FUSION: A VISUALIZATION FRAMEWORK FOR INTERACTIVE ILP RULE MINING WITH APPLICATIONS TO BIOINFORMATICS

Kiran Kumar Indukuri

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

MASTER OF SCIENCE

in

Computer Science and Applications

_____
Chris North

_____ _____
Lenwood S. Heath Ruth Grene

_____
Naren Ramakrishnan

December 2004,
Blacksburg, Virginia

Keywords: Visualization, ILP Rule Mining, Microarrays, Fusion, Snap.

FUSION: A VISUALIZATION FRAMEWORK FOR INTERACTIVE ILP RULE MINING WITH APPLICATIONS TO BIOINFORMATICS

Kiran Kumar Indukuri

ABSTRACT

Microarrays provide biologists an opportunity to find the expression profiles of thousands of genes simultaneously. Biologists try to understand the mechanisms underlying the life processes by finding out relationships between gene-expression and their functional categories. Fusion is a software system that aids the biologists in performing microarray data analysis by providing them with both visual data exploration and data mining capabilities. Its multiple view visual framework allows the user to choose different views for different types of data. Fusion uses Proteus, an Inductive Logic Programming (ILP) rule finding algorithm to mine relationships in the microarray data. Fusion allows the user to explore the data interactively, choose biases, run the data mining algorithms and visualize the discovered rules. Fusion has the capability to smoothly switch across interactive data exploration and batch data mining modes. This optimizes the knowledge discovery process by facilitating a synergy between the interactivity and usability of visualization process with the pattern-finding abilities of ILP rule mining algorithms. Fusion was successful in helping biologists better understand the mechanisms underlying the acclimatization of certain varieties of Arabidopsis to ozone exposure.

ACKNOWLEDGEMENTS

I thank my advisor, Dr. North for his consistent guidance and constant encouragement through out the development of the Fusion system. Dr. North has always brought in fresh ideas and continuously reminded me of the 'user' aspect of the system. I thank him for the support he has given me during the tough times. He has helped me capture all the thinking and to put it together into a Thesis document by continuous reviewing and feedback.

I thank Dr. Ramakrishnan and Deept Kumar for sharing with us the Proteus algorithm and for their assistance on the project. My thanks to the Expresso group for sharing their micro array data. I thank Dr. Grene and Dr. Heath for their guidance and comments at various stages of the Fusion project. I thank Shrinivasrao Mane and Cecilia Vasquez-Robinet for regular feedback and coordination. I thank Allan Sioson for contributions to the development of Fusion. I thank Dr. Bohnert and Pinghua Li of the University of Illinois for sharing their data with me.

My thanks to Nathan Conklin for helping me get started on Snap, which later evolved into the Fusion system. I thank Varun Saini for contributions to the Fusion system. I also thank Ranjit Randhawa, Nupur Pande, Ying Chen, Kibum Kim and Youngyun Chung Baek for contributions to different components of the Fusion system.

I thank my previous employers Dr.Holly Bender, Dr.Pam Vermeer, Dr. Jared A. Danielson, Dr. Mills , Dr.North , Mr. Ken McCrery for giving me an opportunity to work for them. The financial support I have got was crucial for supporting my studies at Virginia Tech. You have given me very flexible work schedules and a good work environment.

I thank my parents and my brother for all they have given me. 'Thanks' is too small a word to express my gratitude. I thank Jeshua Pacifici, Ananta and Dada for the wisdom they have shared with me. I thank my friends in Blacksburg who made life here an enjoyable one.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

Chapter 1

1 Introduction and Motivation

Cells are the fundamental working units of every living system. All the instructions needed to direct their activities are contained within the chemical DNA (deoxyribonucleic acid). DNA from all organisms is made up of the same chemical and physical components. The DNA sequence is the particular side-by-side arrangement of bases along the DNA strand (e.g., ATTCCGGA). This order spells out the exact instructions required to create a particular organism with its own unique traits. The genome is an organism's complete set of DNA [17]. Genomes vary widely in size: the smallest known genome for a free-living organism (a bacterium) contains about 600,000 DNA base pairs, while human and mouse genomes have some 3 billion. DNA in a genome is arranged into distinct chromosomes--physically separate molecules that range in length from about 50 million to 250 million base pairs [17].

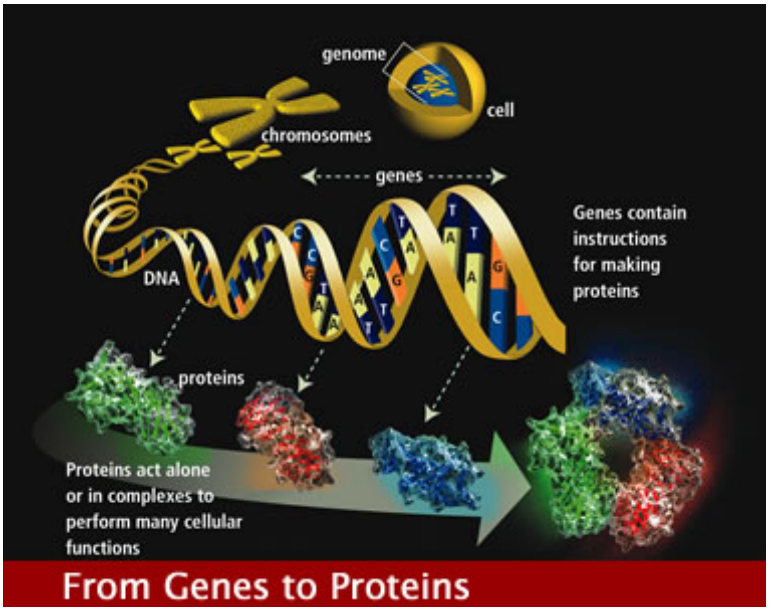

Figure 1: From Genes To Proteins[17]

Each chromosome contains many genes, the basic physical and functional units of heredity. Genes are specific sequences of bases that encode instructions on how to make proteins[17]. Proteins perform most life functions and comprise the majority of cellular structures. Proteins are large, complex molecules made up of smaller subunits called amino acids. Chemical properties that distinguish the 20 different amino acids cause the protein chains to fold up into specific three-dimensional structures that define their particular functions in the cell[17]. Pattern of genes expressed in a cell are characteristic of its current state. Messenger RNAs (mRNAs) are intermediates between DNA sequences and the production of proteins. Virtually all differences in cell state and type are correlated with changes in mRNA levels of many genes.

In the recent years, tremendous progress has been made in identification of genes in genomes. Scientists now are interested in finding how genes interplay and function to produce proteins. They want to study the complex interplay of all the genes simultaneously. This requires high throughput and large-scale technologies. DNA microarrays provide such a high throughput method for exploring the genome at the molecular level. They give the scientists the opportunity to analyze vast amounts of genetic information in a single experiment: the expression of thousands of genes at one time[19]. First described in 1995 [35], high-density DNA microarray methods have already made a marked impact on many fields, including cellular physiology [36-41], cancer biology [27-32], and pharmacology [25,42], [20].

In a microarray experiment, DNA complementary to genes of interest is generated and laid out in microscopic quantities on solid surfaces at defined positions. DNA reverse transcribed from the mRNA samples is washed over the surface to which the complementary DNA binds. Relative expression levels of genes can be studied by use of simultaneous, two-color hybridization. Fluorescent probes are prepared from two mRNA sources to be compared. Cyanine3 (green) and Cyanine5 (red) are the typical probes used. Probes are mixed and washed over the microarray. Each probe is excited using a laser, and its fluroscence at each element detected with a scanning confocal microscope. Relative intensity of the Cy5/Cy3 probes is a reliable measure of the relative abundance of specific mRNA's in each sample. Typically, genes in normal (control) conditions are

compared with genes under stress (treated). If genes under stress (treated) produce more mRNA than genes in normal conditions (control), the ratio of treated/ control is greater than 1, then the gene is said to be positively expressed, otherwise negatively expressed. If ratio of treated /control is the same, then the genes are said to be zero-expressed. Typically, logarithmic values are used. So, a positive log (treated/control) value indicates positive expression, negative values indicate negative expression and zero vales indicate zero-expression.

DNA microarrays generate huge amounts of data of the expression profiles of thousands of genes. One of the microarray data sets, which we have analyzed as part of this project is the gene expression data from three different varieties of Arabidopsis thaliana; Cape Verde (cv), Columbia (col), Wassilewskija (ws) and Thellungiella (th), a relative of Arabidopsis thaliana. The goal is to understand the mechanisms underlying the acclimation to ozone stress, by studying the gene expression patterns. The data was provided by the Bohnert's Lab at the University of Illinois.

Fusion was successful in helping biologists better understand the mechanisms underlying the acclimatization of certain varieties of Arabidopsis to ozone exposure. Figure 2 shows a summary of Fusion results, linking genes of different ecotypes of Arabidopsis to functional categories. Some of the research questions involved were like "What are the categories of genes whose up-expression is associated with acclimation to Ozone stress?" Fusion helps in making broad generalizations like "Up expression of trafficking genes were associated with acclimation to Ozone stress" as shown in Figure 2. It also helps the biologist make specific observations as shown in Figure 3. Figure 2 and figure 3 are discussed in more detail in Chapter 6.

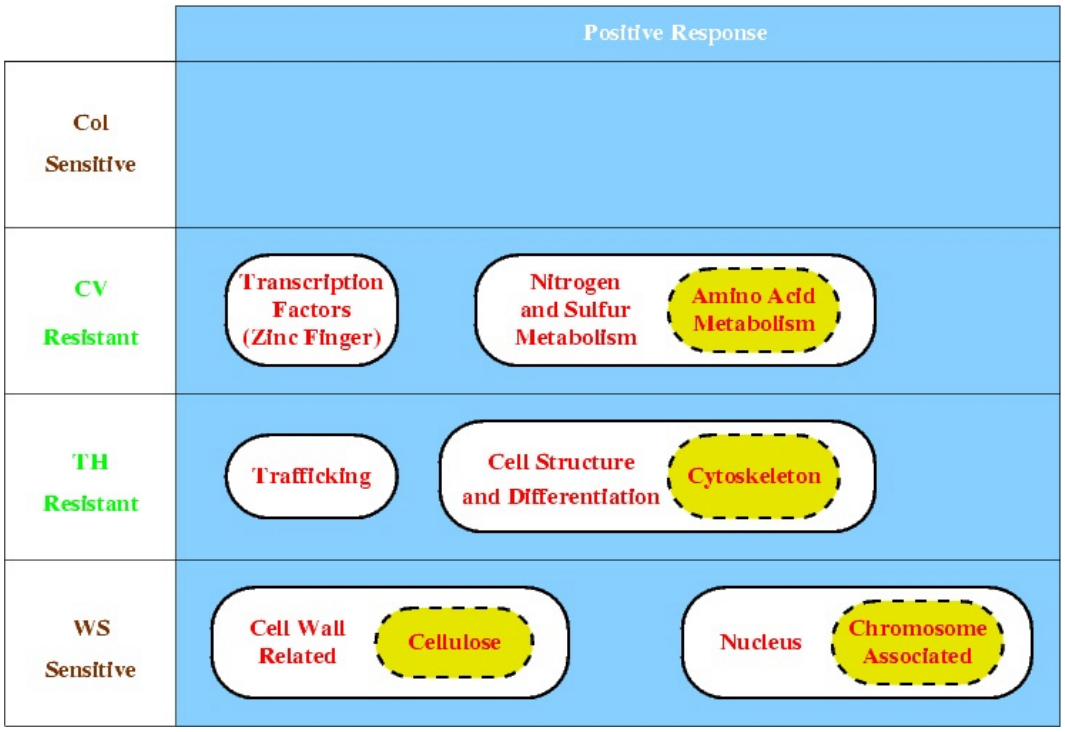

Figure 2: Summary diagram of the results found by the analysis of ozone-stress microarray data, using Fusion. Diagram courtesy: Dr.Ruth Grene. Data provided by Dr.Bohnert and Pinghua Li of the University of Illinois.

Gene	Annotation	Comments
At1g72050	C2H2-type zinc finger Homologous to TF III A	DNA repair (animal protein)
At2g18380	GATA zinc finger protein	Nitrogen regulatory protein (fungi)
At2g45050	GATA zinc finger protein	Four cysteines coordinate a zinc ion.
At3g262250	CHP-rich zinc finger protein	DC1 domain rich in cysteines and histidines

Figure 3: Candidate Zinc Finger proteins associated with acclimation to Ozone (Upregulated in Cape Verde) (Courtesy: Dr.Ruth Grene)

Fusion was built upon Snap, a web-based multiple view visualization system. The following section describes the multiple view visualization capabilities inherited from Snap.

1.1 Visual Data Exploration using Snap

Using Snap, the biologist can choose different views for different types of data. For example illustrated in the Figure 4, the biologist observes the behavior of the genes that were positively expressed in the ecotype *cv*.

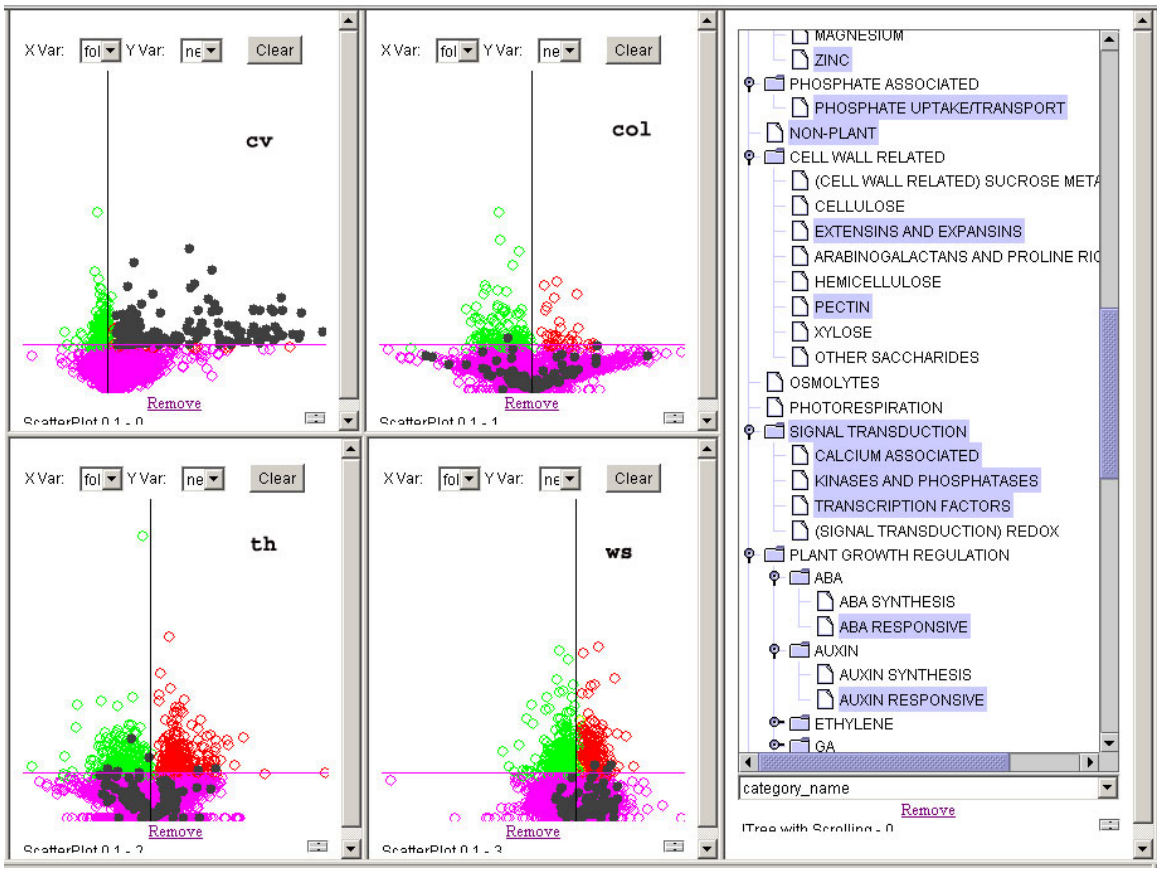

Figure 4: Visual data exploration using Snap

The biologist uses the scatter plot views to visualize the gene expression data and a hierarchical tree view for functional category data. The views are linked together so that a selection in one view highlights the corresponding genes in the other view. She finds

that positively expressed genes in *th* do not exhibit the same behavior in other experiments. She also finds the categories to which those genes belong (highlighted in the hierarchical tree).

