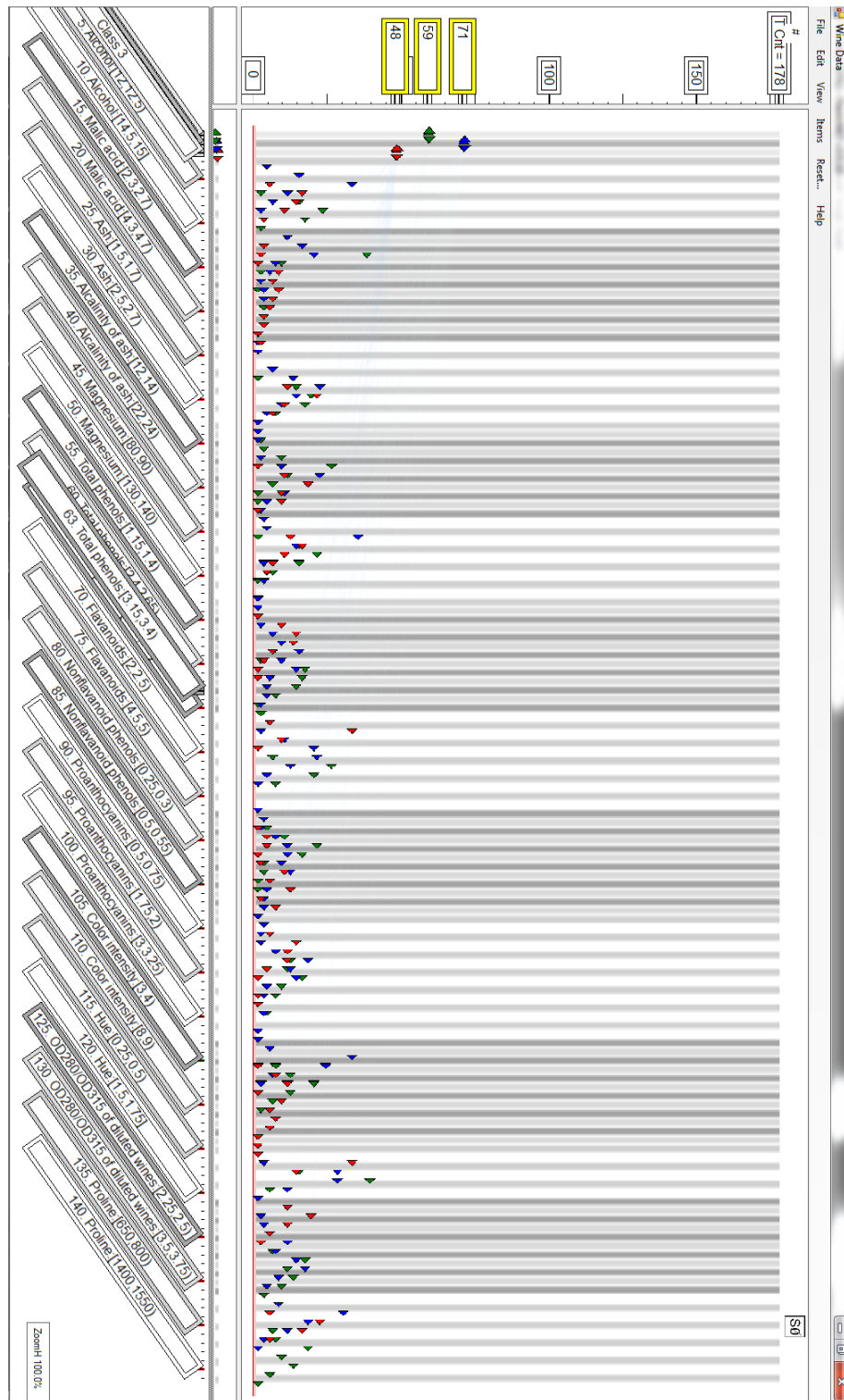


Figure 4.14: Wine data: class 1, 2, 3, extensions

three classes is distributed through the 13 constituents.

The first step is to split the database into 2 types of wines those that are class 1 and those that are not. This is easily done by introducing an aggregate item that is added to all transactions that do not contain the class 1 item. The next step is to mine all 2-itemset extensions of class 1 wines coloured green and all non-class 1 wines coloured red. Figure 4.15 is the result.

Aggregate items are then added to collect all class 1 wines in each constituent. Meaning, for each constituent, an aggregate item is created that can distinguish values of that constituent that most (or all) class 1 wines have. Figure 4.16 is zoomed-in on the malic acid constituent as an example. This figure shows that all class 1 wines (green nodes) have a malic acid value between 1.1 and 4.3. An aggregate item is then created and added to all wine transactions that have malic acid within this range. This image also shows the general distribution of this constituent with respect to both class 1 and non-class 1 wines. For example many class 1 wines fall into the [1.5,1.9) range and only non-class 1 wines fall into the [2.7,3.1) range. It should be noted that we are not going back to the original analog data to determine if it falls into the new aggregate item range. It is sufficient to look at the transaction data to consider whether or not it receives the new item. If a transaction contains any of the malic acid items 12, 13, 14, 15, 16, 17, 18, 19, or 20, it receives the new aggregate item.

Using this same method for the Ash constituent we are confronted with an outlier. Figure 4.17 shows the Ash constituent. The orange columns are the resulting aggregate items. The green columns are columns that contain at least one class 1

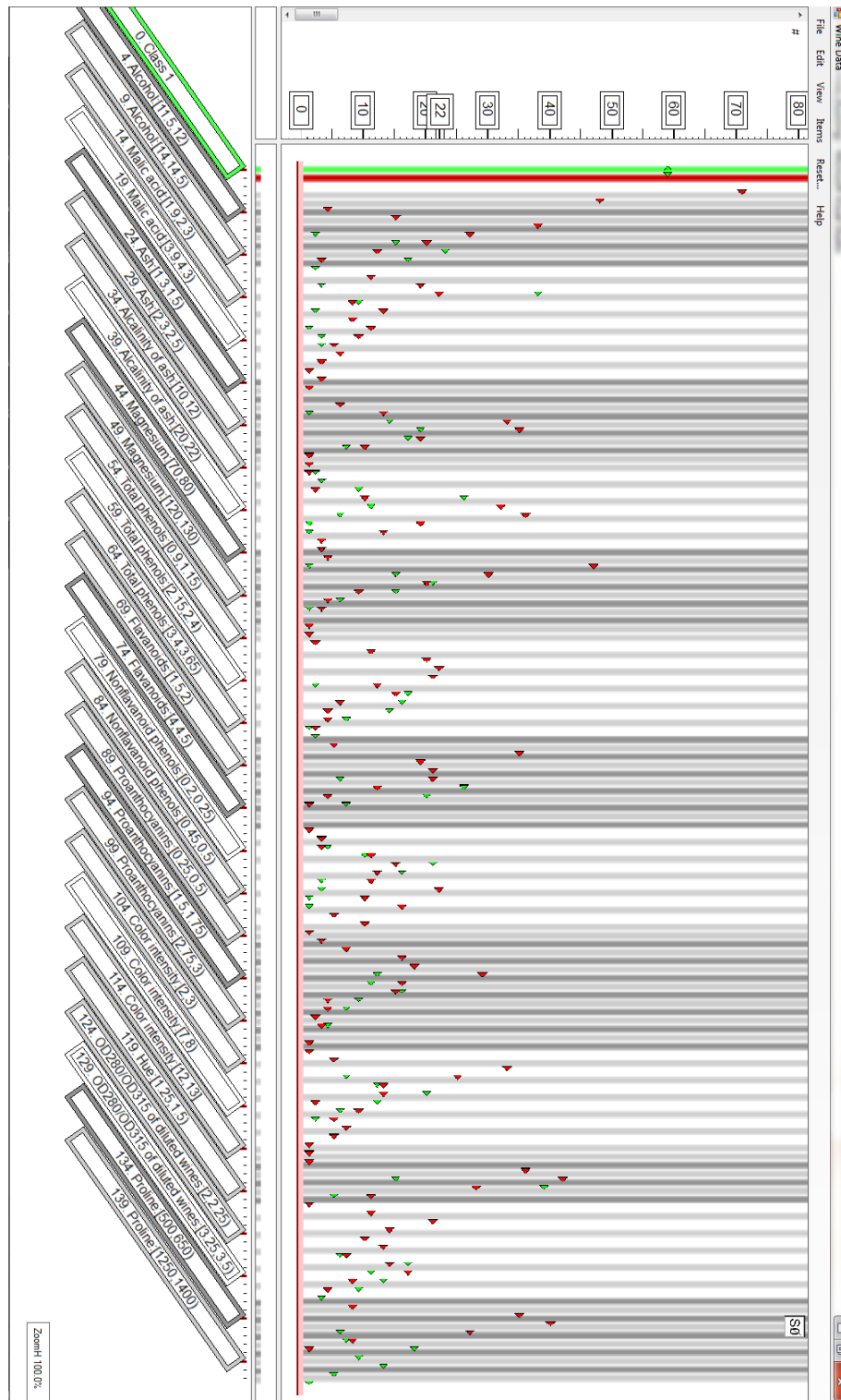


Figure 4.15: Wine data: class 1 and non-class 1 extensions.