The advantages of visual data exploration:

- Interactivity
    - Filter and narrow down on subsets-of-interest.
- Visual presentation of the results, thereby using the human visual pattern recognition abilities to find patterns.
    - Helps easily identify outliers and anomalous data
- Usability

Disadvantages:

- Can miss complex patterns in data
- Non-Exhaustive search

1.2 Data Mining using Proteus ILP

Proteus is an Inductive Logic Programming System, developed by Dr. Naren Ramakrishnan and Deept Kumar, at Virgina Tech.

Proteus takes input data expressed as gene expression levels in particular experiments, functional categories, and experimental conditions. As output it provides rules of the form,

Level(X, MildCycle1,“ + ”) :- cat(X,trafficking)

Where MildCycle1 represents a mild stress condition. This rule states: “If a gene (represented in the rule as A) belongs to ‘trafficking’ category, then it is positively expressed in MildCycle1 [48].

Typical results of Proteus are shown below.

Head	Body	Pos	Neg	Confidence	Support
level(X,ws,-)	cat(X,CYTOSKELETON)	5	1	83.333336	6
level(X,ws,-)	cat(X,CYTOSKELETON), cat(X,ACTIN)	3	0	100.0	3
level(X,ws,+)	cat(X,CELL MEMBRANES)	21	6	77.77778	27
level(X,ws,+)	cat(X,TRANSPORT PROTEINS)	17	5	77.27273	22
level(X,ws,-)	cat(X,NITROGEN AND SULFUR METABOLISM)	3	2	60.0	5
level(X,ws,-)	cat(X,METALS), cat(X,ZINC)	4	2	66.666664	6
level(X,ws,-)	cat(X,PROTEASES)	5	3	62.5	8

Figure 5: Typical results of Proteus

Advantages:

- Exhaustive search
- Complex pattern finding abilities

Disadvantages:

- Non-interactivity
- Difficult to use for non-computer scientists.
- Textual results only

1.3 Fusion

Fusion combines Snap and Proteus to gain the advantages of both visualization and data mining. Fusion allows the user to explore the data interactively, choose biases, run the data mining algorithms and visualize the discovered rules as illustrated in Figure 6. Fusion has the ability to smoothly switch across interactive data exploration and batch data mining modes. This optimizes the knowledge discovery process by facilitating a synergy between the interactivity and usability of visualization process (Snap) with the exhaustive pattern-finding abilities of an ILP rule mining algorithms (Proteus). One of the important features in Fusion is the smooth switch between the interactive and batch modes, rather than being a pipeline of one mode followed by the other.

Figure 6: Fusion supports both visual data exploration and data mining

Fusion is based on the analogy between visualization and data mining concepts (as shown in Table 1).

Table 1: Correspondence between Data mining and Visualization concepts

Data mining concept	Visualization concept
Descriptor	Selection / highlight
Bias	Selection operator/brush
Rule	Brushed selections
Confidence	Relative Brush strength
Support	Brush strength
Niche	Dual brush selection
Antecedent	Brushed Selection
Consequent	Linked Highlight

The contributions of this research are in three areas: Fusion Theory, Fusion Software Architecture and Fusion User Interface.

Fusion Theory

The underlying theory of the Fusion system is based on the analogy between visualization and data mining concepts. The data mining concepts like descriptor, bias, rule and evaluation measure were associated with corresponding visualization concepts.

Fusion Software Architecture

Fusion incorporates a software architecture that supports the integration of interactive data exploration and batch ILP rule mining with possible extensibility over descriptors (bias selection operators), views (visualizations), data mining algorithms and evaluation criteria.

Fusion User Interface

Fusion user interface supports a two-mode exploratory process; the interactive mode where the user gets a feel of the data and chooses the biases; the batch mode in which the data mining algorithm is run, followed by the visualization of discovered rules and associated confidence measures in the data mining component. Fusion also helps the user validate the discovered rules, visually by showing the user their constituent data and attributes.

Chapter 2

2 Related Work

DNA micro arrays generate huge amounts of data of the expression profiles of thousands of genes. A variety of visualization and data mining techniques have been developed in the recent years to help the biologists find interesting relationships in microarray data.

2.1 Visualization

Cho, *et al.* [43] use direct visual inspection to group together genes with similar expression patterns, to cluster genes whose expression correlated with particular phases of the cell cycle. The method is best suited for instances in which the patterns of interest are clear in advance (such as a periodic fluctuation in phase with the cell cycle), but it does not scale well to larger data sets and is less appropriate for discovering unexpected patterns [22].

Carr, *et al.* [45] describe the design of graphical displays for showing the results of clustering in gene expression data. The displays include stereo plots, parallel coordinate (time series) plots and conditioned parallel coordinate plots.

Inputs to Fusion

- Usefulness of multiple views for different parts of the data set.

GeneBox developed by Shah, *et al* [44] is a general-purpose 3-D visualization tool for multi-variable micro array gene expression data. GeneBox is designed to help scientists answer complex queries through interactive visual exploration of micro array data sets. Its strengths are its multidimensional visual interface, coupled with interactive

functionalities and a variety of user controls to customize and enhance the output. A screen shot of GeneBox is shown in Figure 7. One weakness is that it is limited by the human vision and visual pattern finding abilities.

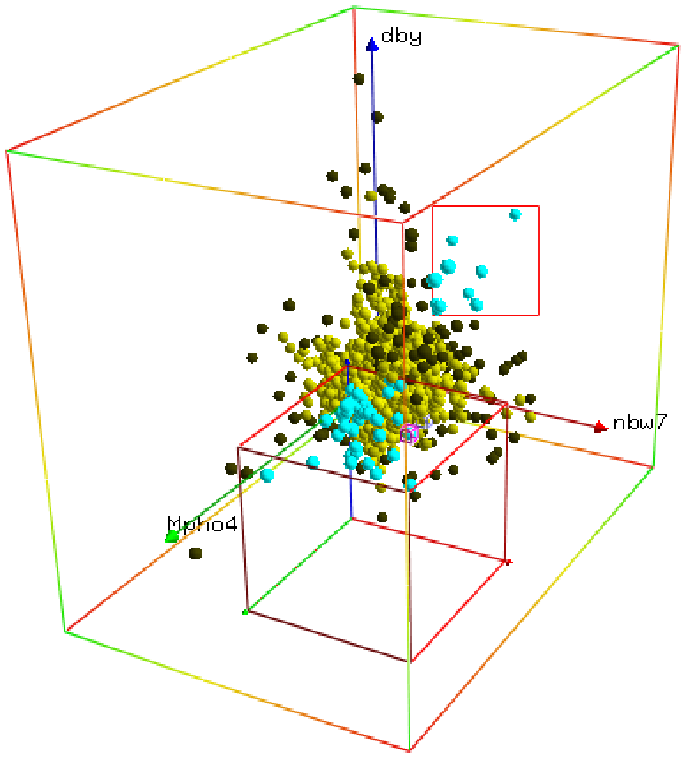

Figure 7: GeneBox screenshot. Red, blue, and green axes represent different experimental conditions. Genes selected by selection plane (red rectangle) and selection box (red cube) are highlighted with a different color.

Inputs to Fusion

- Richness of selection operators
- Use of different colors for different levels of exploration

Hochheiser and Shneiderman [7] describe the use of TimeSearcher for exploring a time series data set involving gene expression profiles. Timeboxes are the primary query tool in the TimeSearcher application, which supports interactive exploration via dynamic queries, along with overviews of query results and drag-and-drop support for query-by-

example. Timeboxes are rectangular, direct-manipulation queries for studying time-series datasets specially to find time-series patterns (Figure 8). This tool is potentially useful to study gene expression data across time series experiments.

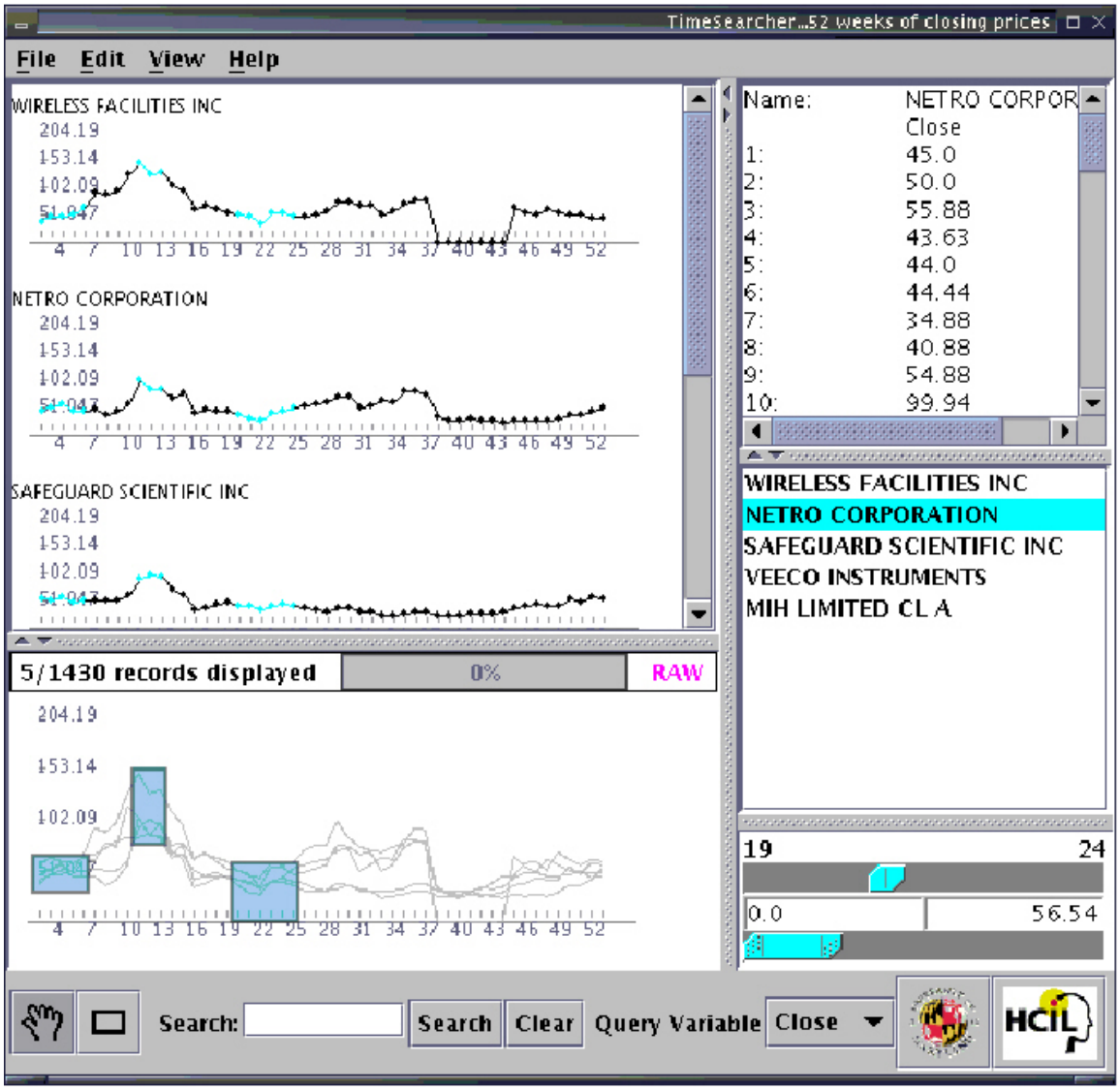

Figure 8: TimeSearcher application window. Clockwise from upper-left: data items, details-on-demand, item list, range sliders for query adjustment, and query space.

Its strengths are that it uses a multiple view visualization technique with support for dynamic querying. However, it lacks a batch mode of pattern finding and is limited by the human vision pattern finding abilities.

2.2 Data Mining

Most global gene expression analyses have used some form of unsupervised clustering algorithms to find genes coregulated across a data set [20] [22,23,24]. Eisen *et al.*[16] use a hierarchical clustering algorithm for mining clustering gene expression data from budding yeast *Saccharomyces cerevisiae*. The algorithm groups together genes of known similar function. They also state that coexpression of genes of known function with poorly characterized or novel genes may provide a simple means of gaining leads to the functions of many genes for which information is not available currently [16]. Tamayo, *et al.* [22] use self-organizing maps (SOMs) to organize the genes into biologically relevant clusters that suggest novel hypotheses about hematopoietic differentiation. Algorithms like hierarchical clustering, k-means clustering, SOM come under unsupervised algorithms that use a exploratory data mining strategy, where no prior knowledge is used.

Supervised algorithms are based on the idea of learning by example. They create a model by running the algorithm on the training data and identify a classification for the incoming new data. When a coregulated class of genes is known, supervised clustering algorithms are trained to recognize known members of the class, can assign uncharacterized genes to that class [20]. For example, a machine-learning method known as a support vector machine has been used to classify yeast genes by function on the basis of shared regulation [33]. Robust determination of coregulated gene clusters may be achieved by using a tiered approach: unsupervised clustering to identify coregulated genes followed by testing and refinement with supervised algorithms [34].

Inductive Logic Programming (ILP) mines information in a single step by hybrid reasoning – integration of unsupervised and supervised algorithmic reasoning. Heath, *et al* [14] use an inductive logic programming technique to find out relationships in the gene expression data. Their algorithm takes input data expressed as gene expression levels in particular experiments, functional categories, and experimental conditions. As output, it provides rules of the form:

Level (A, MildCycle1, +) :- cat (A, Flavonoids)

This rule states, "If a clone (represented in the rule as A) belongs to Flavonoids, then it is negatively expressed in MildCycle1 [14].

Inputs to Fusion

- A batch mode of finding relationships in the data

2.3 Visualization and Data Mining

Seo and Shneiderman [1] use hierarchical clustering to find patterns in multi-dimensional datasets, especially genomic microarray data. They developed four techniques that can be used for interactive explorations of clustering results. The first technique is used to see an overview of the entire dataset, coupled with a detail view so that high-level patterns and hot spots can be easily found and examined. The second technique involves using dynamic query controls to enable users to restrict the number of clusters they view in order to gain a better understanding of those clusters (as shown in Figure 9). The third technique use coordinated displays as an overview tool that has a bi-directional link to 2-dimensional scattergrams.

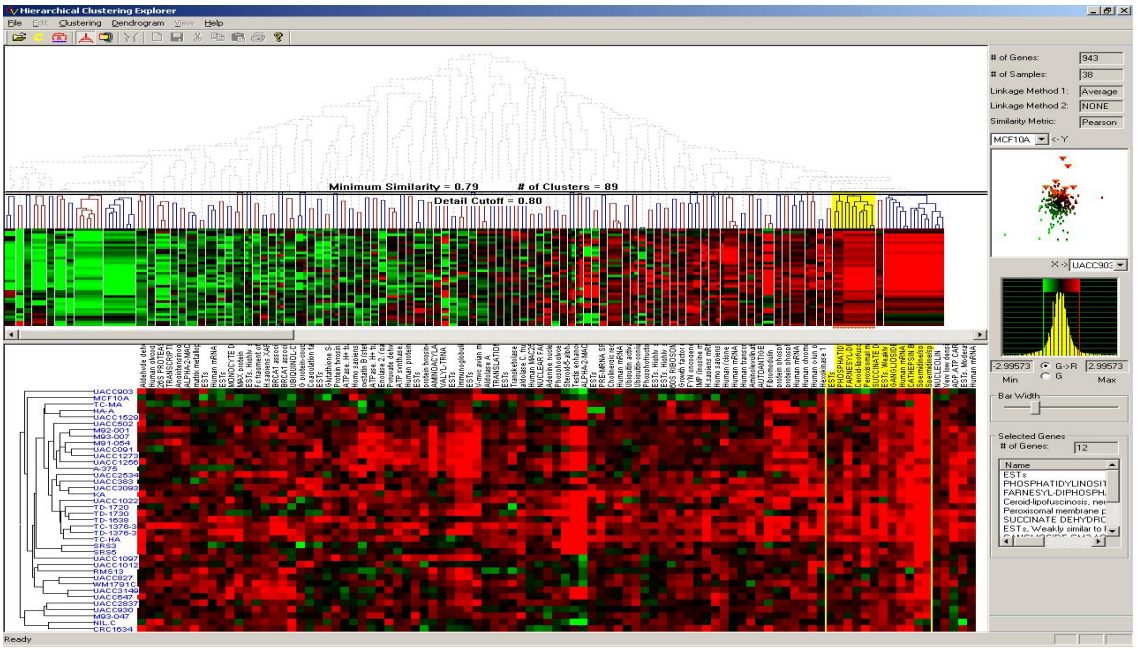

Figure 9: Dynamic Query Controls. Users can adjust the level of detail by dragging up with the Detail Cutoff Bar.

Another technique uses cluster comparisons to allow researchers to see how different clustering algorithms group the genes. This enables the user to compare the results on one screen as shown in Figure 10.

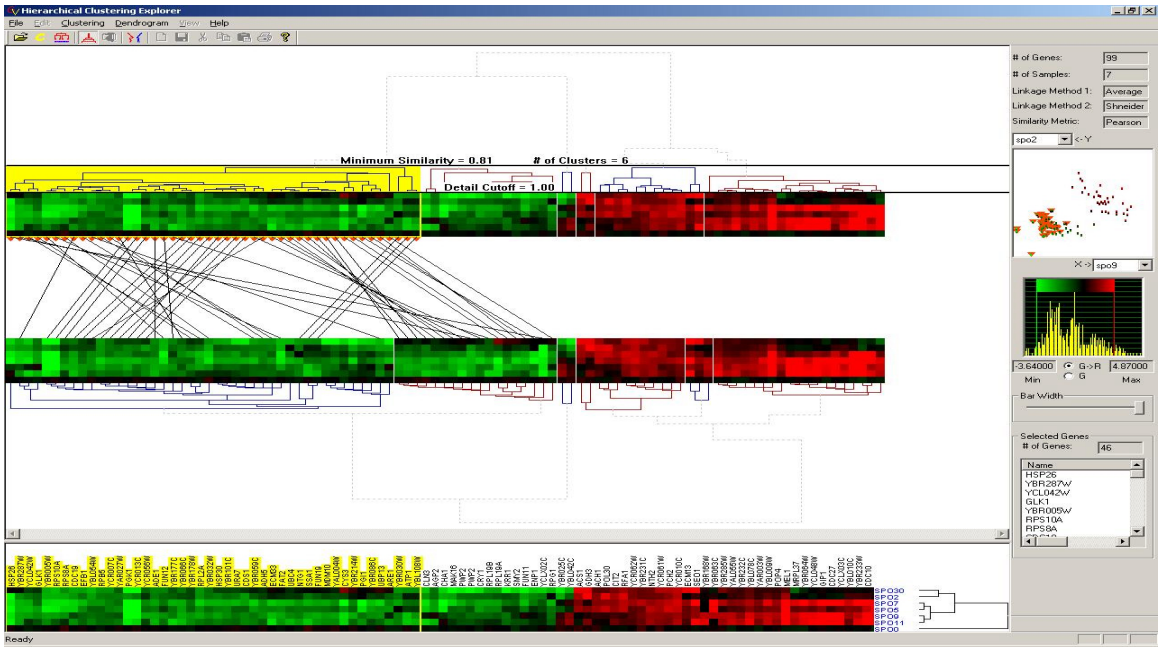

Figure 10: Cluster Comparisons. Users can see the mapping of each gene between the two different clustering results by double-clicking a specific cluster. The selected cluster will highlight and lines from each item in that cluster will be drawn to their position in the second clustering result.

Inputs to Fusion

- Batch mode of finding relationships
- Interactive exploration of the relationships found

Dysvik and Jonassen [5] describe a Java application, named J-Express, which allows flexible analysis of microarray gene data through the use of multi-dimensional scaling, clustering, and visualization in an integrated manner. Implementations of hierarchical clustering, k-means, principal component analysis and self-organizing maps are included in J-Express. Figure 11 shows a snapshot of the user interface.

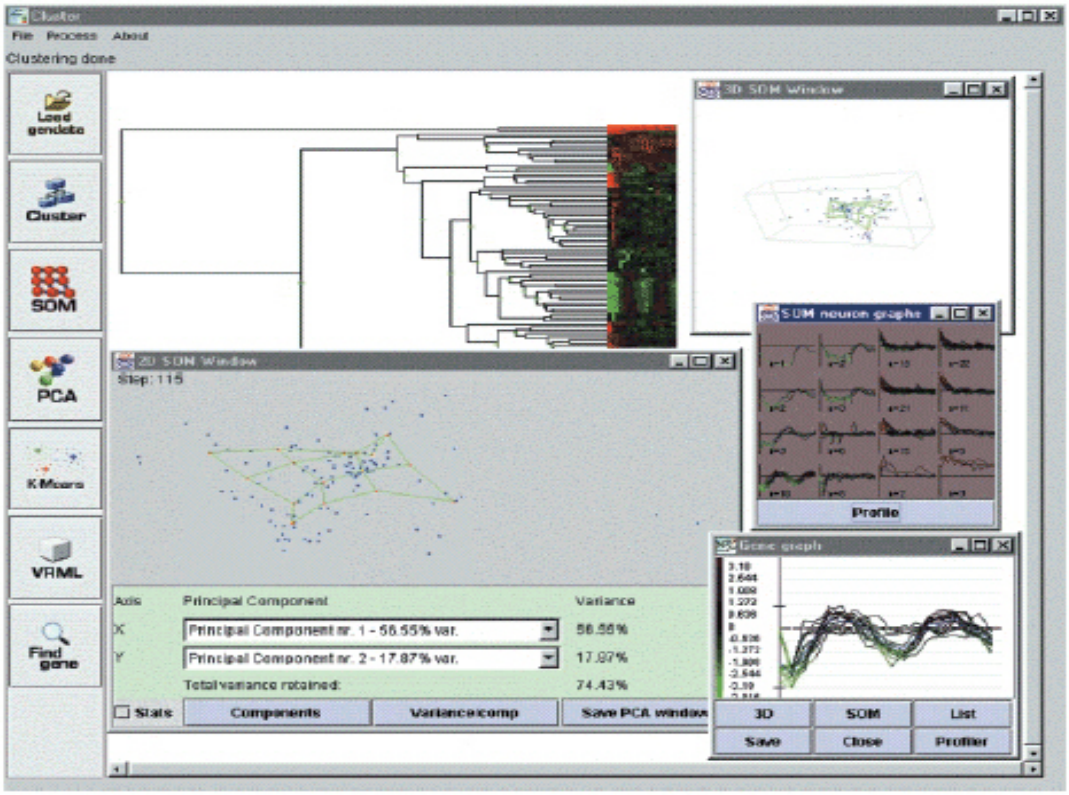

Figure 11: A screen shot of J-Express. The main window contains pull-down menus as well as shortcut buttons. From top left corner, clockwise direction around the screenshot are shown: a tree viewer shows the result of a hierarchical clustering; a 3D view of a SOM displayed using a PCA transformation; a summary result of the SOM showed as a panel of gene graph; a gene graph window; a 2D view of a SOM.

Inputs to Fusion

- Batch mode of finding relationships by data mining.
- Interactive exploration of the data mining results.

Han and Cercone [6] describe an interactive model, named RuleViz, for visualizing the process of knowledge discovery and data mining (Figure 12). The model consists of the five components according to the main ingredients of the knowledge discovery process: original data visualization, visual data reduction, visual data reprocess, visual rule discovery and rule visualization.

The interactive system, CViz, exploits parallel coordinates to visualize the technique of rule induction. The original data is visualized on the parallel coordinates and can be interactively reduced (Figure 13).

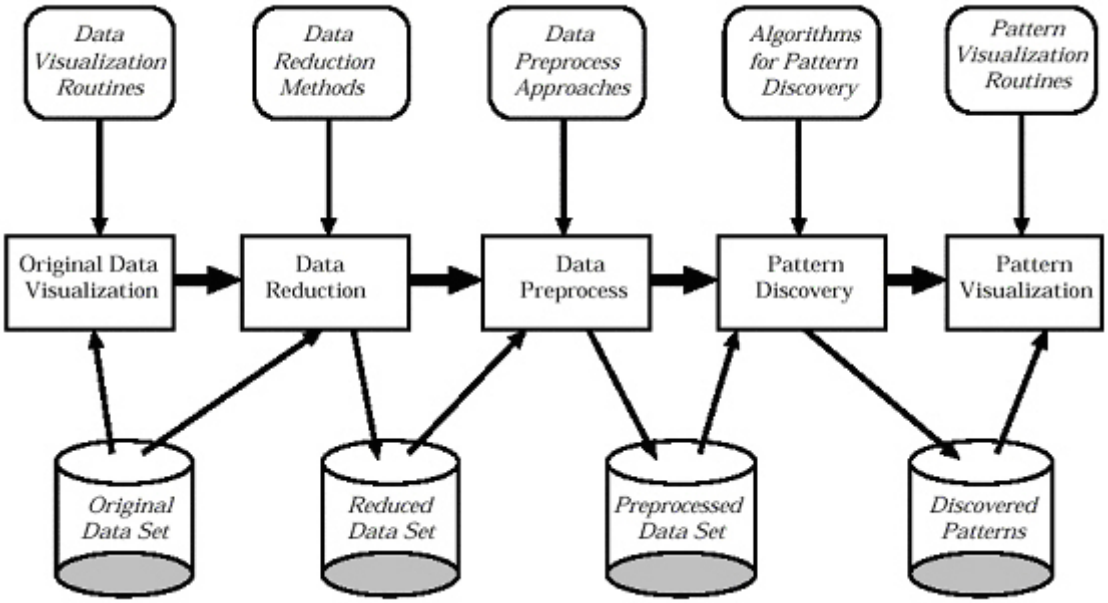

Figure 12: The RuleVis model

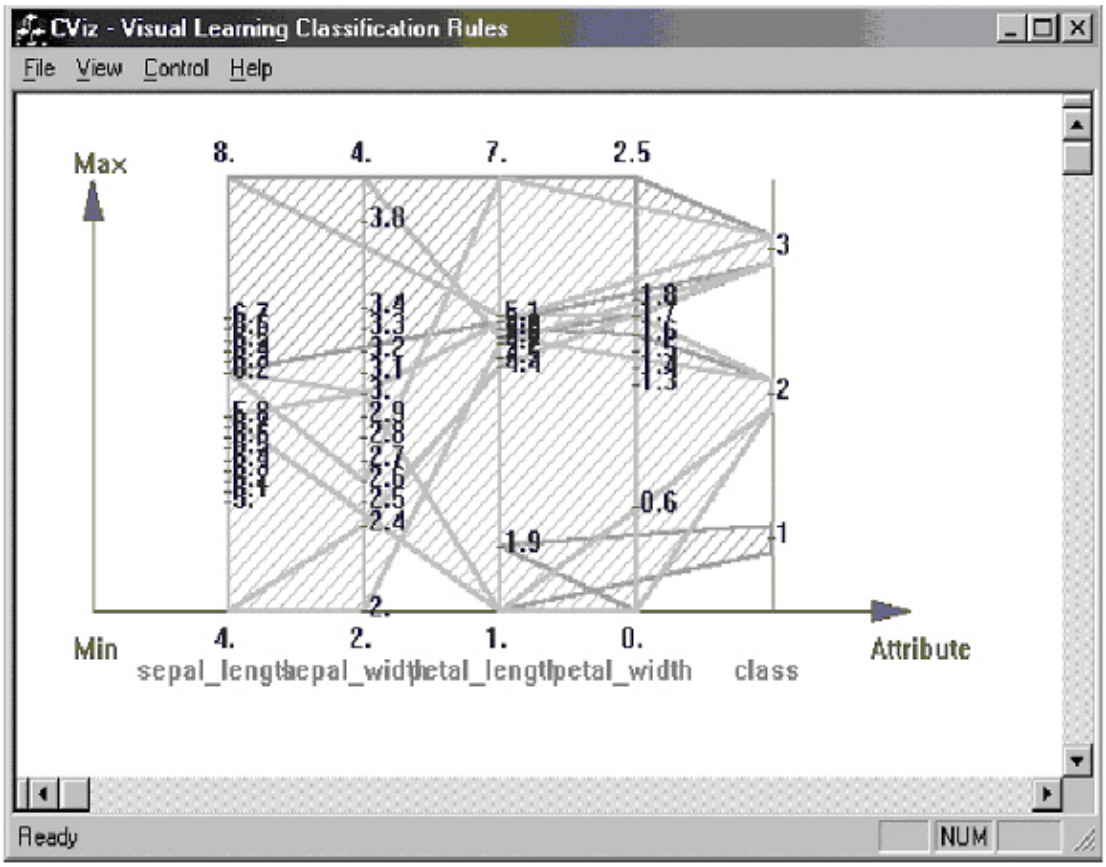

Figure 13: The rules of the IRIS data.

Inputs to Fusion

- Interactive reduction of information space
- Batch Mode of execution of pattern finding
- Visualization of found patterns

Keim [3] proposes a classification scheme of information visualization and visual data mining techniques. The classification scheme is based on the data type to be visualized, the visualization technique and the interaction and distortion technique. The visualizations of the data allow the user to gain insight into the data and come up with new hypotheses. The verification of the hypotheses can also be done via visual data exploration but it may also be accomplished by automatic techniques from statistics or machine learning [3].

Spotfire® [47] presents clustering results in multiple views, placing each cluster in a separate parallel coordinate view. The visualizations are linked for brushing. Selecting data items in any view shows feedback in a common detail window. The fundamental interaction technique in Spotfire® is the dynamic query sliders, which interactively filter data in all views [46].

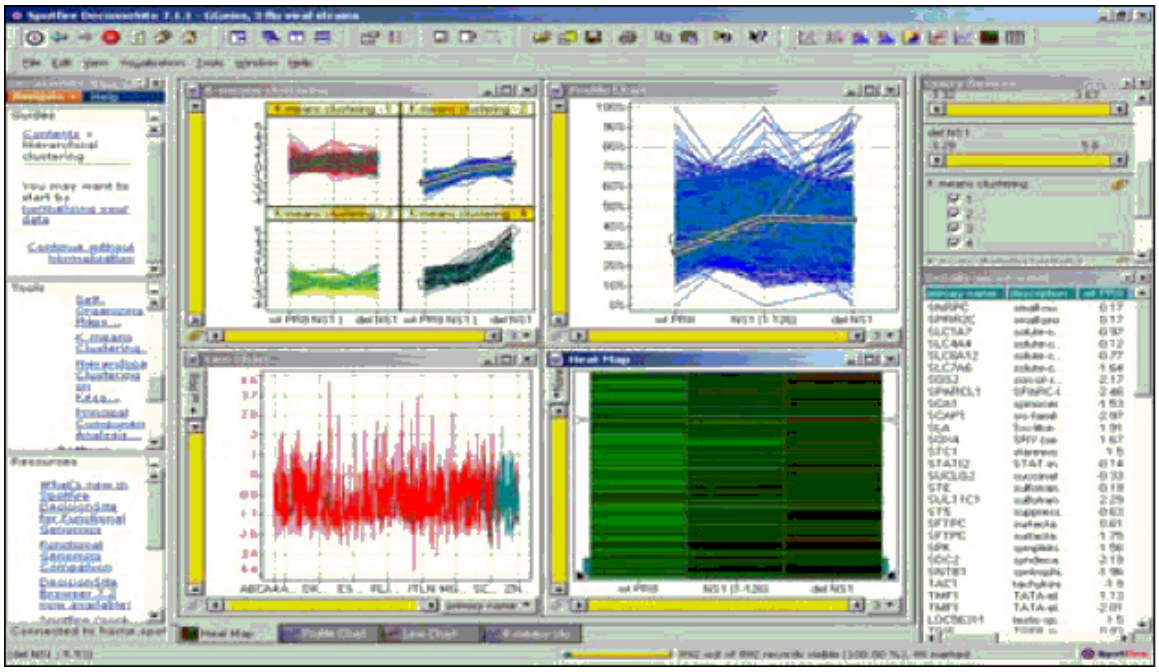

Figure 14: SpotFire®

Each of the existing tools discussed so far in this section, combine visualization and data mining in the following ways:

1) Interactive (visual) specification of parameters to the data mining algorithms
2) Interactive reduction of the input data sets (inputs to the data mining algorithm)
3) Visualization of the results found

Fusion's uniqueness lies in its support for the analogy between visualization and data mining concepts and its smooth switch between the interactive and batch modes of operation.