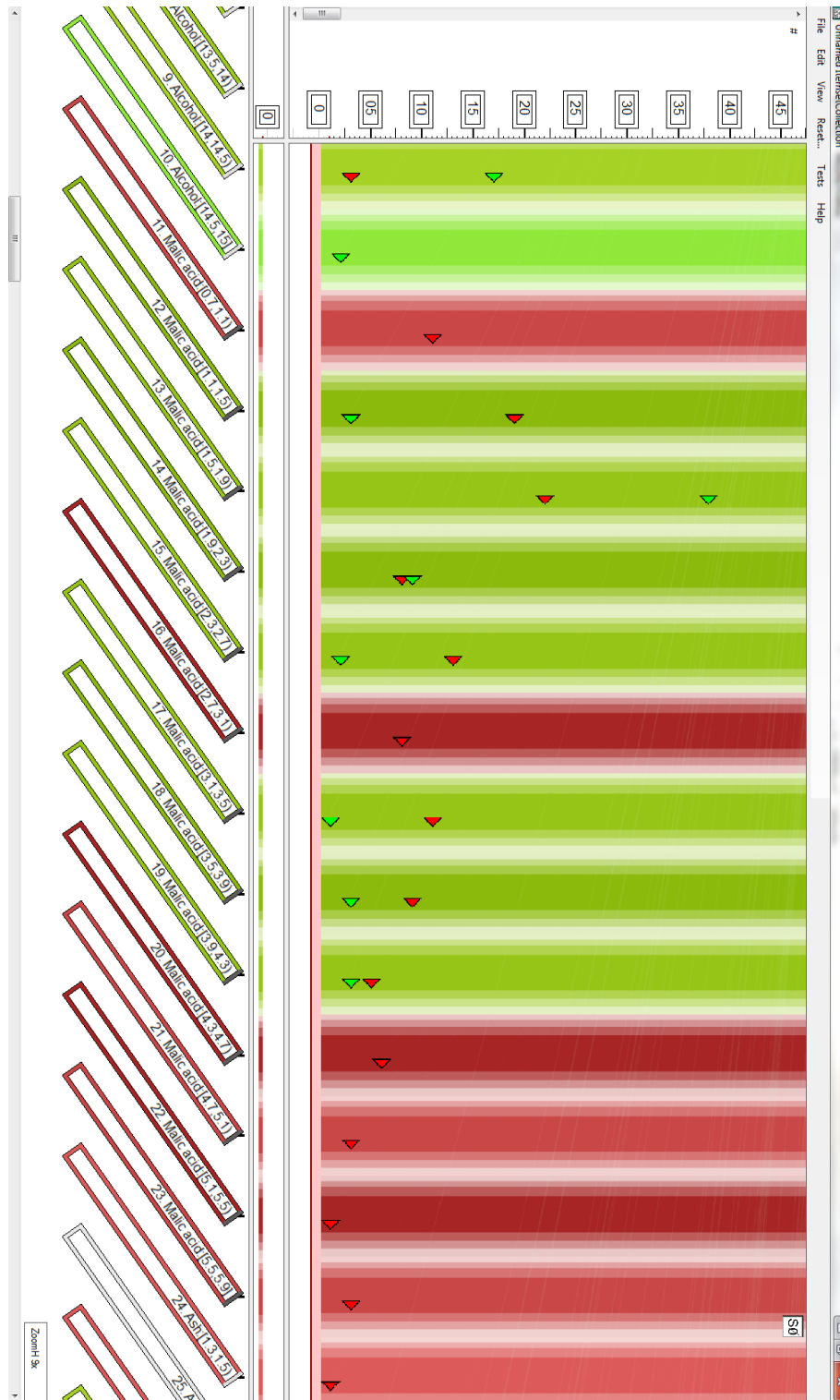


Figure 4.16: Wine data: malic acid wine class distribution.

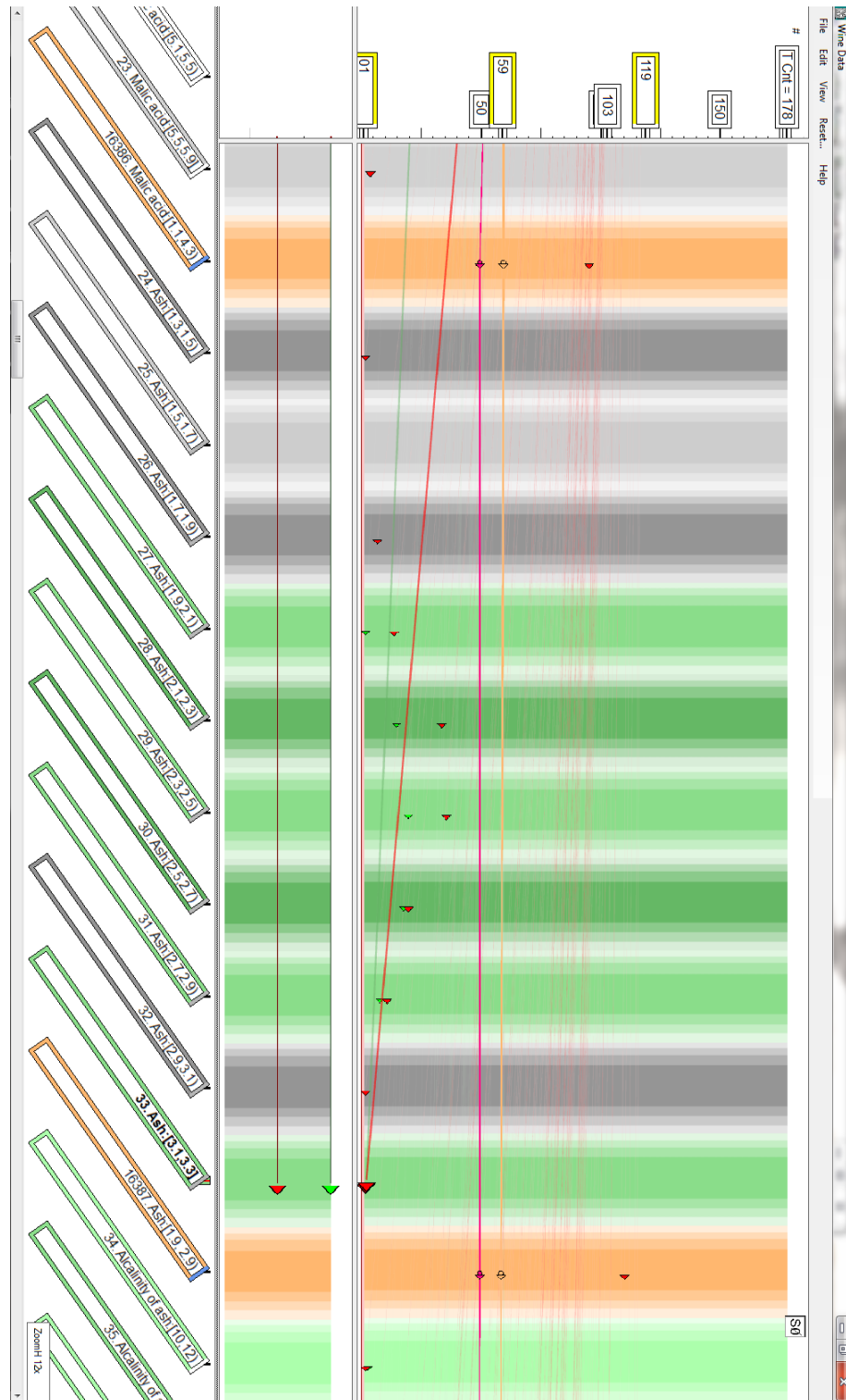


Figure 4.17: Wine data: ash outlier extensions.

wine. The aggregate item that is added spans the ash range [1.19, 2.9), missing one class 1 wine with an ash value in the [3.1, 3.3] range. We could choose to expand the aggregate range to [1.19, 3.3]. Doing this would include the missing class 1 wine but would also add 2 non-class 1 wines.

Figure 4.17 shows an EGraph, which displays the two nodes on the Ash[1.9, 2.9) column. A splitter on the application is then used to decide how to split the available screen space between the EGraph and the FGraph. If the FGraph can display the information adequately it is given 100% of the screen space. In this case, the frequency of both class 1 wines and non-class 1 wines that have an ash content in the [3.1,3.3] is 1. These nodes land on the exact same position in the FGraph, so it is necessary to include the EGraph in this figure to inform the reader there are two nodes on this column.

The next step in the task is to build two itemsets so we can compare the number of class 1 wines that have this constituent in the new aggregate item ranges to non-class 1 wines in these same ranges. See Figure 4.18. Every column that contains a class 1 wine is coloured green to show which ranged items were used to construct that aggregate. The orange itemset has a cardinality of 14 and contains the class 1 attribute plus all 13 new ranged aggregate items. This itemset has a frequency of 58, and covers all but 1 of the 59 class 1 wines. The one that is missing is the ash outlier. The other red itemset is a 13 cardinality itemset that contains the non-class 1 aggregate item and 12 of the new ranged aggregate items. Its only 12 because no non-class 1 wine has all 13 constituents in the aggregate ranges. The frequency of the red line is dropping because, as the items are added to the itemset from left to right,

more and more non-class 1 wines are filtered out from the classification.

The last step is to refine the order of items so that the red line drops as quickly as possible. The quicker it drops the more non-class 1 wines are filtered out. The idea is to find the fewest number of constituent aggregate items that brings the frequency of this line to 0 and thus reduces the number of constituents in the final classification. If it is possible to do this without using the ash constituent we can come up with a classification path that accounts for all 59 class 1 wines and contains no non-class 1 wines. Figure 4.19 now shows the frequency of the red itemset (now yellow because it was highlighted to show the actual frequencies values) dropping much quicker with the new order.

In summary, all class 1 wines have constituents within the following ranges; Proline:[650,1700], Flavanoids:[2,5.4), Total phenols:[2.15,3.9], Alcohol:[12.5,15], Malic acid:[1.1,4.3), Color intensity:[3,9), and Alcalinity of ash:[10,26). All 59 class 1 wines are included in this classification and all 119 non-class 1 wines are filtered out. As an association rule it has 100% confidence and a support of 33% of all 178 transactions.