Using the inputs from various existing and data mining techniques, Fusion has been designed to help the biologists interactively explore and mine microarray data. The following are the features of Fusion:

- Customized visualization components for microarray data analysis
- Multiple-view visualization technique
- Interactive reduction of information space
- Interactive mode of exploration by brushing-and-linking across multiple views
- Batch mode of finding Inductive Logic Programming rules
- Visualization of the rules found in the batch mode of operation
- Smooth switch between the interactive and the batch modes
- Support for the analogy between visualization and data mining concepts

Chapter 3

3 Theory

Users can extract knowledge from databases depending on the type of the tasks. The type of the tasks can range from looking for specific facts to making broad generalizations about the data. For example, a microarray gene-expression database can be used by a biologist to find the expression of a particular gene. The same database can be used to extract knowledge of a more general nature, like information on whether there has been increase in the expression of genes belonging to a particular category.

So, the user tasks can be classified broadly into two categories

1) Specific tasks
2) General tasks

In the following paragraphs, we will discuss with examples, how visualization helps the user in accomplishing these tasks.

3.1 Knowledge extraction using brushing and linking across visualizations

The different types of knowledge that can be extracted using brush-and-link technique can be discussed with the example of the microarray database (discussed in Chapter 1).
This process is often supported in interactive visualization using multiple views with brushing and linking. The data is displayed in two (or more) different visualizations, each of them showing different tables in the underlying database. The visualizations are linked such that interactively selecting (brushing) entities in visualization, highlights the corresponding entities in the other visualizations. Figure 15 and Figure 16 show examples of brushing-and-linking in Snap.

Extreme Specialization

In Figure 15, the table on the left is a listing of the names of the genes. The right table is a listing of the functional categories. The biologist wants to find out the category of a specific gene. She selects the gene she is interested in. The corresponding categories to which the gene belongs are highlighted in the other view. This visualization setup does help the user in specific fact-finding, but has limited use as it does not give any information about trends or patterns in the data.

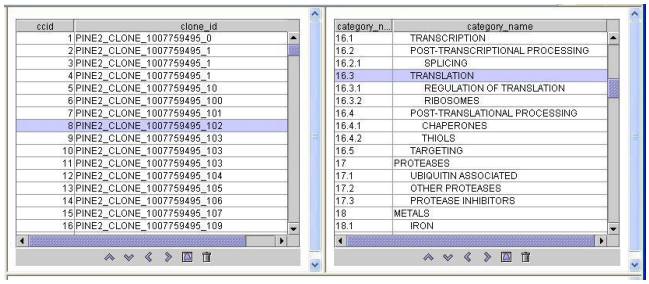

Figure 15 : Too Specific

Extreme Generalization

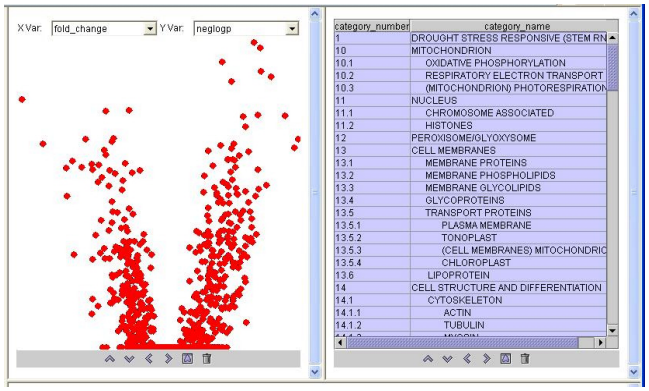

Figure 16 : Too General

In Figure 16, the scatter plot shows the expression profiles of genes in a particular experiment. The tree view is a hierarchical listing of the functional categories. If the

biologist selects the whole tree, all the genes belonging to those categories are seen highlighted in the scatter plot. This means that 'all of the genes belong to one of the categories'. Though this helps the biologist in verifying that every gene has been categorized, no more useful knowledge about the relationship between genes and their categories can be derived from such visualization.

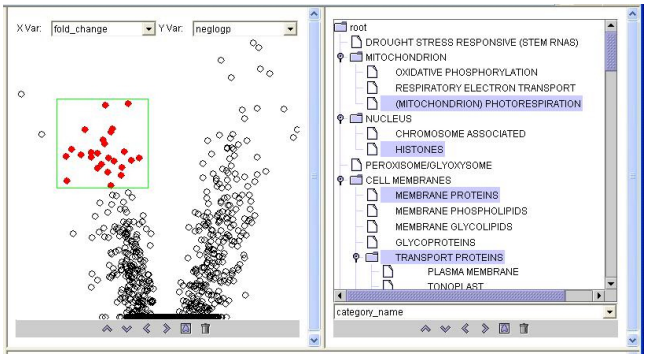

Figure 17: Middle Ground

In Figure 17, the biologist chooses a region on the upper left of the scatter plot and finds the categories to which those genes belong. This enables her to make a sort-of-general statement that "Genes that are significantly highly negatively expressed in Experiment1 belong to categories PHOTO RESPIRATION, HISTONES, MEMBRANE PROTEINS and TRANSPORT PROTEINS".

So a balance between generalization and specialization is very crucial for deriving interesting relationships from data. In a visualization setting, users can achieve this balance by appropriate choice of views and selection operations.

The advantages of knowledge extraction by visual data exploration:

- Interactivity
    - Filter and narrow down on subsets-of-interest.
- Visual presentation of the results, thereby using the human visual pattern recognition abilities to find patterns.
    - Helps easily identify outliers and anomalous data
- Usability

Disadvantages:

- Can miss complex patterns in data
- Non-Exhaustive search

3.2 Knowledge extraction using Inductive Logic Programming (ILP)

Induction is typically implemented as a search through the space of possible hypotheses. Such searches usually employ some special characteristic or aspect to arrive at a good generalization [15]. Figure 18 shows example of ILP rules mined by Proteus, an ILP system.

ILP is a technique that provides a way

a) to correlate output variables (Gene expression) with input variables (functional categories)
b) allows the incorporation of a priori domain knowledge [14]

ILP takes input data expressed as gene expression levels in particular experiments, functional categories, and experimental conditions. As output it provides rules of the form,

Level(X, MildCycle1,“ + ”) :- cat(X,trafficking)

Where MildCycle1 represents a mild stress condition. This rule states: “If a gene (represented in the rule as A) belongs to ‘trafficking’ category, then it is positively expressed in MildCycle1 [48]. Proteus is an ILP system, developed by Dr. Ramakrishnan and Deept Kumar, which Fusion uses to generate such rules.

Typical results of Proteus are shown below in Figure 18 .

Head	Body	Pos	Neg	Confidence	Support
level(X,ws,-)	cat(X,CYTOSKELETON)	5	1	83.333336	6
level(X,ws,-)	cat(X,CYTOSKELETON), cat(X,ACTIN)	3	0	100.0	3
level(X,ws,+)	cat(X,CELL MEMBRANES)	21	6	77.77778	27
level(X,ws,+)	cat(X,TRANSPORT PROTEINS)	17	5	77.27273	22
level(X,ws,-)	cat(X,NITROGEN AND SULFUR METABOLISM)	3	2	60.0	5
level(X,ws,-)	cat(X,METALS), cat(X,ZINC)	4	2	66.666664	6
level(X,ws,-)	cat(X,PROTEASES)	5	3	62.5	8

Figure 18: Typical results of Proteus

Advantages:

- Exhaustive search
- Complex pattern finding abilities

Disadvantages:

- Non-interactivity
- Difficult to use for non-computer scientists.
- Textual results only

3.3 Fusion: Visualization and ILP Rule Mining

The goal of Fusion is to use the advantages of both visualization and data mining to optimize the knowledge discovery process.

Fusion has the following advantages:

- Interactivity
- Usability

} Visualization Strengths

- Exhaustive search
- Complex pattern finding abilities

} Data Mining Strengths

Fusion accomplishes this by supporting the analogies between visualization and data mining concepts. The analogies between Data Mining and Visualization are listed below in Figure 19.

Data mining concept	Visualization concept
Descriptor	Selection / highlight
Bias	Selection operator/brush
Rule	Brushed selections
Confidence	Relative Brush strength
Support	Brush strength
Niche	Dual brush selection
Antecedent	Brushed Selection
Consequent	Linked Highlight

Figure 19: Correspondence between Data mining and Visualization concepts

Each of the analogies is discussed below in detail.

Descriptor:

Descriptor is the symbol/language/schema used for describing the entity under consideration.

Foe example to describe the category information of a subset/group of genes belonging to the categories SPLICING and TRANSLATION, the user might use a schema like cat(X, SPLICING), cat (TRANSLATION) meaning, each gene in that class belongs to both SPLICING AND TRANSLATION. In this example, the descriptor is cat.

In a visual setting, selection and highlight accomplish the task of a descriptor. For example, the above-mentioned task of, displaying the category information of a subset of genes can be represented by a highlight of the corresponding category as shown in the following Figure 20. Other examples of selection operators are subtree selection operators, circular-region selection operators etc.

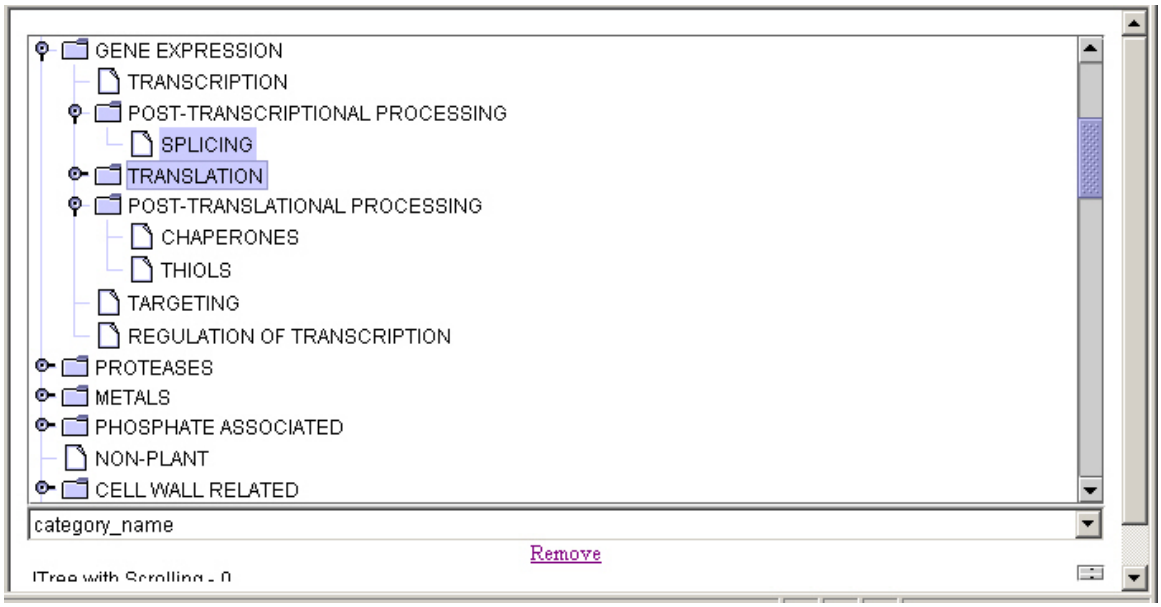

Figure 20: Highlight showing that the subset/group of genes under consideration belong to SPLICING and TRANSLATION.

Bias:

Bias is a specification of a pattern for subset selection. For example, a bias can be something like level(X, -2 < fold_change < -1, 0.45 < neglogp < 0.85), indicates that the subset/group of genes under consideration have fold_change between –1 and 1; and that the neglogp is between 0.45 and 0.85.

In a visual setting, selection operators help in bias specification. For example, using a rectangular selection operator can specify the above bias as shown in the Figure 21.

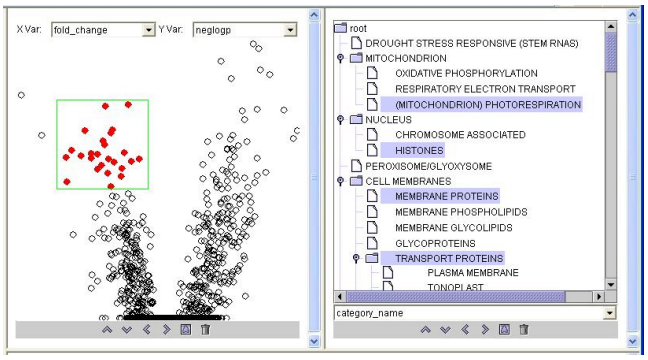

Figure 21: The users chooses subsets in terms of rectangular regions on the scatter plot (rectangular bias)

Antecedent, Consequent

In the above example, the user selects a subset of the negatively expressed genes (on the left side of the scatter plot). The user finds that they belong to categories PHOTO RESPIRATION, HISTONES, MEMBRANE PROTEINS and TRANSPORT PROTEINS.

So the user concludes that if genes are negatively expressed, then they belong to one of those highlighted categories as shown in Figure 21. The antecedent in this conclusion is the 'if' part of it, where the user brush the selection (region on the scatter plot); and the consequent, the 'then' part of it, where the user sees the linked highlight (highlighted categories in the hierarchical tree).

Rule, Confidence, Support:

Rule is a statement that generalizes the relationships across subsets. For example, a statement like level (X, th, -) :- cat(X, TRANSLATION), defines a relationship between subset of genes belonging to TRANSLATION and a subset of genes that were negatively expressed in the ecotype *th* as shown in Figure 22. The relationship says that if a gene belongs to TRANSLATION, then it is negatively expressed in *th*. Confidence is one of the evaluation criteria used to judge a rule. In the example, the confidence of the rule is calculated by the following ratio:

$$\frac{\text{(Number of genes that belong to TRANSLATION and are negatively expressed in ecotype th)}}{\text{Number of genes that belong to TRANSLATION}}$$

Support is the relative frequency or number of times a rule produced by a rule induction system occurs within the database. The higher the support, the better the chance of the rule capturing a statistically significant pattern.

In a visualization setting, a rule is represented as linked selection/highlight. For example, in the following Figure 22, the highlighted genes in the scatter plot and the linked highlight in the category tree represent the six genes that belong to TRANSLATION. The brush strength shows the support of the rule. In this example, the support is six represented visually by the six highlighted genes. The confidence is 5/6, represented visually as the relative brush strength of the black selections (genes satisfy the rule) to the total selections (the genes under consideration).

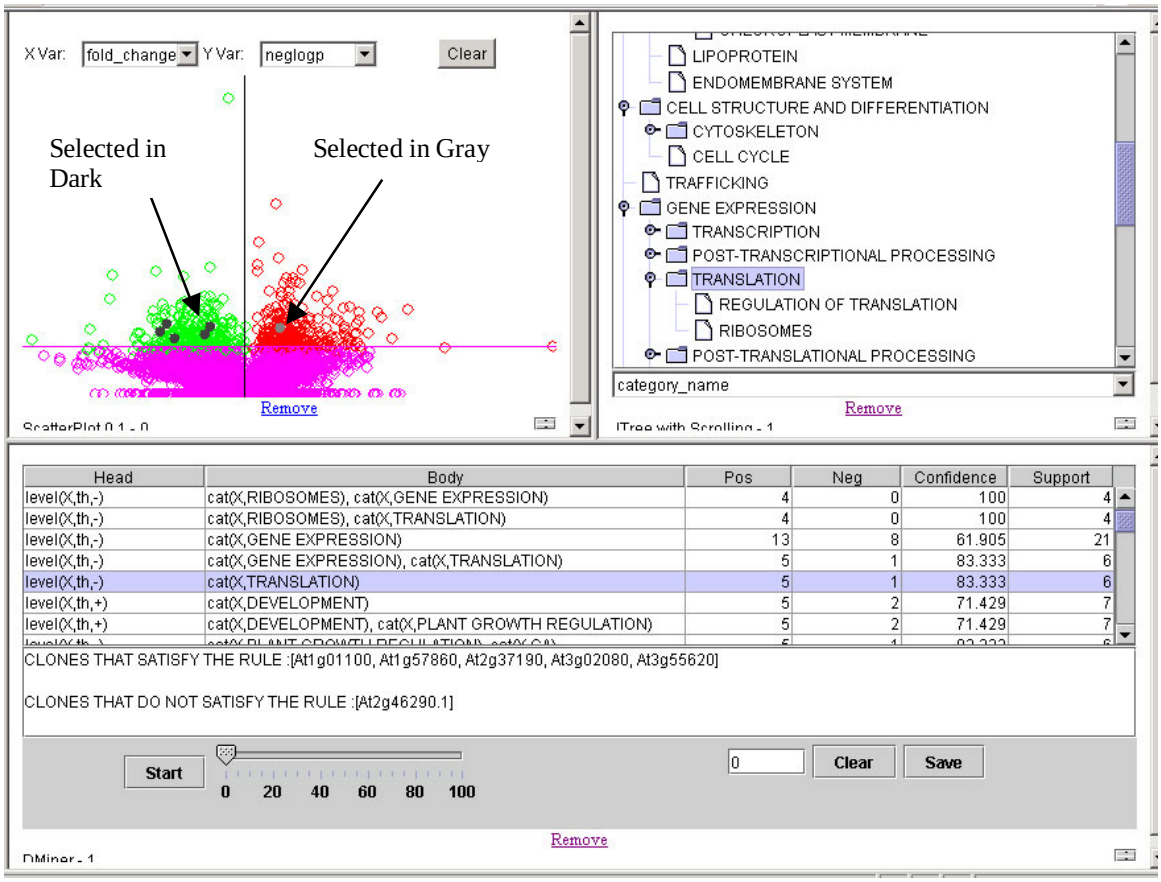

Figure 22: Visualization of rules and their evaluation measures

Niche:

Niche refers to a bidirectional rule with 100% confidence. For example, level (X, *cv*, +):-level(X,*th*, +) and level (X, *th*, +):-level(X,*cv*, +) each with 100% confidence, refers to those genes that are positive in both *cv* and *th*. The user can find this niche visually, as

illustrated in Figure 23. The user selects the positively expressed genes in variety *th* and observes the behavior of the corresponding genes in variety *cv*. The user is interested in narrowing down on genes that are positively expressed in *cv* as well. So the user sub-selects the positively expressed gene in *cv*, and finds the corresponding gene in the variety *th*. User thus, narrows down on the gene that is positively expressed in both *th* and *cv* (niche).

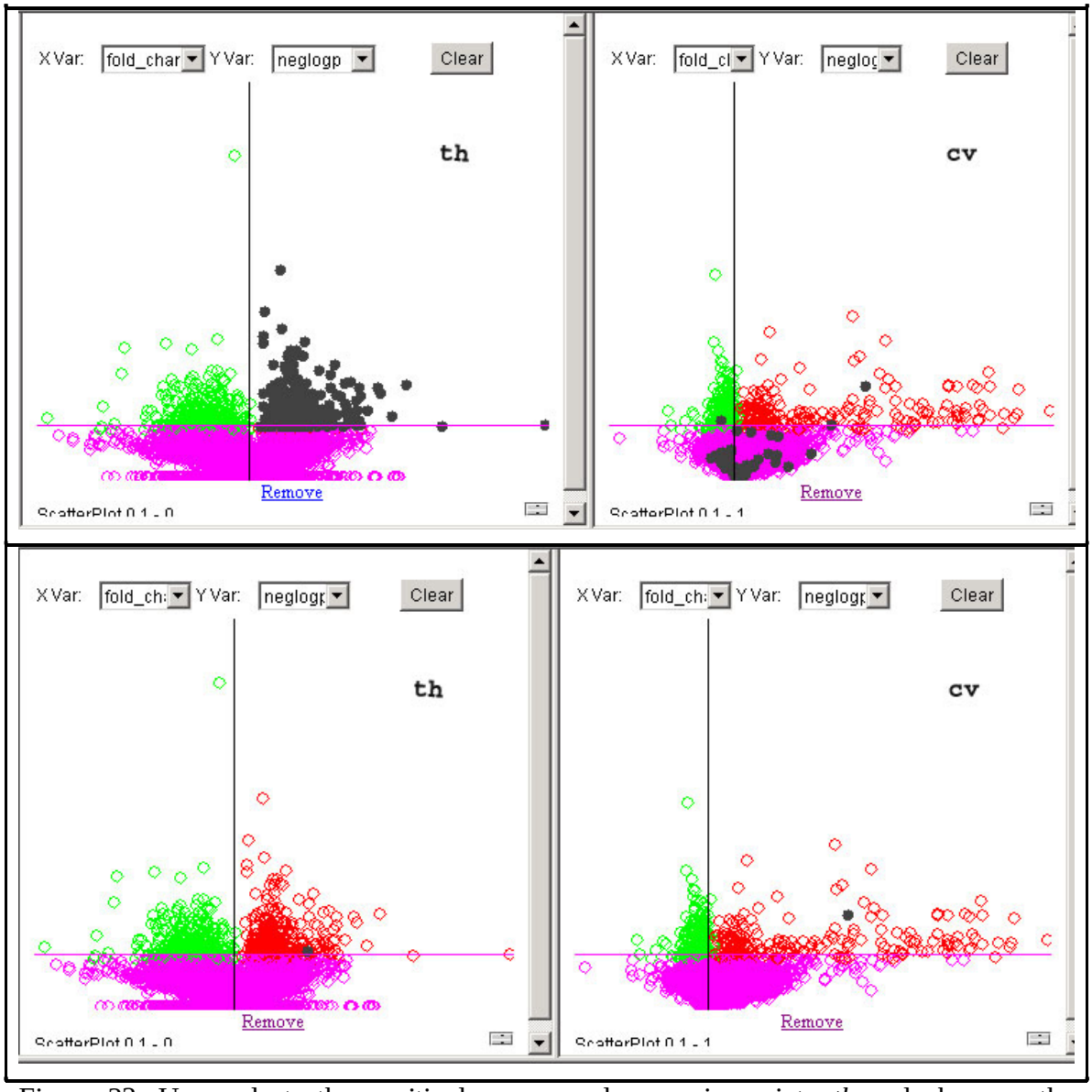

Figure 23: User selects the positively expressed genes in variety *th* and observes the behavior of the corresponding genes in variety *cv;* User sub-selects the positively

expressed gene in cv, and finds the corresponding gene in the variety *th*. User thus, narrows down on the gene that is positively expressed in both th and cv (niche).

3.4 Steps in the Fusion Knowledge Discovery Process

The following paragraphs discuss the different steps of knowledge discovery process using Fusion.

3.4.1 Choosing data and visualizations

The first step is to choose the data, attributes, and visual representations. Choice of the data would mean deciding which tables in the database are to be analyzed. Attributes are the columns of each of those tables. Visual representations are views used to visualize the data. Fusion enables exploration of any relational database. Its multiple view visualization framework gives the user an option to choose appropriate visualization components according to the underlying data. The choice of views depends on the data, the way the user wants to view the data, and the kind of subset selections the users want to make. In the example shown in Figure 24, the data chosen are the tables gene-expression, category, and a gene-to-category mapping table in the underlying microarray database. The attributes (columns) chosen to be visualized in the scatter plot are the *fold_change* on the X-axis and the *neglogp* on the Y-axis. The scatter plot view is chosen for visualizing gene-expression data, while the hierarchy tree is chosen for the categorical data.

3.4.2 Choosing the selection bias

The second step is to choose the selection biases in each view to specify the desired kind of generalizations. Biases are specifications of patterns for subset selection, and correspond to the interactive selection operators of visualizations.

Different visualizations afford different types of subset selections. A scatter plot might afford different two-dimensional region selections, such as squares, rectangles etc. A hierarchical tree might afford selections of detail at varying levels of tree depth. Users choose the visualization and bias based on what kind of generalizations they are interested in. In the example shown in Figure 24, the hierarchical tree constrains the user to select only subsets of genes, grouped by their functional categories.

3.4.3 Rule generation

At this point, a data-mining algorithm can accept the data and biases as inputs and begin a search process to find generalizations. In the example shown in Figure 24, the scatter plot sends the cutoff value (below which the expression of the genes is considered insignificant; shown in magenta color) to the Data Miner. The Data Miner (at the bottom of Figure 24) uses this information to filter out data and find patterns within the refined data.

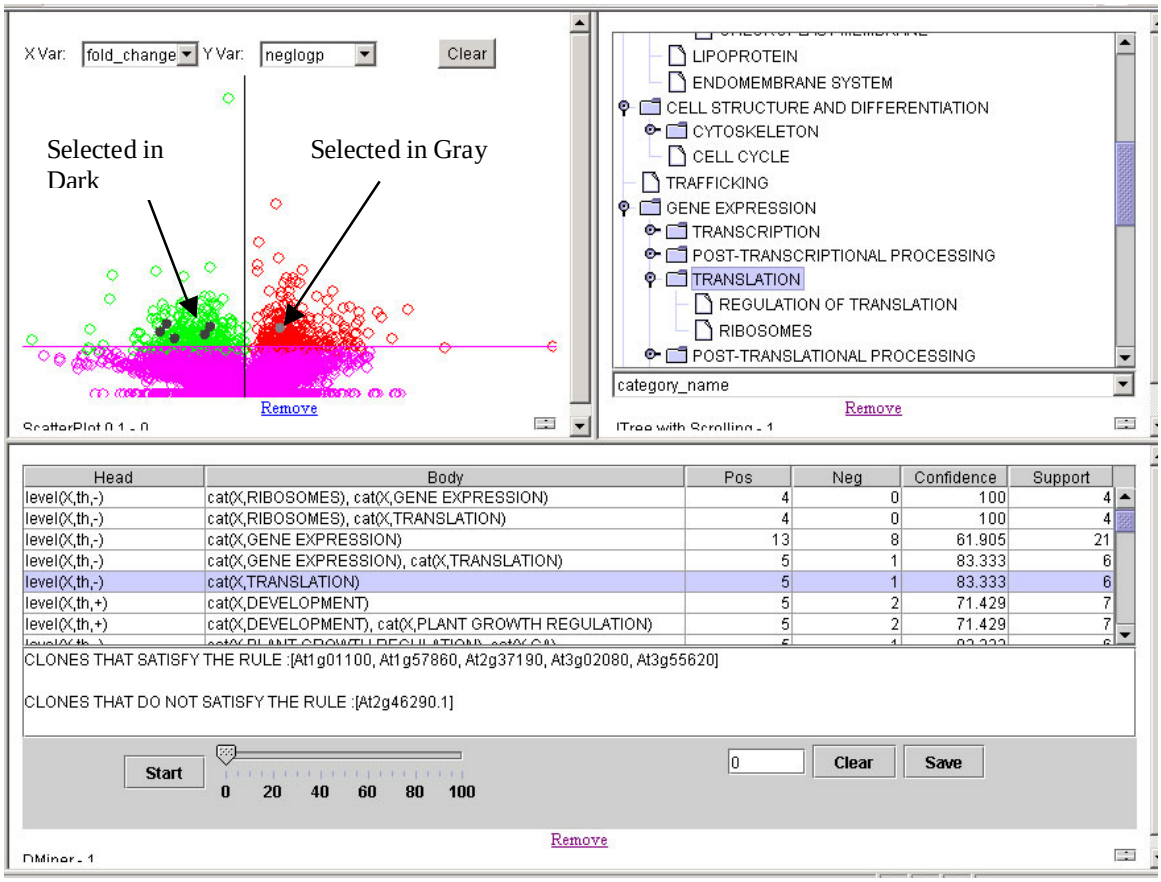

Figure 24: Interactive visual data exploration and mining using Fusion

3.4.4 Evaluation Criteria

Evaluation criteria can be used to judge the quality of the data mining results. A common evaluation measure is support. For example, support for the rule A :- B (A if B) is given by n(B). Users can specify a minimum threshold for the support measure. This enables them to filter out many unsupported rules. In the example shown in the Figure 24, the slider in the Data Miner can be used to filter out less confident rules.

3.4.5 Rule Visualization

Rules returned by the algorithm can then be viewed within the visualizations. Rules can be listed in a tabular form or can be presented as an overview. They can also be sorted by

their measure of evaluation criteria. The numerical evaluation measures can be visualized in many ways using histograms, pie charts etc. Users can select individual rules to display the brushed generalization in the visualizations. This displays the bias instances in each view, revealing the generalization. For example, in Figure 24, the ILP rules found by the Data Miner are listed in a tabular format. The user can select a rule in the table and see the genes and the categories comprising the rule in the other visualization components. The genes that satisfy the rule are shown in black and the genes that do not satisfy the selected rule are shown in gray.

3.4.6 Interactive discovery feedback loop

Fusion enables an interactive discovery feedback loop. When the user first begins brushing, the algorithm may return rules that are too specific or too general, or not well supported. As a result, users can adjust the biases. If results are too general (Figure 16), or too specific, (Figure 15), users can further constrain biases, to find more interesting results. If results have very less support, then the biases might have been too constrained and should be adjusted. This enables the user to guide the data mining towards more interesting results. In the example shown in Figure 24, if the user selects a very high threshold value, lot of genes will be regarded as insignificantly expressed and so the Data Miner might find very less number of rules within the unfiltered genes. So the user can adjust the threshold if the results are very less.

The underlying theory of Fusion is thus based on the analogy between the common concepts in visualization and data mining, allowing the user to take advantage of strengths of both of them.

Chapter 4

4 Fusion User Interface

As discussed in Chapter 3, the underlying theory of Fusion is based on the analogy between visualization and data mining. Fusion user interface enables the user to work in both interactive visualization and batch data mining modes, while allowing them to smoothly switch between them. Figure 25 shows different parts of the Fusion user interface.

Fusion User Interface comprises of three parts:

1. Visualization Schema
2. Data Schema
3. Visualization Workspace

Visualization Schema

Fusion Visualization Schema is an extension of the Snap visualization schemas. A sample visualization schema is shown in Figure 26. Snap visualization schemas provide the user with the capability to create a visual concept for the organization of interactions supported by the multiple view visualizations [13]. The Visualization schemas layer visually provides the structure of the coordinations along with providing an interface for manipulating that structure [13]. The Visualization Schemas layer is shown in Figure 25. Fusion extends the Snap Visualization schema by adding new data mining ports, for communicating data mining bias inputs and discovered rules across the components.