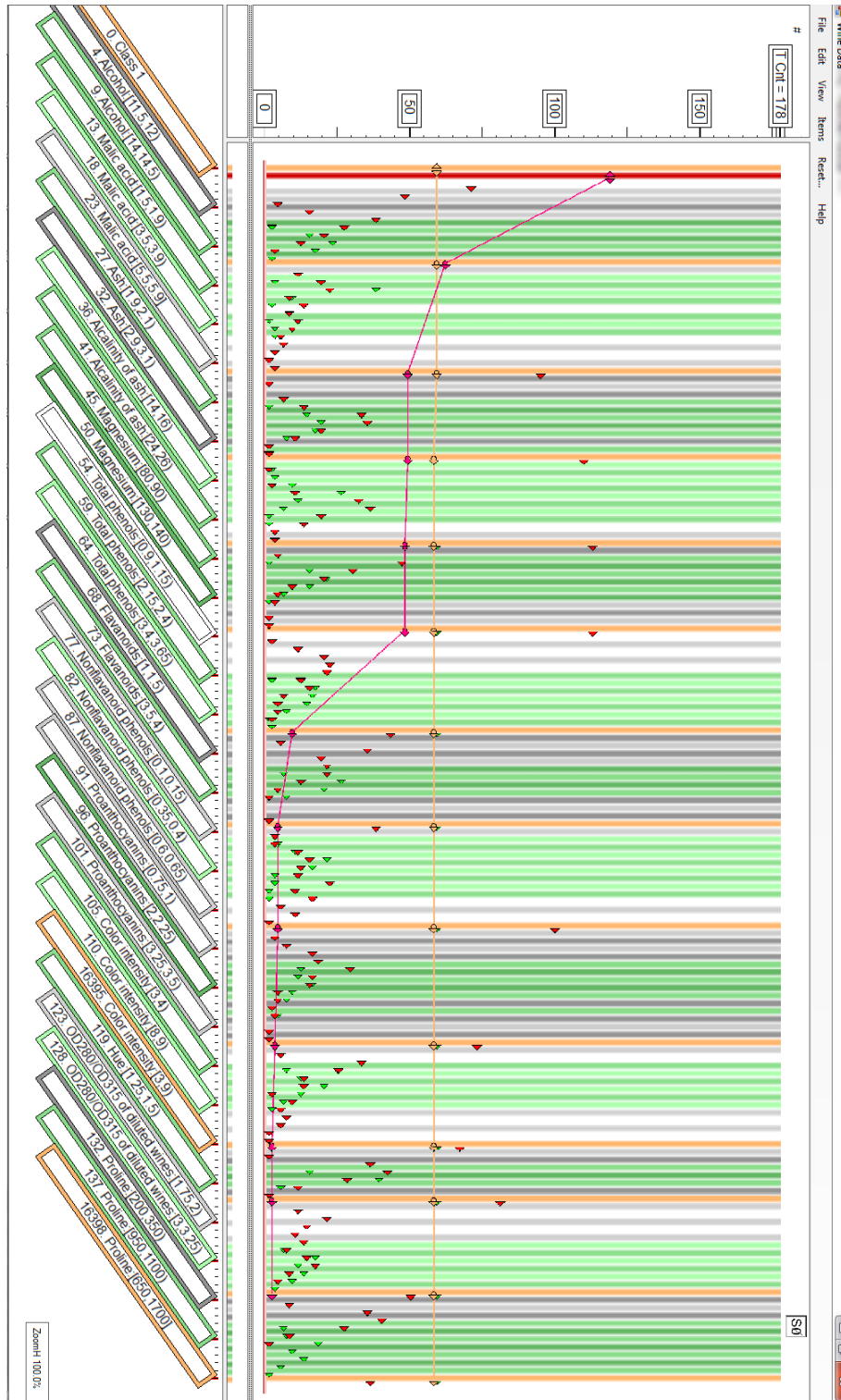


Figure 4.18: Wine data: class 1 aggregate ranges added

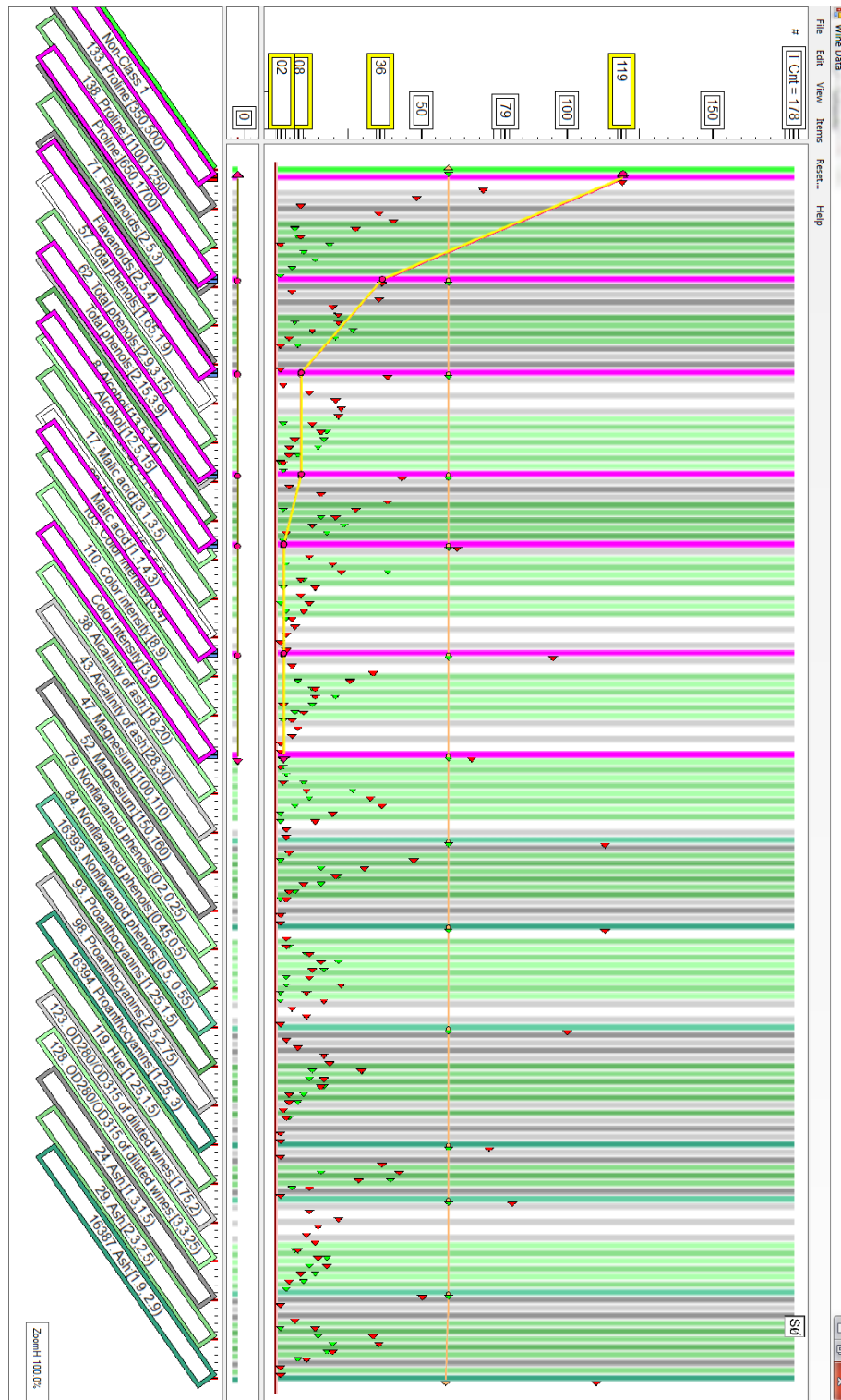


Figure 4.19: Wine data: refined order, class 1 aggregate ranges

4.3 Coauthorship Data

Along with itemset frequency, this system also has the ability to visualize and interact with transaction counts. Recall that each transaction in a database represents a single purchase consisting of several items that occur together with no other items. For instance, a transaction $\{a, b, c\}$ means that these three items occur together, but with no other items such as d . In this case study, each record represents a paper coauthored by several researchers. For instance, a record $\{\text{Alice, Bob, Carol}\}$ means that these three researchers coauthored a paper. This paper is written by these three researchers; there is no fourth author for this paper.

Here, we look at a collection of publications and compare the number of papers written by a group of authors in total to those papers written by that group alone. The works of three authors from the DBLP Computer Science Bibliography website have been combined into one data set. These transactions represent the papers written by Dr. Domaratzki, Dr. Irani, and Dr. Leung. In this database, the raw transaction cardinalities are relatively short but vary in length, as compared to both the mushroom data and the wine data which had fixed size lengths of 23 and 15 items long, respectively. In this data set transactions are taken from publications and represent authors that have worked together to co-author that publication. The size of author collaborations ranges from one to seven.

Along with the itemset information we have considered so far, many new questions can be answered with the inclusion of transaction counts. Examining this database, we answer the following questions;

1. “Who has collaborated most with Dr. Domaratzki?”

2. “Who (one or multiple authors) have collaborated with Dr. Leung?”,
3. “Who alone (i.e., only one author) has collaborated with Dr. Leung?”, and
4. “What is the typical size of Dr. Irani’s collaboration group?”

If enough information was available questions like these could be applied to any author. However this database is limited in its scope. Only papers written by the three main authors are contained. So if we were to determine that Dr. Leung has written a few papers with Jiang and myself, we can determine how many times the three of us have worked alone together, but we cannot determine how many times Jiang has worked alone or how many times Jiang and Carmichael have worked alone together. This short coming is due to the lack of information in the database. If this information was available, with this system we could determine these answers as well.

4.3.1 Overview of the Coauthorship Data

Figure 4.20 shows the raw authorship data. Authors are ordered alphabetically by first name. The three main authors are highlighted and, as we would expect, these stand out as the highest frequency nodes. The frequencies are 69, 59, and 67. This data base has 192 transactions made up of 140 different authors. Besides the three main authors highlighted one author stands out because their frequency of contribution is higher than the rest. Exploring this node leads to the answer to the first question.

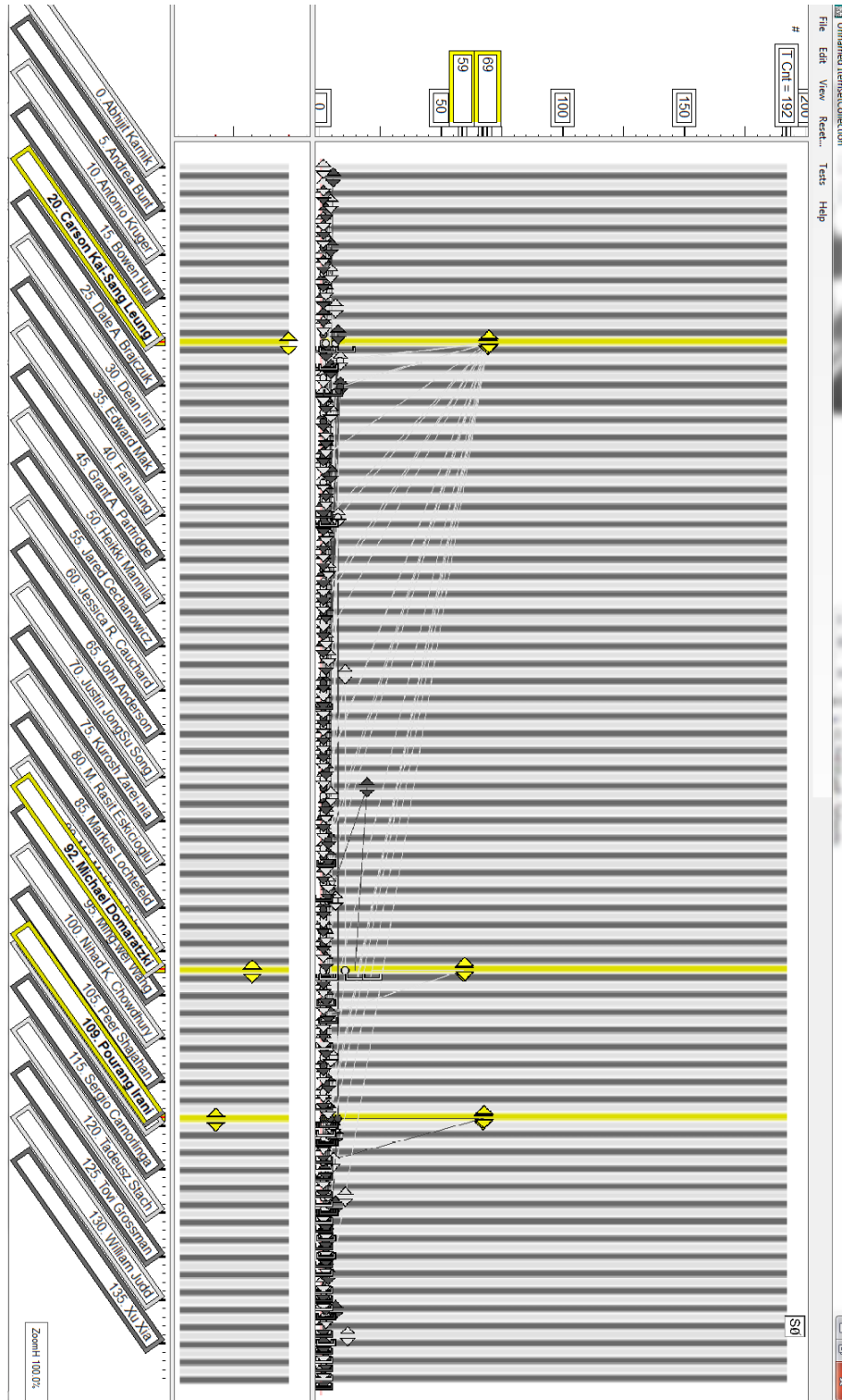


Figure 4.20: Coauthorship data overview.