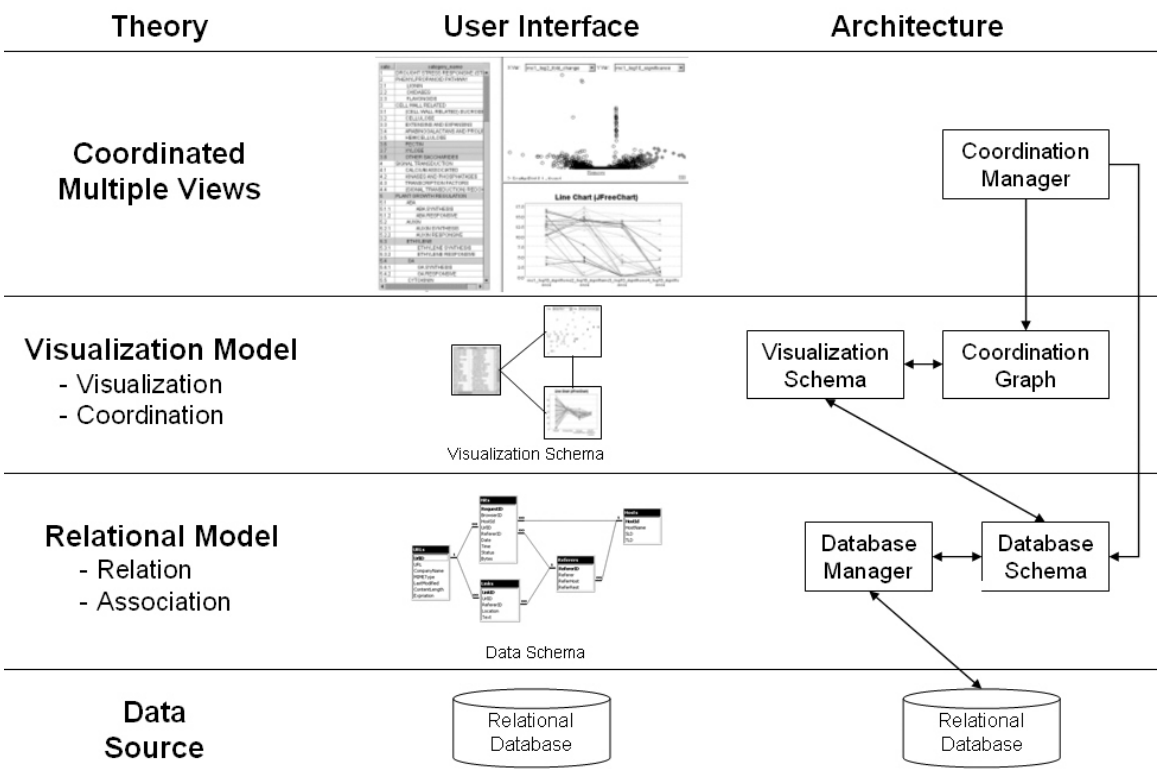

Figure 25: Visualization Schemas layer provides a framework for visual construction of coordinations [13]

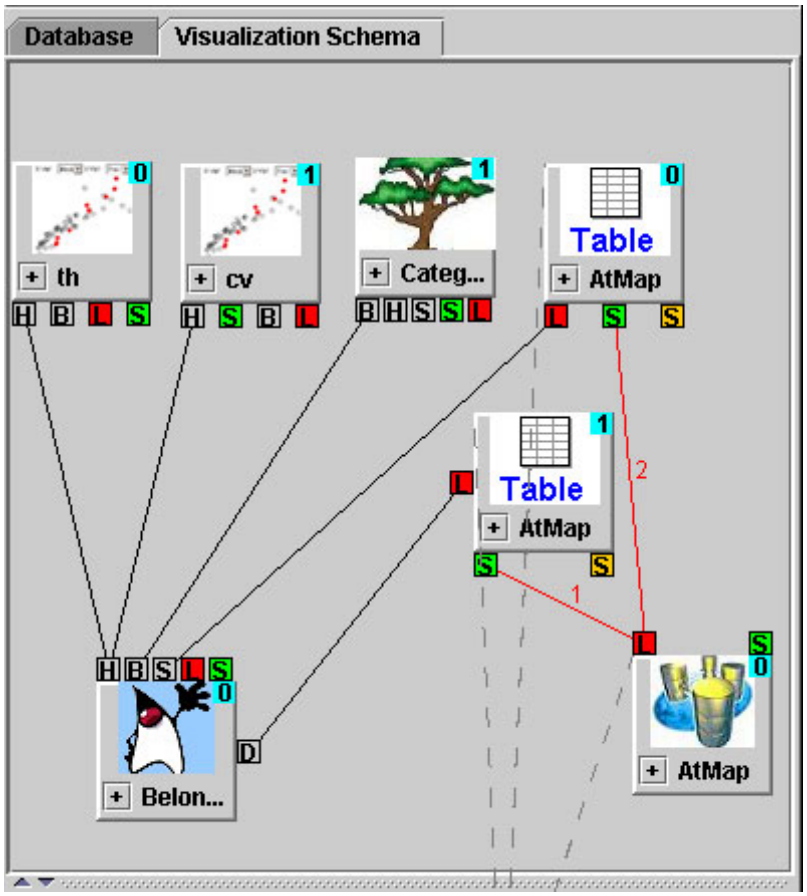

Figure 26: Visualization Schema

Data Schema:

The data schema gives an overview of the underlying database, and also gives details-on-demand. The nodes represent relations and the edges represent database relationships. Users can drag all the attribute of a table to the visualization schema node by dragging the '^' symbol at the right hand side of each node. They can also choose which attributes to drag from the attribute list and then drag only those attributes. The attribute list displays all the attributes for a selected visualization [13]. A sample data schema is shown in Figure 26.

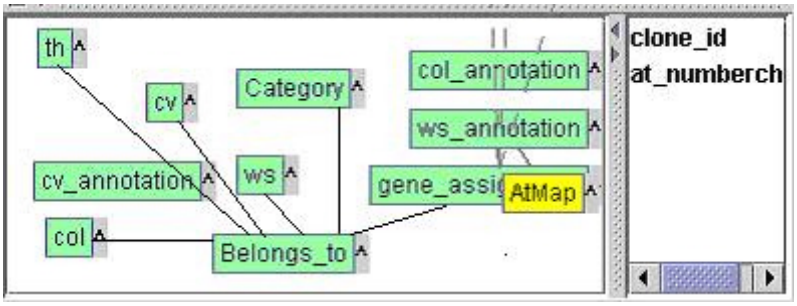

Figure 27: Data Schema: Shows an overview of the data base

Visualization Workspace:

The Fusion visualization workspace uses the same workspace as the one used for Snap. The Snap visualization workspace allows the user to choose Visualization components and arrange across it. A sample visualization workspace is shown in right portion of Figure 28.

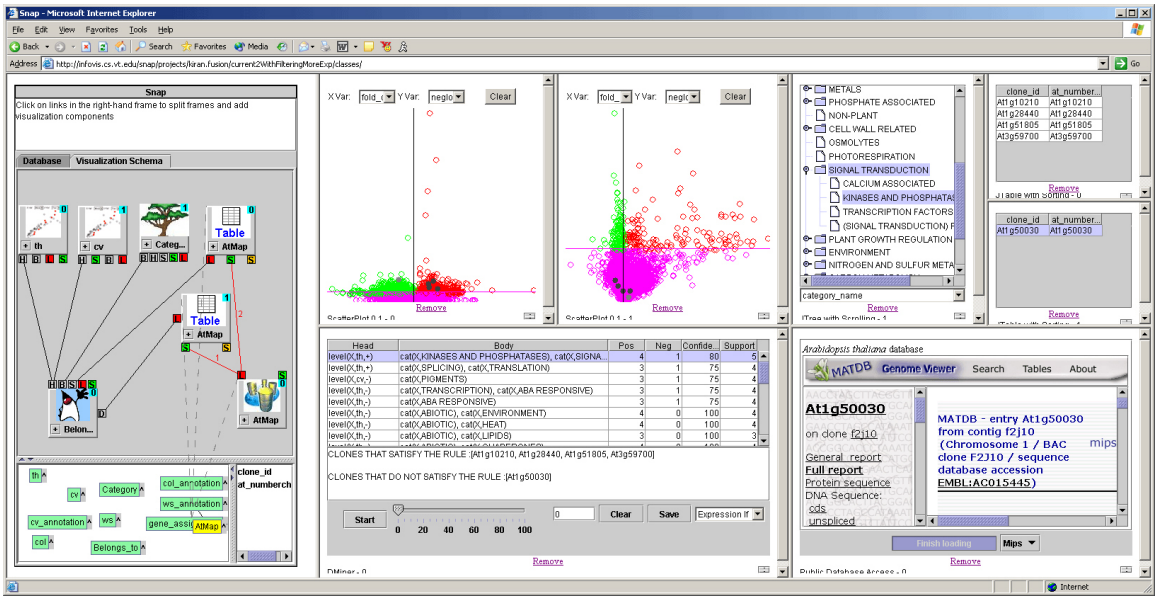

Figure 28: Fusion User Interface, showing the Visualization Schema (center left), Data Schema (bottom left) and Visualization workspace (right).

4.1 Fusion Visualization Components

Each of the components useful in gene-expression analysis and mining is discussed in the following sections.

Existing components in Snap

1) Tree
2) Table

Existing components in Snap that were customized in Fusion

1) Scatterplot (customized for gene-expression data)

New components in Fusion

1) Data Miner
2) Public Database

4.1.1 Existing components in Snap

4.1.1.1 Tree Component For Visualizing Functional Category Hierarchy

This component is used for visualizing hierarchical categorization of the gene functional categories.

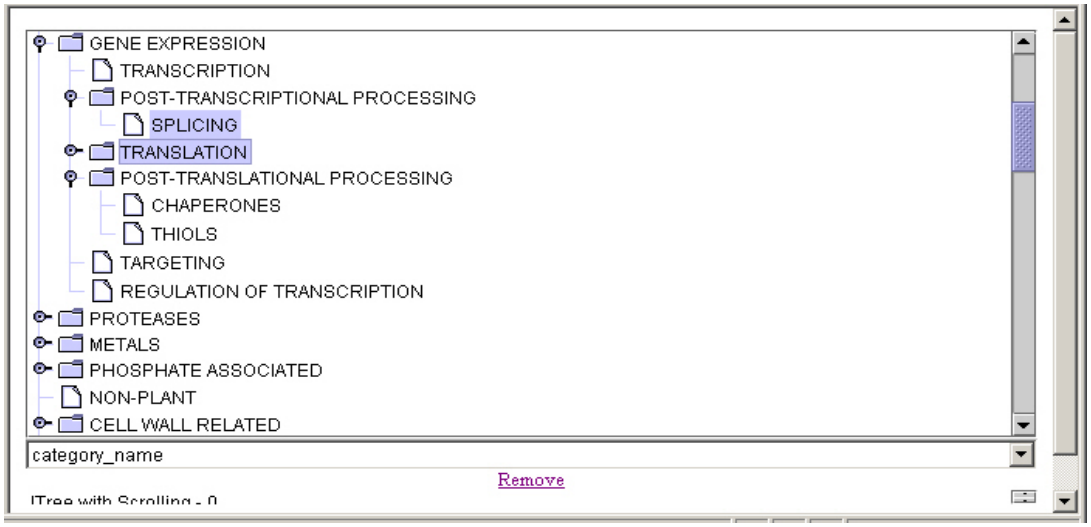

Figure 29: The hierarchical tree that facilitates the visualization of functional categories

Subcategories in the category tree can be seen by clicking on the expandable icons in the tree. The user can also select certain categories of interest and see the behavior of the genes belonging to those categories in the scatter plots.

4.1.1.2 Table Component

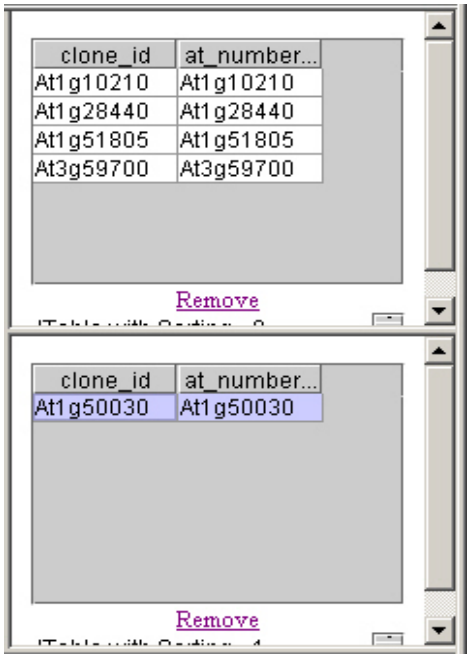

Figure 30: Table components, the top one showing the names of the genes that satisfy the selected rule, the bottom table showing the genes that do not satisfy the selected rule

Table components can be used for finding the names of the genes and their annotations. They can also be used to see the names of the genes satisfying a rule and those that do not (Figure 30).

4.1.2 Existing components in Snap, that were customized in Fusion

4.1.2.1 Scatterplot for Expression Level Visualization

Expression level visualization is facilitated by visualizing the data in a scatter plot, the fold_change on the X-axis and the significance on the Y-axis, as shown in Figure 31. The red colored genes are the genes classified, by the draggable red bias line, as positively expressed by the biologist. The green colored genes are the genes classified, by the

draggable green bias line, as negatively expressed by the biologist. The magenta colored genes are the genes filtered out, by the draggable magenta bias line, based on by their statistical significance.

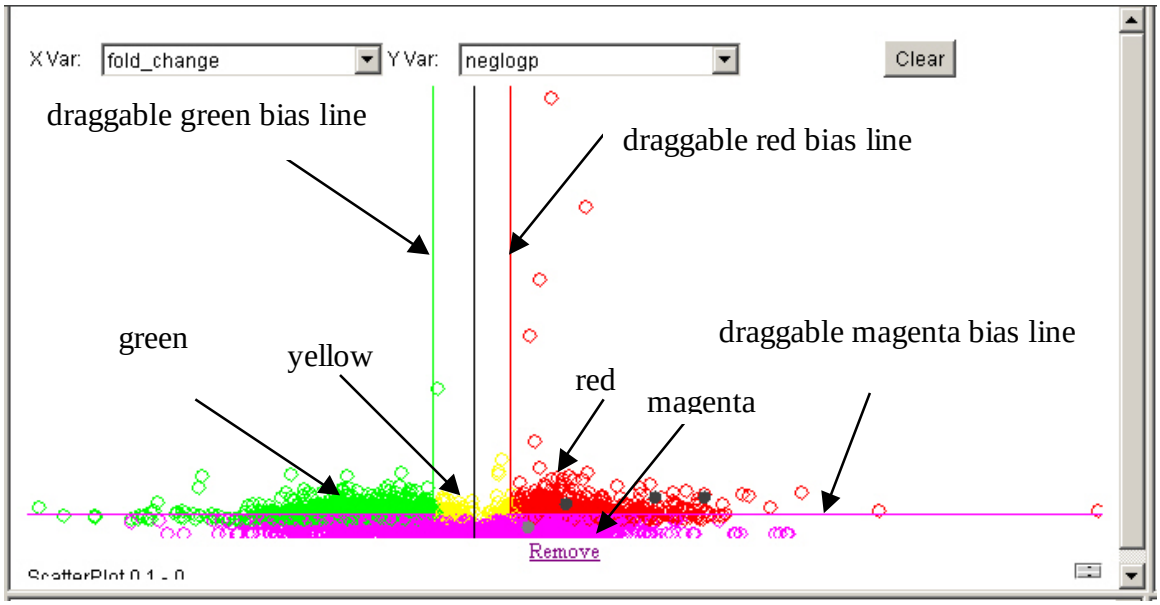

Figure 31: The scatter plot shows the *fold_change*(on the x-axis) and the significance of the gene expression, *neglogp* on the y-axis.

The yellow colored genes are the zero classified genes, which are the unfiltered genes that are neither positive nor negative. Normal selections of the genes are shown in the black. When visualizing rules, the genes that satisfy the selected rule are shown in black and the genes that do not, are shown in light gray.

4.1.3 New components in Fusion

4.1.3.1 Data Miner Component

The Data Miner component consists of a tabular component that displays the various attributes of the rules found by Proteus. The rule text box shows the names of the genes that satisfy the rule, and the genes that do not satisfy the rule. The user can start the Data Mining process by clicking the “Start” button. She can filter out the rules, based on their confidence by using the slider. Rules can be sorted by each of the attributes by clicking on the column header of each of the attributes.