4.3.2 Q1: Who has collaborated most with Dr. Domaratzki?

From Figure 4.20, a node stood out with a frequency higher than all other contributing authors. Figure 4.21 is a zoomed image of that node with its extensions highlighted. Here, we can see that this author, Salommaa, has collaborated with at least one of the main authors 19 times. Following the itemsets edges, we also see that Salommaa has worked with Dr. Domaratzki as a 2-researcher team (because this set ends in a transaction glyph) 14 times, and Salommaa has worked with Daley and Dr. Domaratzki 3 times. Working on the math we realize that two papers are missing. This system indicates that not all of Salommaa's publications have been considered because the contained node glyph, directly under Salommaa's singleton at frequency 19, is not highlighted. This node indicates that there is a superset containing Salommaa. This node was not highlighted because we only highlighted the *extensions* of Salommaa. If we adjust the filter to filter out all but those transactions that contain Salommaa, $l_{\text{hany}}(71)$, we can see all transactions in Figure 4.22. This can be accomplished without changing the order or grouping or even data mining. The EGraph displays the transactions of all the sets. Moving top to bottom on all of the transaction end glyphs in the Dr. Domaratzki column on the EGraph indicates the counts of each transaction in the FGraph. Okhotin, Salommaa, and Dr. Domaratzki have published together once. Rozenberg, Salommaa, and Dr. Domaratzki have published together once; Salommaa, Daley and Dr. Domaratzki have published together three times. Finally, there are 14 papers coauthored by only Daley and Dr. Domaratzki (with no additional coauthors).

To summarize, among all frequent patterns containing Dr. Domaratzki, the an-

swer to Q1 is the frequent pattern with the highest frequency (i.e., {Dr. Domaratzki, Salommaa}).

4.3.3 Q2: Who has collaborated with Dr. Leung?

This question was explored with some of the techniques used in the mushroom data set. Dr. Leung will be moved to the left and coloured red. To give some context of his relationship with other two main authors they will also be moved to the left. Dr. Domaratzki is the green column and Dr. Irani's column is coloured blue (but because he has collaborated with Dr. Leung, his column colour has been overridden by pink). A simple extension mining of Dr. Leung's work is constrained to produce only 2-sets. These are grouped and all columns in this group are coloured pink. All authors in this group have collaborated with Dr. Leung at least once and may have collaborated with Dr. Leung alone or with additional coauthors. 4.23.

To summarize, the answers to Q2 are all frequent patterns containing Dr. Leung (e.g., {Carmichael, Dr. Leung}, {Carmichael, Dr. Irani, Dr. Leung}, {Dr. Irani, Dr. Leung}).

4.3.4 Q3: Which individual researcher has collaborated with Dr. Leung?

This question can be answered very similar to Q3 but, instead of adding itemset extensions, we now add transactions extensions. Again, pink is used as the column colour. See Figure 4.24. In other words, the answers to Q3 are all frequent 2-itemsets containing Dr. Leung (e.g., {Carmichael, Dr. Leung}, {Dr. Irani, Dr. Leung}).

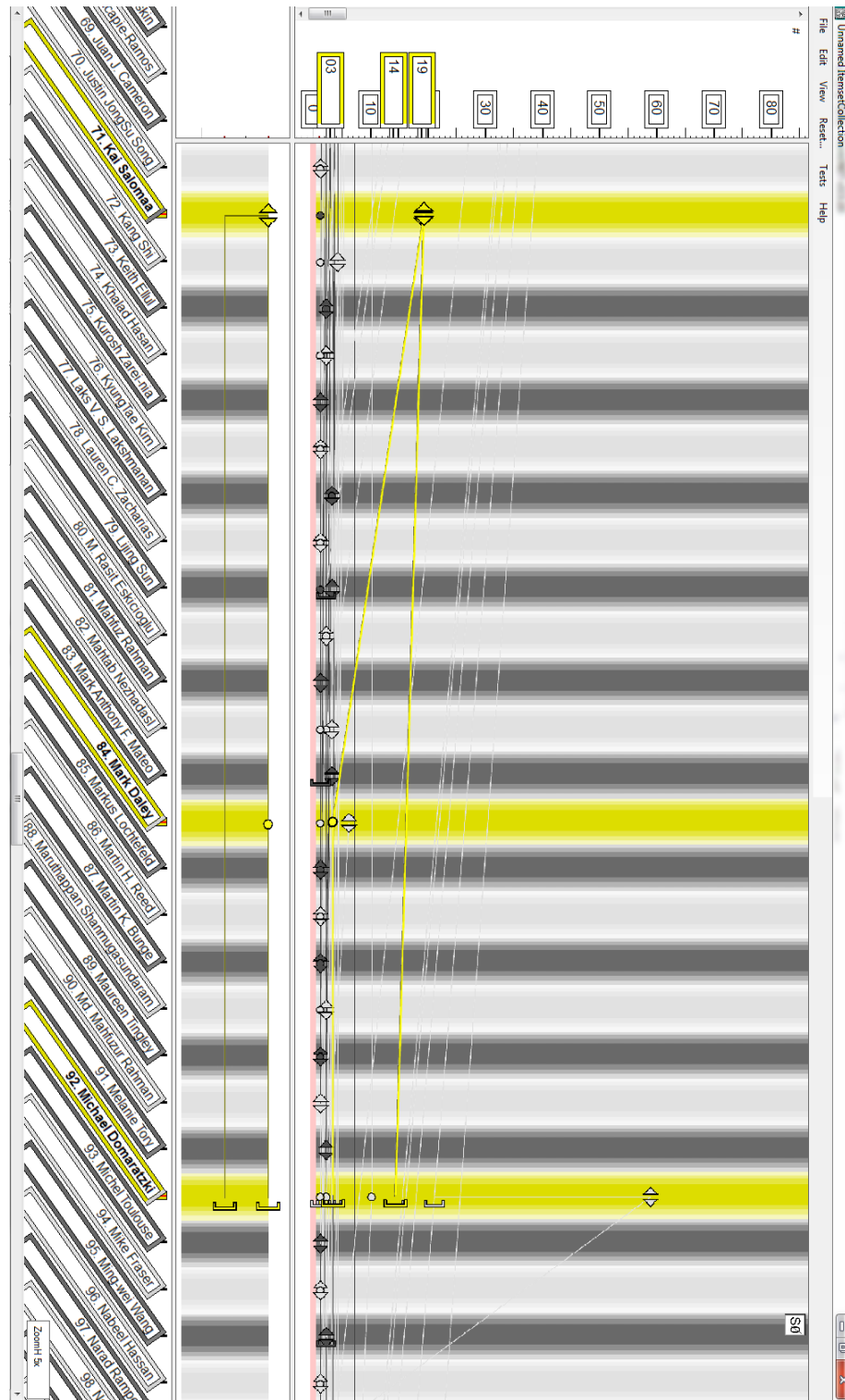


Figure 4.21: Q1: Who have collaborated most with with Dr. Domaratzki with?

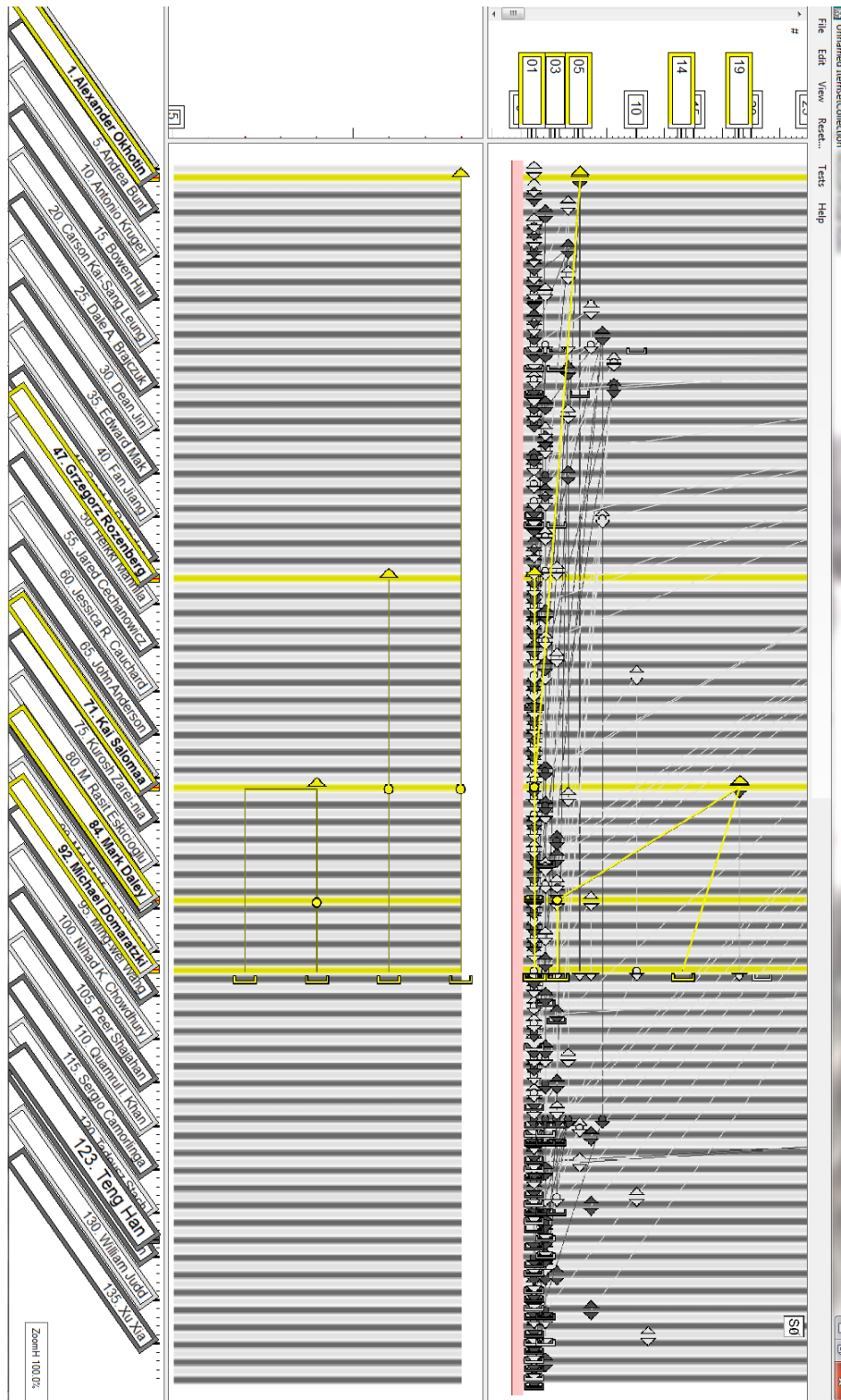


Figure 4.22: Salomea's publications.