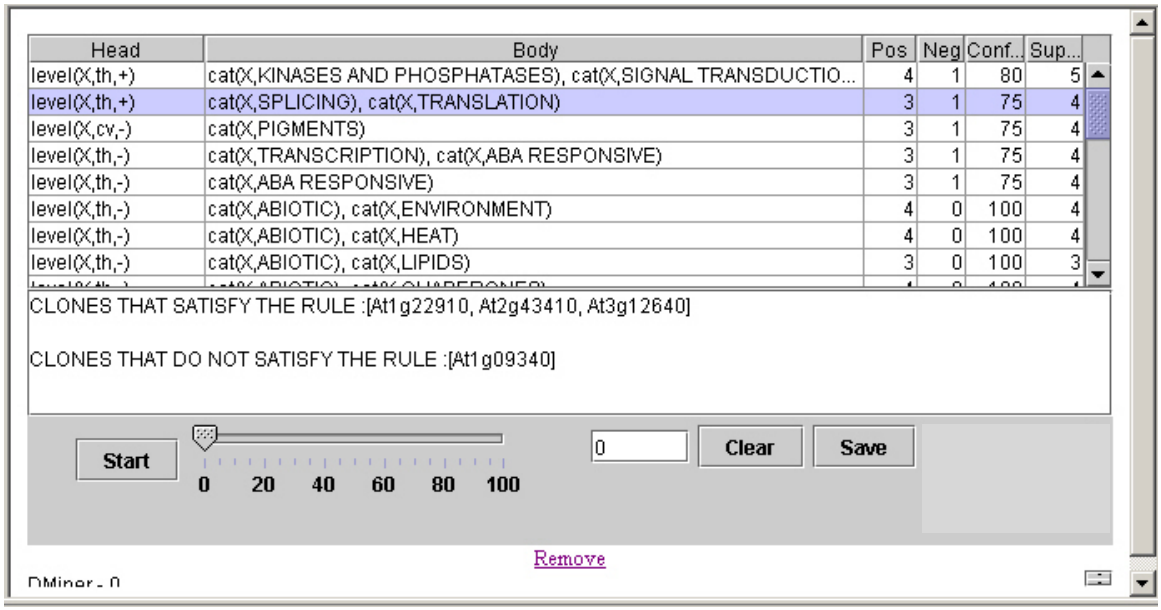

Figure 32: Data Miner component allows for the filtering and visualization of the selected rule.

4.1.3.2 Public Database Component

The public database component helps the user to search for genes in public databases. They can also validate their own annotations with the annotations found from the public database search results. The user can choose to search across a variety of public databases like Munich Information Center for Protein Sequences (MIPS), The Institute for Genomic Research (TIGR) and The Arabidopsis Information Resource (TAIR) from the menu at the bottom of the component.

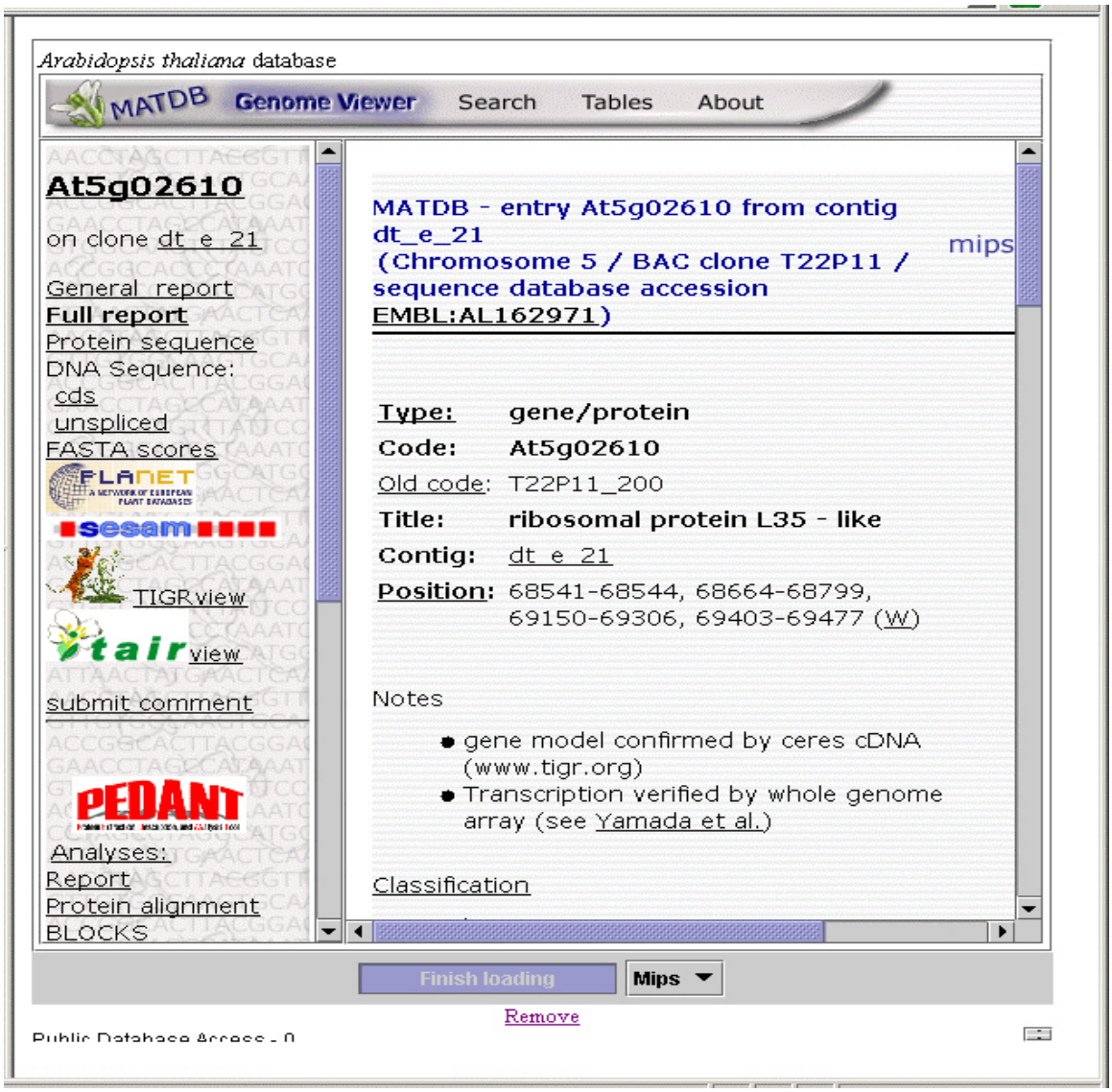

Figure 33 : Public database component that shows the results of the public database search

4.2 Building the Visualization Schema

Visualization schema diagrams are visually represented similar to data schema diagrams. Visualization schemas are represented as a graph, and support direct manipulation [13]. Nodes in the graph represent instantiated visualization components. Edges represent coordinations between components [5].

4.2.1 Nodes

Opening a new visualization component adds a node to the graph. The actual instantiated visualization component that the node represents is displayed in a separate frame for the multiple-view visualization, called the visualization workspace, tiled next to the visualization schema diagram. Nodes indicate the visualization component type, a miniature icon of the component, and the name of the data relation displayed in the component (Figure 34) [13]. Each port displays a descriptor of its corresponding action. User can see the complete action name of the port by moving the mouse over it.

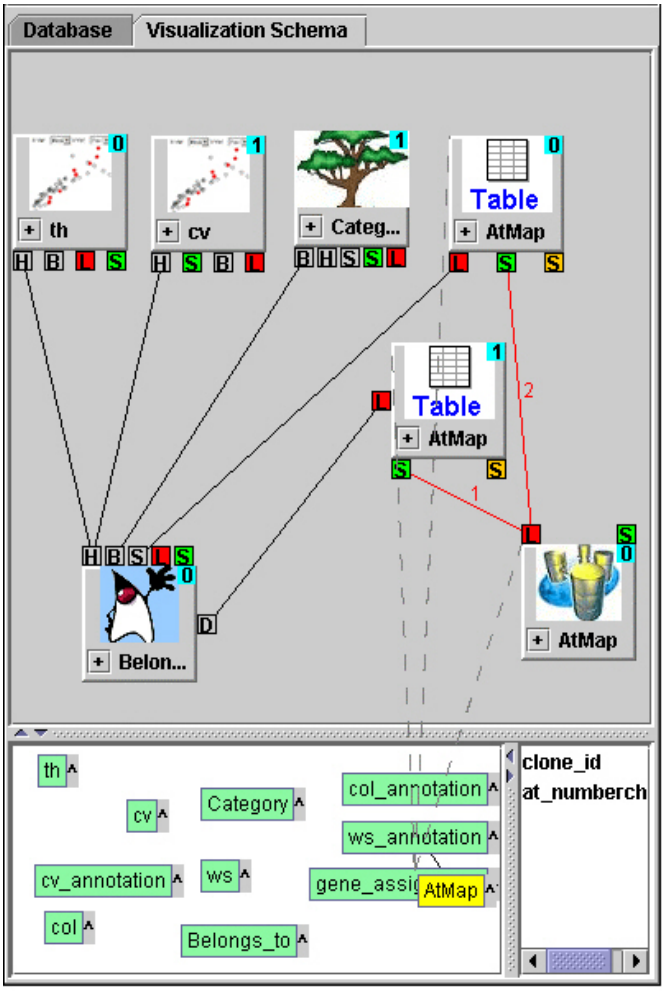

Figure 34 : Visualization Schema (top) and the underlying Data Schema (bottom)

Users can drag a relation from the data schema onto a component's node to display that relation in the component.

4.2.2 Edges

Edges between nodes in the visualization schema represent coordinations between visualizations. Edges are established by dragging a link from a port on one node to a port on another node.

The different ports supported by the current Fusion system can be classified into two types.

1) Normal Visualization Ports
2) Data Mining Enabled Ports

Normal Visualization Ports:

The normal visualization ports are Select, Load and Highlight. They were part of the Snap system. The user uses the Select, Load and Highlight ports to establish coordinations between the visualization components. These ports support the interactive data exploration mode.

Data Mining Enabled Ports

New ports have been added to the Fusion system to support data mining. These ports support bias specification and rule visualization.

- **H(Head) port:**
    - The H port connects the underlying table comprising the Head (left hand side) of the ILP rule to the Data Miner.
- **B(Body) port:**
    - The B port connects the underlying table comprising the Body (the right hand side) of the ILP rule to the Data Miner.

- **S(Satisfy) port:**
    - The Data Miner fires the gene names that satisfy the selected rule on the Satisfy port.

- **D (Do Not Satisfy) port:**
    - The Data Miner fires the gene names that do not satisfy the selected rule on the Do Not Satisfy port.

4.3 Sequence of User Interactions

The following is the sequence of user interactions:

1) Connection to the database
2) Dividing the visualization workspace
3) Loading the visualization components and selecting the attributes to be visualized in each component.
4) Establishing co-ordinations between the visualization components, by building the visualization schema.
5) Interactive visualization exploration and data mining.

4.3.1 Connection to the database

The user can select the data source from the drop down list. A brief description of the database is also displayed. The user connects to the database by clicking on the "Connect" button, as shown in Figure 35. The user can connect to both local and remote ODBC databases [13], like MS SQL, MSAccess, MS Excel etc.

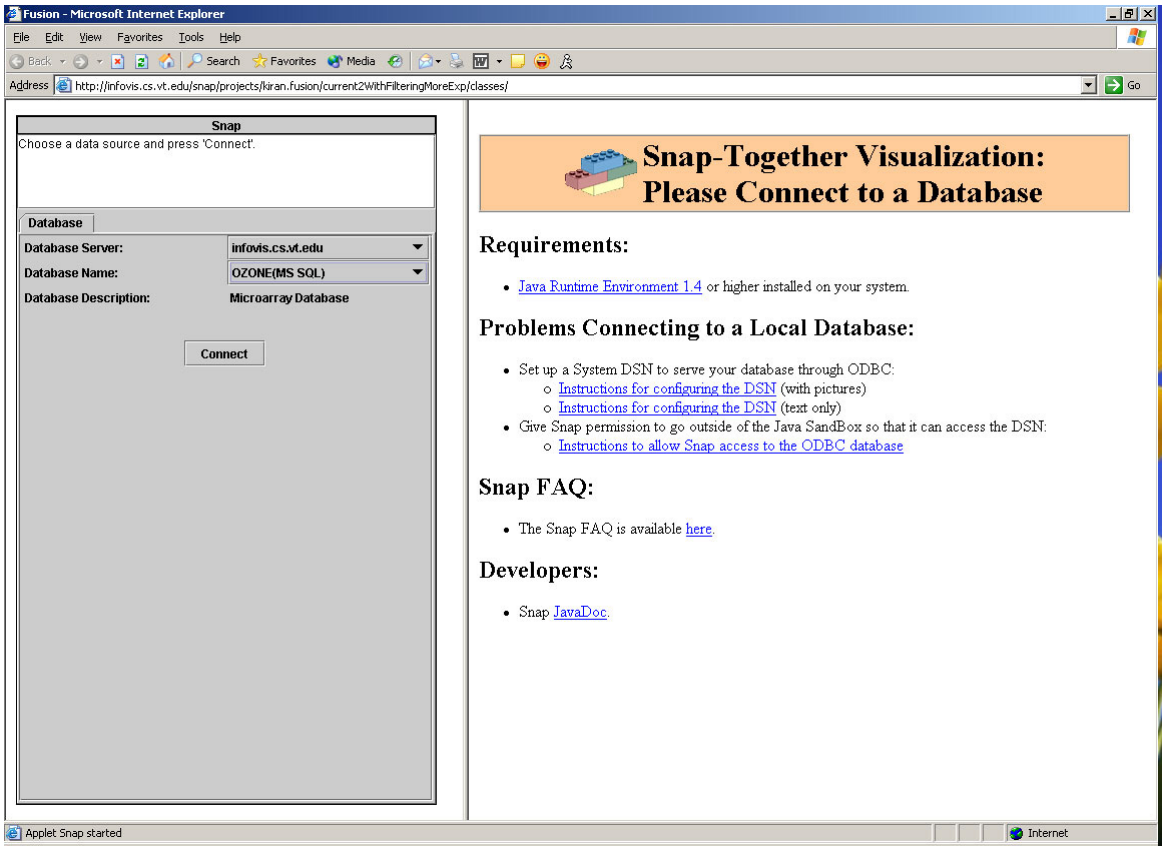

Figure 35: Connection to the database

4.3.2 Dividing the visualization workspace

An empty visualization workspace opens up on the right, and visualization schema on the upper left and data schema on the lower left. The user can divide the space as many times as needed by clicking on the "Horizontally" and "Vertically" links, as shown in Figure 36.

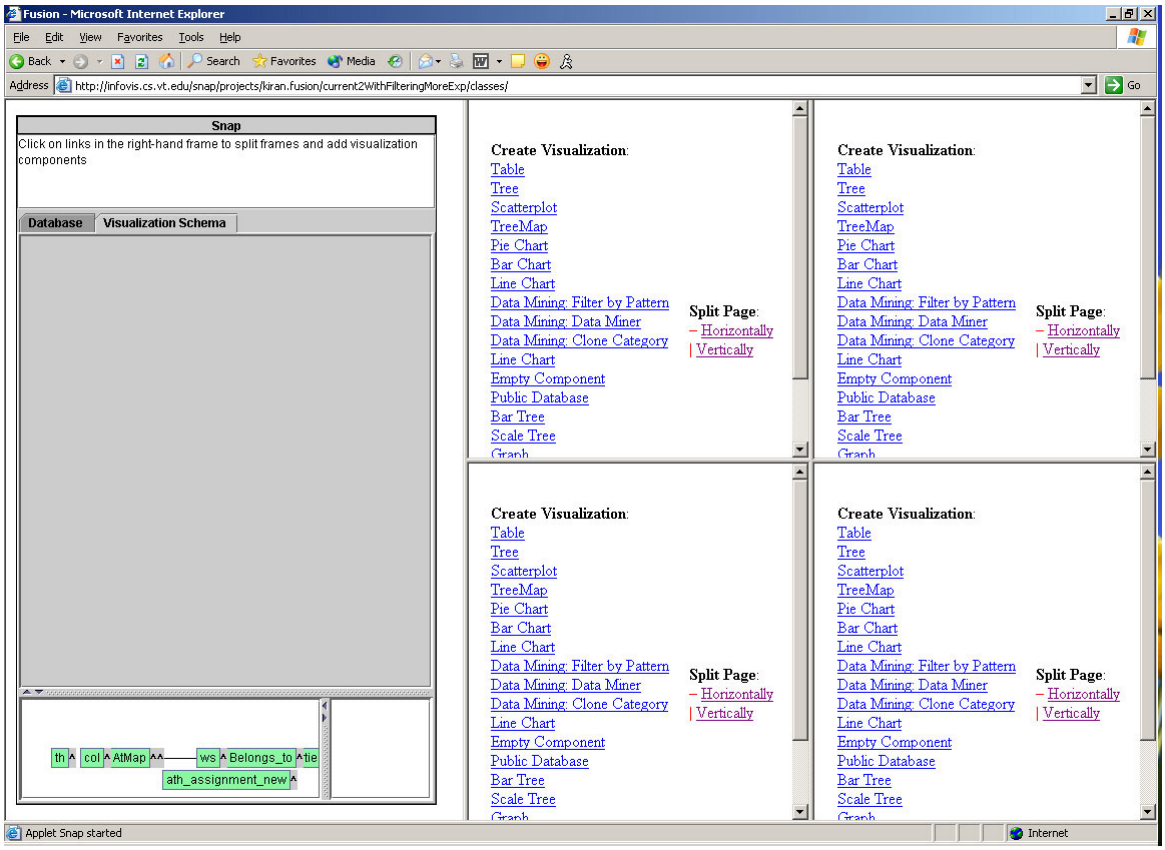

Figure 36: Dividing the visualization workspace

4.3.3 Loading the visualization components and selecting the attributes to be visualized in each component

In the example shown in Figure 37, the user selects the scatterplot by clicking on the "Scatterplot" link. The user loads the attributes by selecting the table *th* and columns to be loaded (*fold_change* and *neglogp*) in the data schema and drag-and-dropping onto the icon corresponding to the scatterplot in the visualization schema. *fold_change* refers to the ratio of gene-expression in treated vs. control, *neglogp* refers to the statistical significance (discussed in Chapter 1).