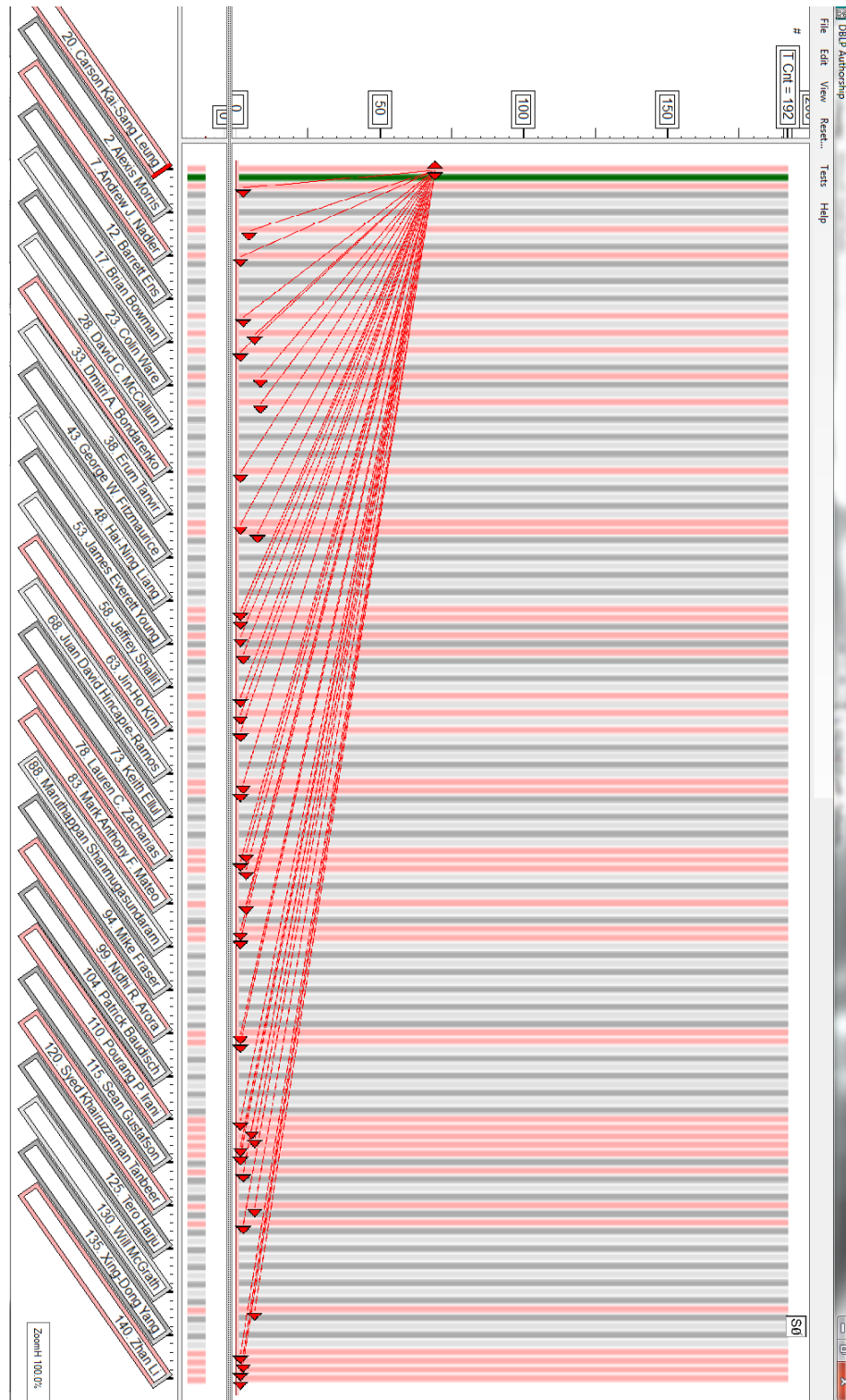


Figure 4.23: Q2: Who (one or multiple authors) have collaborated with Dr. Leung?

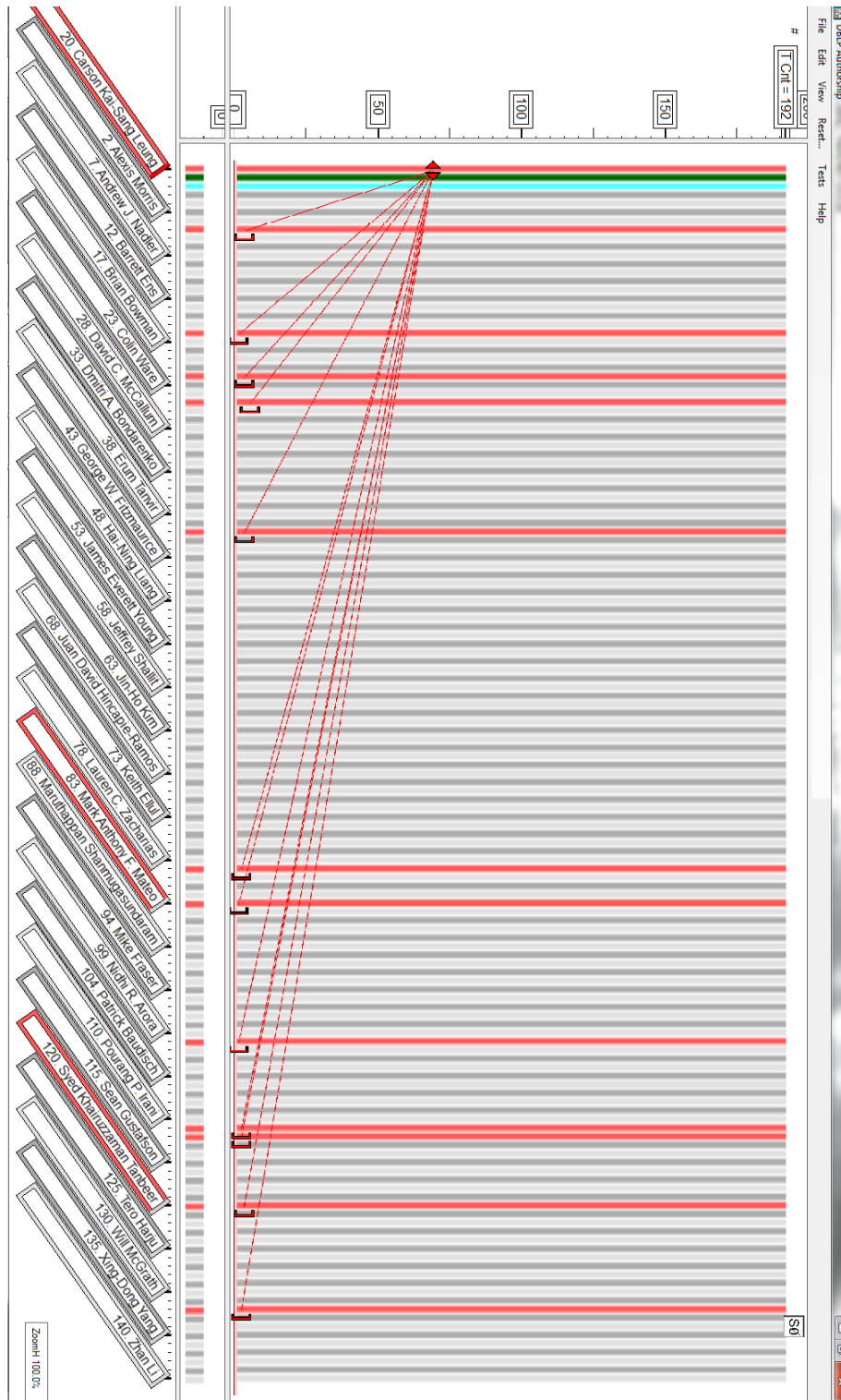


Figure 4.24: Q3: Which individual researcher has collaborated with Dr. Leung?

4.3.5 Q4: Compare the results to Q2 and Q3.

The results of Q1 and Q2 are brought together and shown at the same time, but we need to choose another colour. Purple is used to indicate which authors have collaborated with Dr. Leung but not alone. All authors resulting from Q1 and Q2 are moved to the left. Those authors that have collaborated with Dr. Leung alone are first followed by those that have not. In this figure, we can see both frequency and transaction counts for each author. Moving the mouse over the nodes, it highlights their exact frequency. Looking at the two highest nodes (highlighted yellow), we see that Dr. Leung has collaborated with Brajczuk and Carmichael (myself) the most, 8 times each. Dr. Leung has collaborated alone with Brajczuk 5 times and only 3 times with Carmichael.

To summarize, the answers to Q3 are subsets of answers to Q2. To elaborate, the latter include all frequent 2-itemsets that is a superset of {Dr. Leung}, whereas the former include all frequent k -itemsets (where $k \geq 2$) that is a superset of {Dr. Leung}.

4.3.6 Q5: What size of group does Dr. Irani usually participate in?

This can be accomplished by searching for all of the people that Dr. Irani has collaborated with, colouring them blue, and moving them to the left. A new aggregate item is added to represent collaborators group size or cardinality. This is accomplished by using the group string “card() == 1” for the group size of 1, “card() == 2”. These aggregate items are moved just right of collaborating authors and coloured purple to

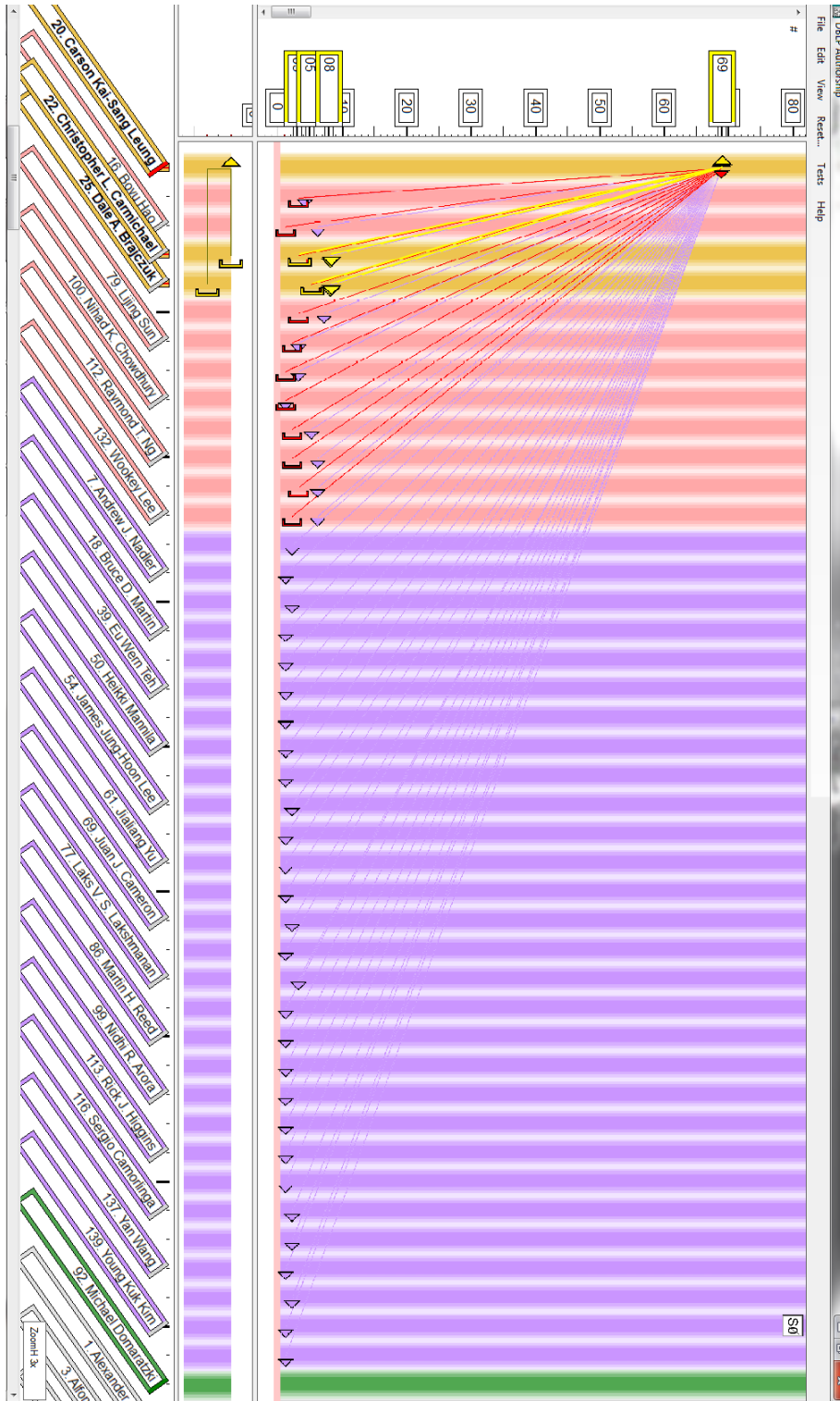


Figure 4.25: Q4: Compare the results to Q2 and Q3

stand out. Figure 4.26 is the result. Dr. Irani usually collaborate in a group size of 3 or 4 with 25 and 23 publications in each group size respectively.

To summarize, among all transactions containing Dr. Irani, the answer to Q5 is the most popular transaction cardinality (i.e., 3 researchers).

4.3.7 Q6: Of the authors from Q5, who did Dr. Irani collaborate with and how often did he do it?

See Figure 4.27. Looking at a group size of 3, there are many authors in this group. To find them all, we find all the transactions that contain both Dr. Irani and the aggregate item “group size = 3”. These itemsets and name labels are highlighted yellow and provide the answer. We can zoom in. The EGraph has been expanded and one itemset has been highlighted in red. This itemset contains publications of Dr. Irani, Shi, and Subramanian. The frequencies on the left of the FGraph represent how many publications Dr. Irani has, 67, how many Dr. Irani and Shi have together, 4, and how many Dr. Irani, Shi, and Subramanian have, 2, and how many Dr. Irani, Shi, and Subramanian have *alone* together, 1. At this scale, the frequencies on the FGraph overlap. However, when the mouse moves to each node in the path, the frequency of that node is drawn on top of the others. In other words, the exact frequencies are clearly shown in the dynamic display.

To summarize, the answers to Q6 include all the transactions consisting of Dr. Irani and his two other collaborators (e.g., {Dr. Irani, Shi} with frequency of 4).

The final display can be cleared up by moving the new “group size 3” aggregate item to the left along with all authors that have collaborated with Dr. Irani in a

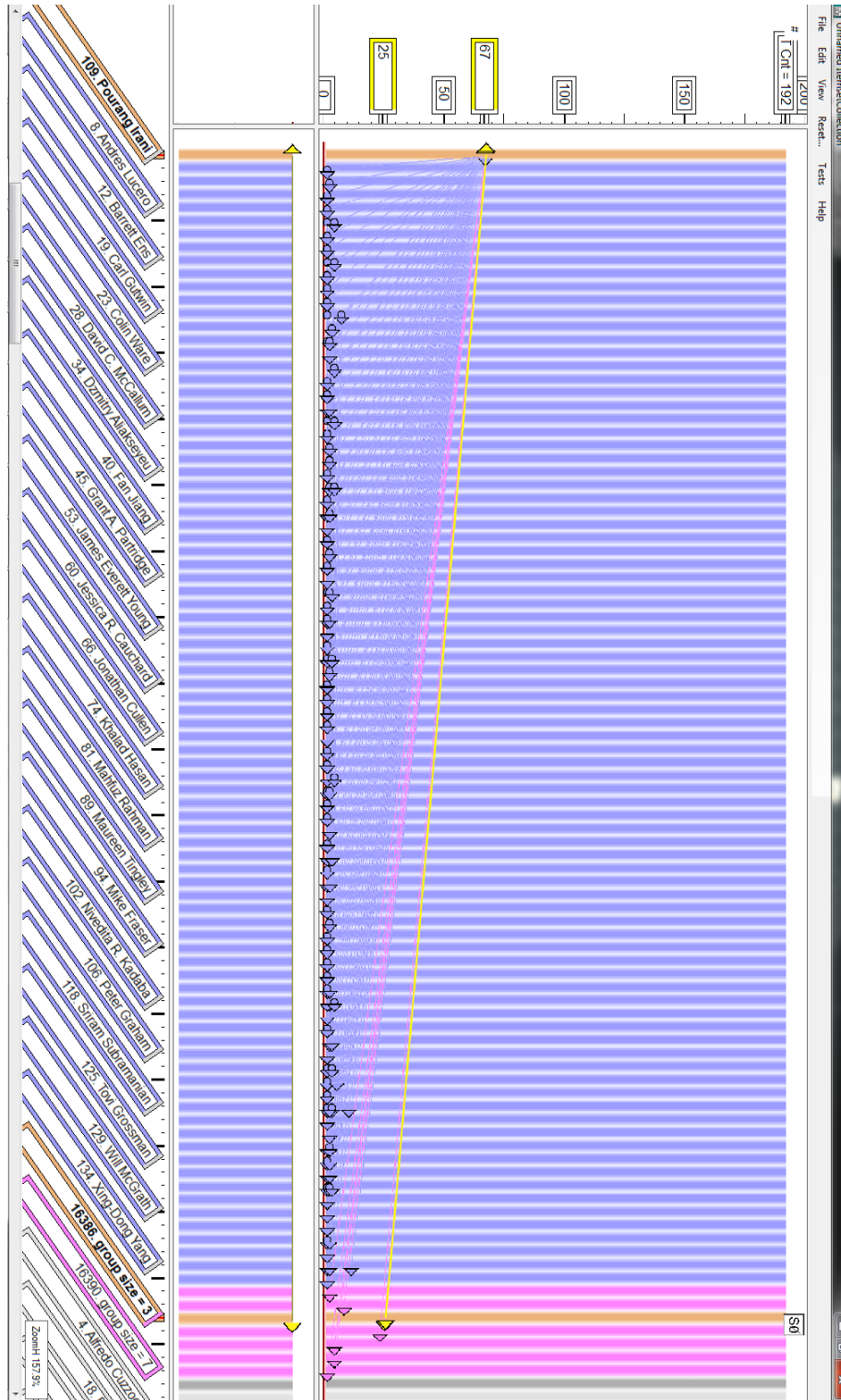


Figure 4.26: DBLP: Compare Q1 and Q2

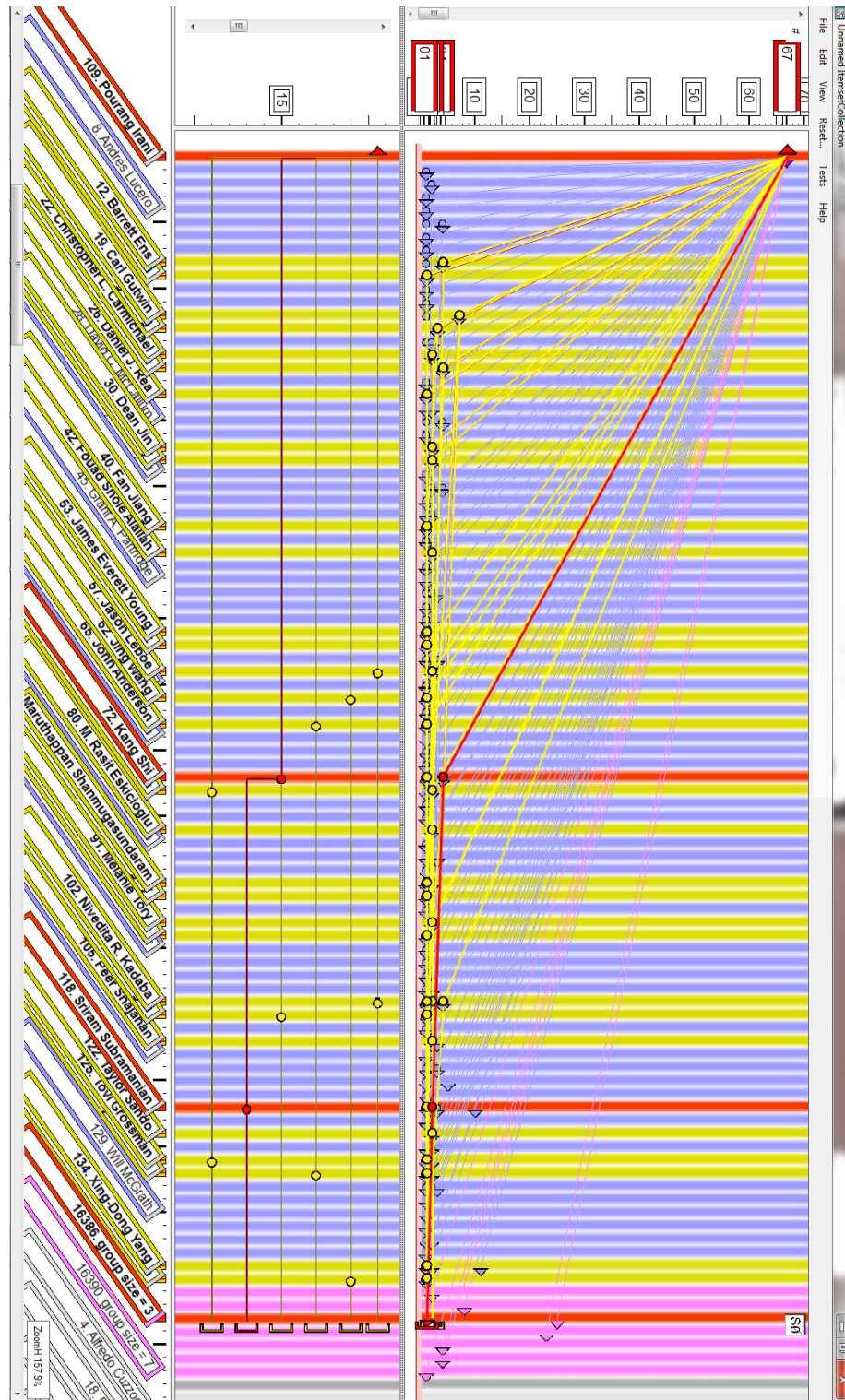


Figure 4.27: Q6: Of the authors from Q5, who did Dr. Irani collaborate with and how often did he do it?

group of this size. All of these items are coloured blue, and all other itemsets are removed. From this figure, we can see how many times each three set has occurred from the height of each end transaction glyph. We can also probe for details as to who was in each group, like what was done for the author Kadabe. Highlighted in yellow are all the size 3 groups she participated in with Dr. Irani.

4.4 Summary

In this chapter, we evaluated our visualization system by using real-life datasets (mushroom dataset, wine dataset, and selected DBLP dataset). By using the system, we identified all mushrooms attribute sets that only occur on edible mushrooms or only on poisonous mushrooms.

Our system converts an analog database into transactions. Each item represents a range of constituents. By using the system, we classified class 1 wines. The constituents are then reordered to show a more efficient classification that eliminates more non-class 1 wines with fewer constituents.

The final database is an authorship database extracted from the DPLP website. Three University of Manitoba authors were chosen and their publications were collected in a transactions database. We used our visualization system to study several questions regarding the co-author relationships. In general, our visualization system is capable of (a) finding the number of papers published by any researcher, (b) verifying whether or not a group of k specific researchers collaborated together, (c) finding the number of paper coauthored by a group of k specific researchers, (d) revealing the complete author list for a specific paper, (e) counting the number of authors of a



Figure 4.28: Dr. Irani 3-set collaborations

specific paper, (f) counting the numbers of papers coauthors by some or all authors in a specific groups of researchers.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Data mining algorithms mine itemsets to discover implicit, previously unknown and potentially useful knowledge. Frequent pattern mining is used to extract patterns of that occur often in data. These patterns are in the form itemsets or database items that have occurred together. They can occur together on some objects (e.g., attributes on a mushroom) or event (e.g., SIGMOD and PODS conference collocated). Comparing frequencies of related itemsets provides associate information between them. If someone purchases apples, how likely would they also purchase bananas? If a mushroom has a certain set of attributes, how likely is that mushroom edible? What constituents make a better wine? Applying data mining techniques helps to find important hidden relationships between medications and diseases.

Many of the frequent pattern mining algorithms represent the discovered knowledge in the form of a long textual list of itemsets. The number of itemsets can be huge

as a transaction of n items can generate $2^n - 1$ different itemset. As an alternative, a visualization system can be used to organize and present the data in an easy-to-understand way so that users can acquire knowledge contained in the raw transaction data or the processed mined itemsets.

Here, in my M.Sc. thesis, I represented the discovered knowledge in pictorial form. I developed an easy-to-understand interactive visual system to display and explore frequent itemsets mined from a database. Finding the right visualizer to present this itemset information had four main challenges:

- Find a way to present millions of itemsets
- Present the information in context
- Provide a dialogue so the user can change the context base on the information they are interested in.
- Provide a way to compare and contrast different subset of the data presented

In this thesis, we developed a visualization system, which consists of two graphs. The FGraph displays millions of itemsets at one time. This graph provides a great overview perspective. It is also good at providing the details of just a few itemset. The EGraph displays the details of a few itemsets within a limited context like a mouse selection. The combination of these two graphs makes it possible to see an overview of millions of itemsets and transactions as well as probe the details of just a few.

Specifically, Chapter 2 provided the background and related works. The concepts of data mining and data visualization were introduced. We reviewed the concepts of

transactions, itemsets, and association rules. We also discussed Mackinlay's ranked perceptual tasks and the importance of mapping the most important information to the correct task. In our visualizer, we use position to represent items within an itemset and its frequency.

Chapter 3 introduced our visualization system. The basic representation of itemsets uses four glyphs to describe where itemsets starts, which intermediate items are contained, where an itemset ends, and where a transaction ends. Different visual compression techniques are demonstrated. These itemsets are presented in two graphs. The FGraph gives an overview of millions of itemsets. Any itemset collection mined from a transaction database of n items and m transactions is displayed on a $n \times m$ visual space that maps items to the x -position and the frequency the y -position. This mapping allows the user to infer association rule information like rule support and confidence. This mapping also provides much information like how many transactions do not contain a given item or items, for example. The overall representation used a prefix/extension paradigm that provides context as far as which items are more important. The EGraph complements the FGraph by providing information as to exactly which itemsets are contained in a given context like mouse selection.

Many interactive features are also provided to the user so they can interactively discover and display the exact information. Basic feature of selecting specific data is provided with a mouse selection. This feature is enhanced by the options of adding specific extensions or by filtering. Filtering can be done based on simple ideas like minimum support or cardinality or items contained. A user-defined equation can be used for more advanced criteria like specifying only those consequences of a rule that

a have a confident and a support over a given minimum threshold. Grouping itemsets and transactions gives the user the options of contrasting different groups by changing the colour and sizes of glyphs and columns.

Chapter 4 presents three case studies. (1) The mushroom dataset is a benchmark database for data mining. Using this database we explored all of the attributes that occur on edible or poisonous mushrooms. Taking control of the constraint base mining engine we showed how some areas can be mined using a minimum support of 1. Using itemset selection and grouping we could compare and contrast those attributes that only occurred on edible mushrooms, to those that only occurred on poisonous mushrooms, to those that occur on both poisonous and edible mushrooms. Aggregate items were introduced to divide the database based on which group they belonged to. (2) The wine data showed how an analog database could be discretized and how the system as a whole could be used as a visual classifier. (3) Finally, the authorship database showed relationships among authors. The authors of several publications were studied and we were able to explore questions pertaining to who collaborated with who, how often a group of authors worked together with other collaborators and how often they worked alone. We were even able to determine typical group sizes and who participated in groups of that size.

Overall, this system displays items and mines hidden knowledge contained in it.

5.2 Future Work

Future work will include user feedback and users studies to make iterative improvements. Three major additions will be to include an option for (1) automatic

aggregate item mining, (2) user guided transactions count adjustments, and (3) improvements to the itemset search tool. Future work may also extend the visualization to display graphs instead of just itemsets.

5.2.1 Aggregate Item Automatic Mining and Grouping

While working with this application a common pattern of exploration emerged. After a set of data is mined, grouped and evaluated, a new aggregate item is added to contrast one set of data from another. From there it is quite often necessary to re-mine the data to include the new aggregate item. A time-saving feature would be to provide the user the option to automatically perform this task. The program would look at all the items included in the construction of the aggregate item, look at all the itemsets mined for these existing items and then mine the new item in a similar manner. However, the “similar manner” is still undefined. For all itemsets that contain an existing item, sometimes it was necessary to know how many times those items occur with the new aggregate item. In other cases, to come up an appropriate aggregate item the user followed steps *A*, *B*, and *C* and only step *C* needs to be applied to the new item. For example, the user may have mined the extensions of item *a*, then mine the extension of *b*, then mine the extensions of item *c* which finally leads them to create the new aggregate item. Only the extensions of *c* that contain the new aggregate item are required. The problem will be to come up with a set of options and to communicate these to the user. Once this is known the new itemset can be mined and grouped automatically.

5.2.2 Transaction Counts

When removing items from transactions, their transaction counts decrease. The problem is determining how the user expects them to change. For example, in the authorship case study we may want to contrast all the 2-authors papers written by Dr. Leung and Carmichael alone to those 2-author papers written by Dr. Leung and someone else alone. As soon as we add the aggregate item “!Carmichael” to the 2-item transactions that contains the items “Dr. Leung” and not “Carmichael”, they become 3-item transactions. All the transaction counts for the 2-author papers written by Dr. Leung and someone else alone are now zero. Is this what the user expected? Should another transaction count value be added to the underlying itemset-tree to store the original value so it can still be displayed? Are there cases where the transaction count should be changed? Should a flag be provided to the user to indicate whether or not the item should be counted in the transaction count? There are still quite a few things to consider when altering the database.

5.2.3 Itemset Search

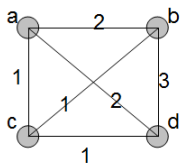
While working with itemsets there is often a need to find a itemsets that fit a variety of different criteria. For example, in the wine case study it was necessary to find the shortest item path where the frequency dropped to zero with least number of items. Furthermore, there are many different and complex association rule measures. The current search engine can search for the consequence of association rules base on support or confidence because it has the ability to build equations base on an itemset and any one of its prefixes. Future work will include user generated equations based

on subsets and supersets, not just prefixes.

5.2.4 Generalization of the Visualization for Presenting Other Graphs

The solution presented here could be extended to visualizing any non-directional connected graphs, $G = (V, E)$, where each vertex $v \in V$ can be named, ordered and has a value $r \in \mathbb{R}$. Each path in the tree that continuously flows from a lower ordered vertex to a higher one would be considered a transaction. Along the x-axis, vertex names would replace item and the heights of nodes could be the accumulated path values or some other meaningful function that describes a value at that point in the vertex path. Figure 5.1 shows an example of what this may look like.

Graph with ordered vertices:



Graph paths to transactions:

- {a}:0 {b}:0 {c}:0 {d}:0
- {a,b}:2 {b,c}:1 {c,d}:1
- {a,b,c}:3 {b,c,d}:2
- {a,b,c,d}:4 {b,d}:3
- {a,b,d}:5
- {a,c}:1
- {a,c,d}:2
- {a,d}:2

Future graph visualization:

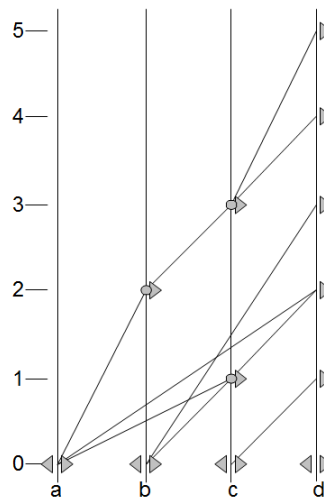


Figure 5.1: Possible graph visualization

Bibliography

- [Ahl96] Christopher Ahlberg. Spotfire: an information exploration environment. *ACM SIGMOD Record*, 25(4):25–29, 1996.
- [AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216. ACM Press, 1993.
- [AR11] Rajul Anand and Chandan K. Reddy. Constrained logistic regression for discriminative pattern mining. In *Proceedings of European Conference on Machine Learning / Principles of Data Mining and Knowledge Discovery (ECML/PKDD)*, pages 92–107. Springer, 2011.
- [AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, pages 487–499. Morgan Kaufmann Publishers Inc., 1994.
- [BJ01] Jean-François Boulicaut and Baptiste Jeudy. Mining free itemsets under constraints. In *Proceedings of International Database Engineering & Applications Symposium (IDEAS)*, pages 322–329. IEEE Computer Society, 2001.
- [BJR00] Stefan Berchtold, H. V. Jagadish, and Kenneth A. Ross. Independence diagrams: A technique for data visualization. *Journal of Electronic Imaging*, 9(4):375–384, 2000.
- [CF11] Xu Chi and Zhang Wen Fang. Review of association rule mining algorithm in data mining. In *Proceedings of the Third IEEE International Conference on Communication Software and Networks (ICCSN)*, pages 512–516. IEEE Computer Society, 2011.
- [CHL11] Christopher L. Carmichael, Yaroslav Hayduk, and Carson Kai-Sang Leung. Visually contrast two collections of frequent patterns. In *Workshops*

- Proceedings of the 11th IEEE International Conference on Data Mining (ICDM)*, pages 1128–1135. IEEE Computer Society, 2011.
- [CL10] Christopher L. Carmichael and Carson Kai-Sang Leung. Closeviz: visualizing useful patterns. In *Proceedings of the ACM SIGKDD Workshop on Useful Patterns (KDD-UP)*, pages 17–26. ACM Press, 2010.
- [FA13] A. Frank and A. Asuncion. UCI machine learning repository, University of California, Irvine, CA, USA, 2013.
- [FPM91] W.J. Frawley, G. Piatetsky-Shapiro, and C.J. Matheus. *Knowledge Discovery in Databases: An Overview*, pages 1–27. AAAI/MIT Press, Menlo Park, CA, USA, 1991.
- [Fri05] M. Friendly. Milestones in the history of data visualization: A case study in statistical historiography. In *Proceedings of the 28th Annual Conference of the Gesellschaft für Klassifikation*, pages 34–52. Springer, 2005.
- [FTM⁺11] Robson Leonardo Ferreira Cordeiro, Caetano Traina, Junior, Agma Juci Machado Traina, Julio López, U. Kang, and Christos Faloutsos. Clustering very large multi-dimensional datasets with MapReduce. In *Proceedings of the 17th ACM Special Interest Group on Knowledge Discovery and Data Mining International Conference on Knowledge Discovery and Data Mining*, pages 690–698. ACM Press, 2011.
- [FWR99] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of Visualization (VIS)*, pages 43–50. IEEE Computer Society, 1999.
- [GH06] Liqiang Geng and Howard J. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys*, 38(3), September 2006.
- [HAC00] Jianchao Han, Aijun An, and Nick Cercone. CViz: An interactive visualization system for rule induction. In *Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence (Canadian AI)*, pages 214–226. Springer, 2000.
- [HC99] Jianchao Han and Nick Cercone. DVIZ: A system for visualizing data mining. In *Proceedings of the Third Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 390–399. Springer, 1999.
- [HC00] Jianchao Han and Nick Cercone. AViz: A visualization system for discovering numeric association rules. In *Proceedings of the Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 269–280. Springer, 2000.

- [HLD02] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *Proceedings of IEEE Symposium on Information Visualization (InfoVis)*, pages 127–130. IEEE Computer Society, 2002.
- [HPYM04] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, January 2004.
- [ID90] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the First IEEE Conference on Visualization*, pages 361–378. IEEE Computer Society, 1990.
- [Kei00] D.A. Keim. Designing pixel-oriented visualization techniques: theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, January 2000.
- [Kei02] Daniel A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, January–March 2002.
- [KK94] Daniel A. Keim and Hans-Peter Krigel. VisDB: Database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications*, 14(5):40–49, September 1994.
- [Kon06] Qiang Kong. Visual mining of powersets with large alphabets. Master’s thesis, Department of Computer Science, The University of British Columbia, Vancouver, BC, Canada, 2006.
- [LC09a] Carson Kai-Sang Leung and Christopher L. Carmichael. FpVAT: a visual analytic tool for supporting frequent pattern mining. *ACM SIGKDD Explorations*, 11(2):39–48, 2009.
- [LC09b] Carson Kai-Sang Leung and Christopher L. Carmichael. FpViz: a visualizer for frequent pattern mining. In *Proceedings of the First ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery (KDD-VAKD)*, pages 30–39. ACM Press, 2009.
- [LC10] C.K.-S. Leung and C.L. Carmichael. Exploring social networks: A frequent pattern visualization approach. In *Proceedings of the IEEE Second International Conference on Social Computing (SocialCom)*, pages 419–424. IEEE Computer Society, 2010.
- [LC11] Carson Kai-Sang Leung and Christopher Carmichael. iVAS: An interactive visual analytic system for frequent set mining. In *Visual Analytics*

- and Interactive Technologies: Data, Text and Web Mining Applications*, pages 213–231. IGI Global, 2011.
- [LCH07] Carson Kai-Sang Leung, Christopher L. Carmichael, and Boyu Hao. Efficient mining of frequent patterns from uncertain data. In *Workshops Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)*, pages 489–494. IEEE Computer Society, 2007.
- [LCT11] Carson Kai-Sang Leung, Christopher L. Carmichael, and Eu Wern Teh. Visual analytics of social networks: Mining and visualizing co-authorship networks. In *Proceedings of the Sixth Conference on Foundations of Augmented Cognition held as Part of HCI International 2011 (HCII-FAC)*, pages 335–345. Springer, 2011.
- [LIC08a] Carson Kai-Sang Leung, Pourang P. Irani, and Christopher L. Carmichael. FIsViz: A frequent itemset visualizer. In *Proceedings of the 12th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 644–652. Springer, 2008.
- [LIC08b] Carson Kai-Sang Leung, Pourang P. Irani, and Christopher L. Carmichael. WiFIsViz: Effective visualization of frequent itemsets. In *Proceedings of the Eighth IEEE International Conference on Data Mining (ICDM)*, pages 875–880. IEEE Computer Society, 2008.
- [LJ12] Carson Kai-Sang Leung and Fan Jiang. RadialViz: An orientation-free frequent pattern visualizer. In *Proceedings of the 16th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 322–334. Springer, 2012.
- [LJI11] Carson Kai-Sang Leung, Fan Jiang, and Pourang P. Irani. FpMapViz: A space-filling visualization for frequent patterns. In *Workshops Proceedings of the 11th IEEE International Conference on Data Mining (ICDM)*, pages 804–811. IEEE Computer Society, 2011.
- [LJSW12] Carson Kai-Sang Leung, Fan Jiang, Lijing Sun, and Yan Wang. A constrained frequent pattern mining system for handling aggregate constraints. In *Proceedings of the 16th International Database Engineering & Applications Symposium (IDEAS)*, pages 14–23. ACM Press, 2012.
- [LLN03] Laks V. S. Lakshmanan, Carson Kai-Sang Leung, and Raymond T. Ng. Efficient dynamic mining of constrained frequent sets. *ACM Transactions on Database Systems*, 28(4):337–389, December 2003.
- [LWW90] J. LeBlanc, M.O. Ward, and N. Wittels. Exploring N-dimensional databases. In *Proceedings of the First IEEE Conference on Visualization*, pages 230–237. IEEE Computer Society, 1990.

- [Mac86] Jock Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, April 1986.
- [MKN⁺05] T. Munzner, Q. Kong, R.T. Ng, J. Lee, D. Radulovic J. Klawe, and C.K.-S. Leung. Visual mining of power sets with large alphabets. Technical report, UBC CS TR-2005-25, Department of Computer Science, The University of British Columbia, Vancouver, BC, Canada, December, 2005.
- [PG88] R.M. Pickett and G.G. Grinstein. Iconographic displays for visualizing multidimensional data. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics (ICSMC)*, pages 514–519. IEEE, 1988.
- [STH02] Chris Stolte, Diane Tang, and Pat Hanrahan. Query, analysis, and visualization of hierarchically structured data using polaris. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 112–122. ACM Press, 2002.
- [SVA97] Ramakrishnan Srikant, Quoc Vu, and Rakesh Agrawal. Mining association rules with item constraints. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 67–73. AAAI Press, 1997.
- [WAT11] Leland Wilkinson, Anushka Anand, and Dang Nhon Tuan. CHIRP: a new classifier based on composite hypercubes on iterated random projections. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 6–14. ACM Press, 2011.
- [WCF⁺00] Pak Chung Wong, Wendy Cowley, Harlan Foote, Elizabeth Jurrus, and Jim Thomas. Visualizing sequential patterns for text mining. In *Proceedings of IEEE Information Visualization 2000*, pages 105–111. IEEE Computer Society, 2000.
- [Yan05] Li Yang. Pruning and visualizing generalized association rules in parallel coordinates. *IEEE Transactions on Knowledge and Data Engineering*, 17(1):60–70, January 2005.