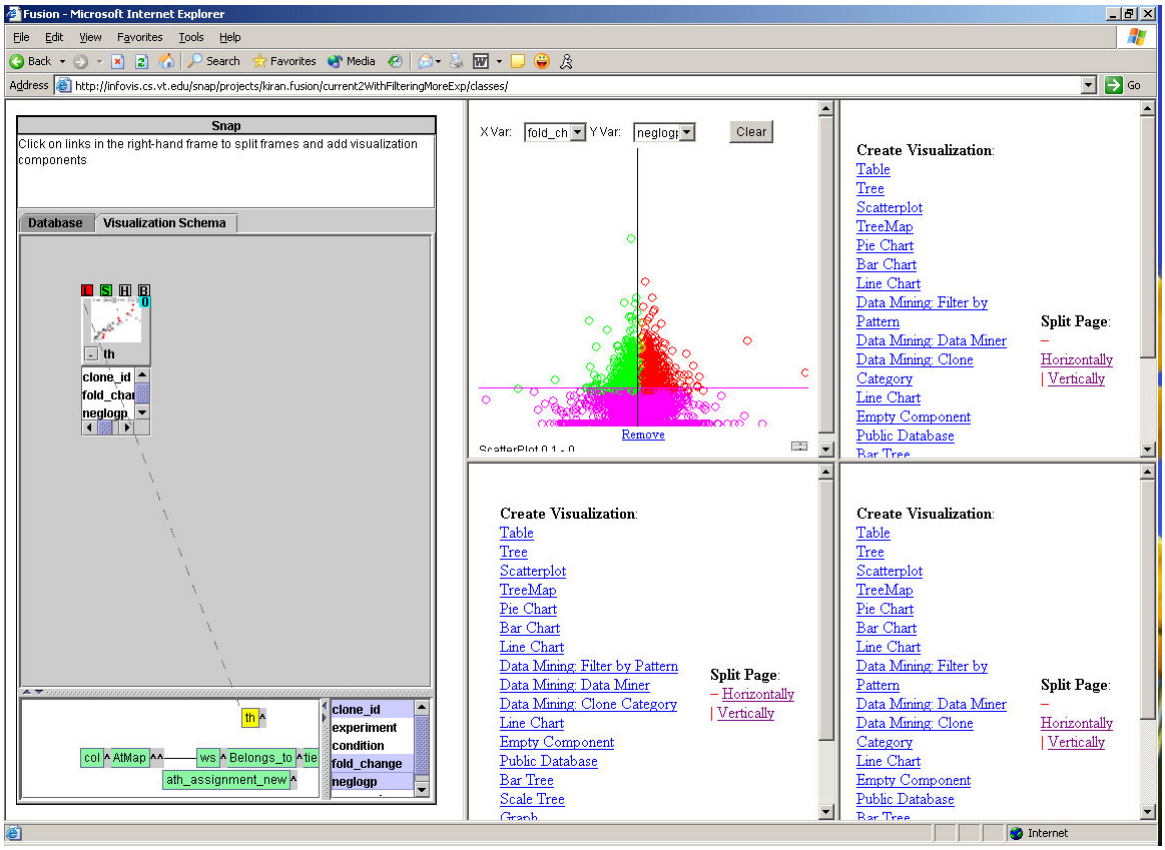

Figure 37: Loading the visualization components

4.3.4 Establishing co-ordinations between the visualization components, by building the visualization schema.

In the example shown in Figure 38, the user connects the Select (coded as S) ports of each of the components, so that the user can select some genes in the Scatter plot and see the corresponding selection in the functional category tree.

4.3.5 Establishing the co-ordinations for the mining mode

The current mode is an interactive exploration mode where user can select genes-of-interest in one component and see their corresponding categories in the other components.

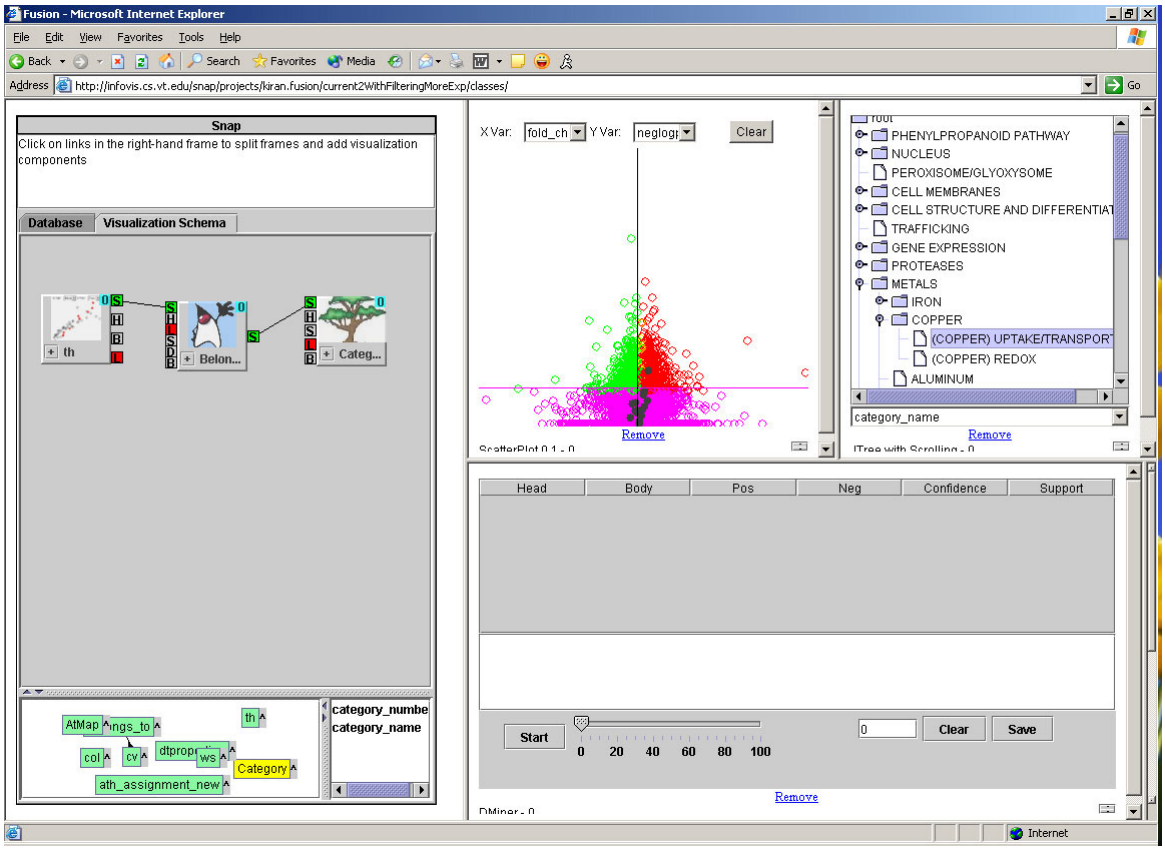

Figure 38: Interactive data exploration mode

User can begin the data-mining mode by linking the H ports of Scatterplot and Data Miner and the B ports of Data Miner and the Tree component, as shown in Figure 39. The user then adjusts the threshold for filtering out the genes (which is –log (0.05) for 95% confidence, by default) by dragging the line with the help of the mouse. The data-mining algorithm can be started by clicking the "Start" button.

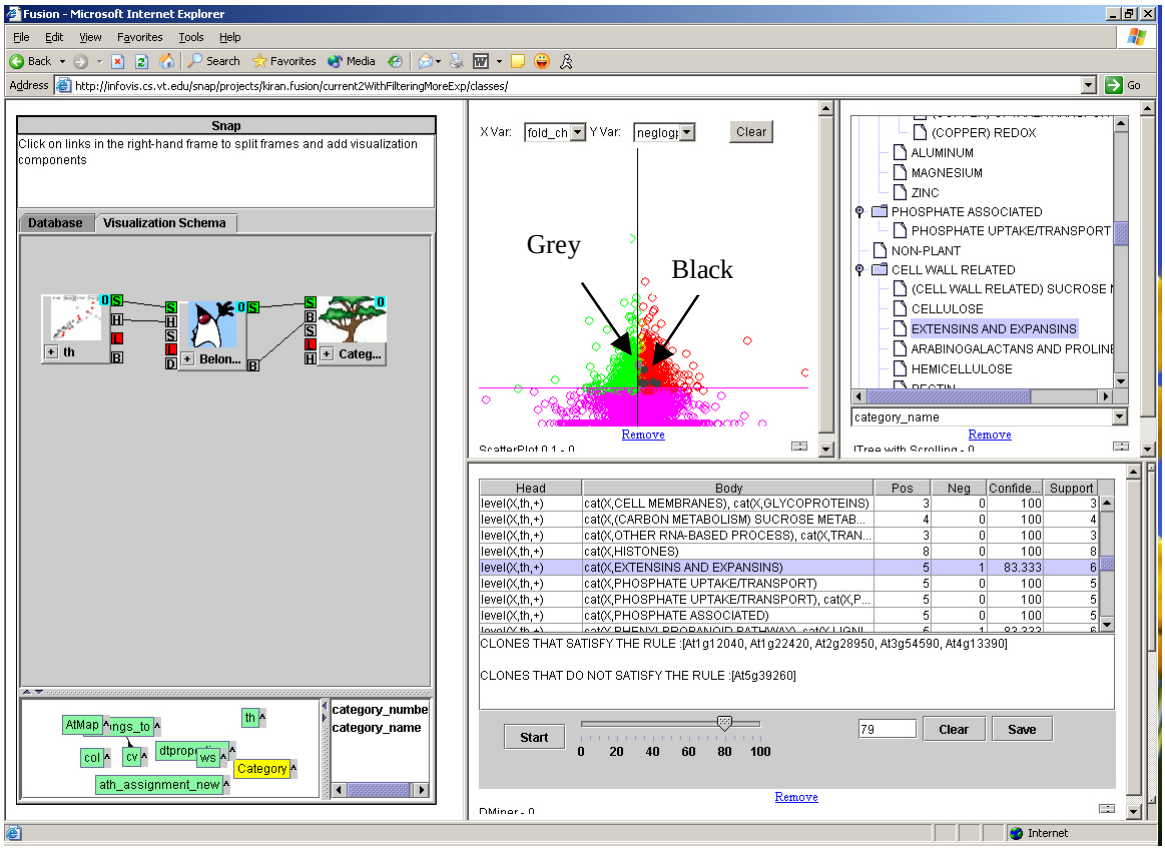

Figure 39: Interactive visualization and batch data mining mode

4.3.6 Interactive visualization and batch data mining mode

The results of the algorithm are displayed in a tabular view as shown in Figure 39. The user can select a rule in the table and see the genes and the categories comprising the rule in the other visualization components. The genes that satisfy the rule are shown in black and the genes that do not satisfy the selected rule are shown in grey, on the scatter plot. The constituent categories are highlighted in the category tree. The rules can be sorted by various measures like confidence, support etc. They can also be filtered by using the slider bar. If the results are very low in number, the user can lower the threshold and rerun the algorithm. In this way the user can, iteratively refine the rule-finding process.

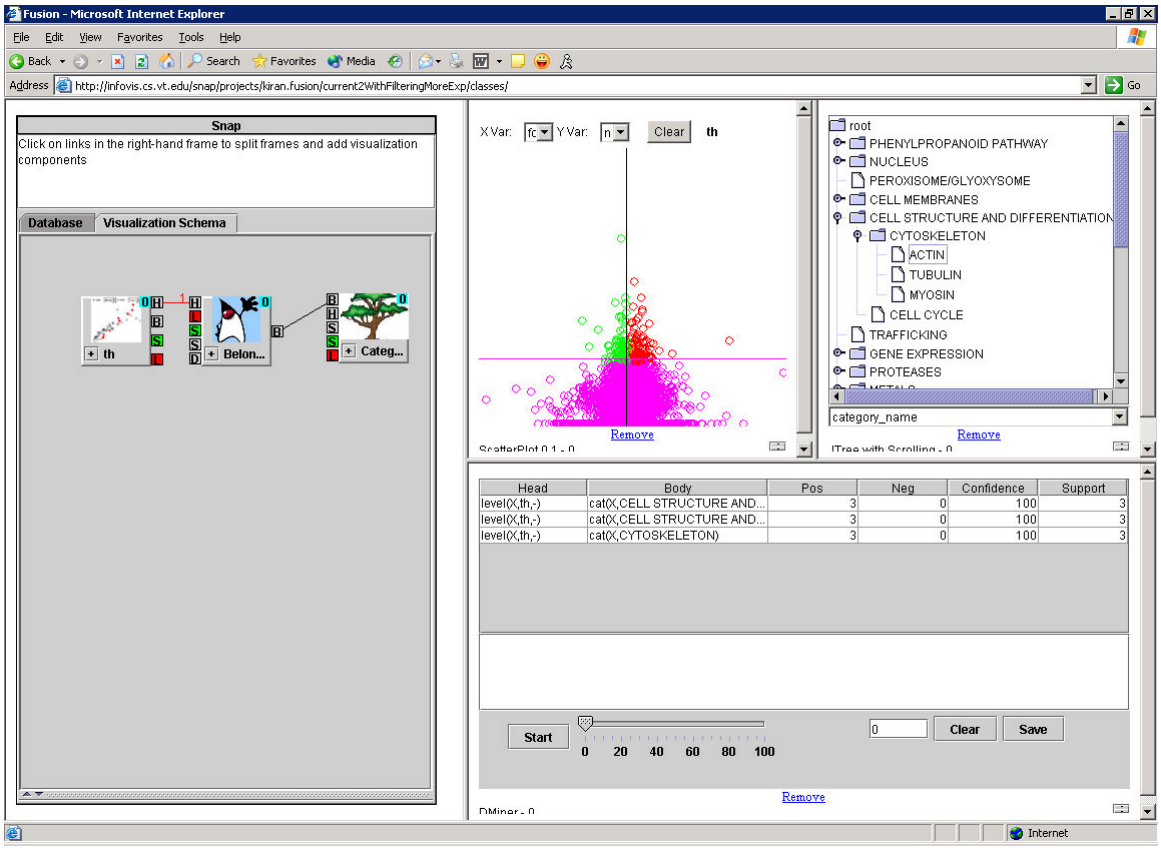

Figure 40: Scatterplot with high threshold

For example, in Figure 40, the threshold bias (the magenta line) is set to very high due to which Proteus gives very few results. Generally the threshold is set to (-log(-0.05) = 1.30). But the users can set it to high, if they want to be more stringent.

The Fusion user interface helps the user interactively explore and mine the data by providing a flexible multiple-view visualization work space, visualization schema, data schema and various visualization components customized for gene-expression data analysis.

Chapter 5

5 Fusion Software Architecture

Fusion was built upon the existing Snap software architecture and was enhanced to support additional capabilities like rule visualization, bias specification and ILP rule mining.

The existing Snap architecture is based on the support for realization of the analogy between visualization and relational data base concepts. Coordination between two visualizations is mapped to a join between two tables linked by a foreign key relationship. This coordination is achieved by key conversion across related tables.

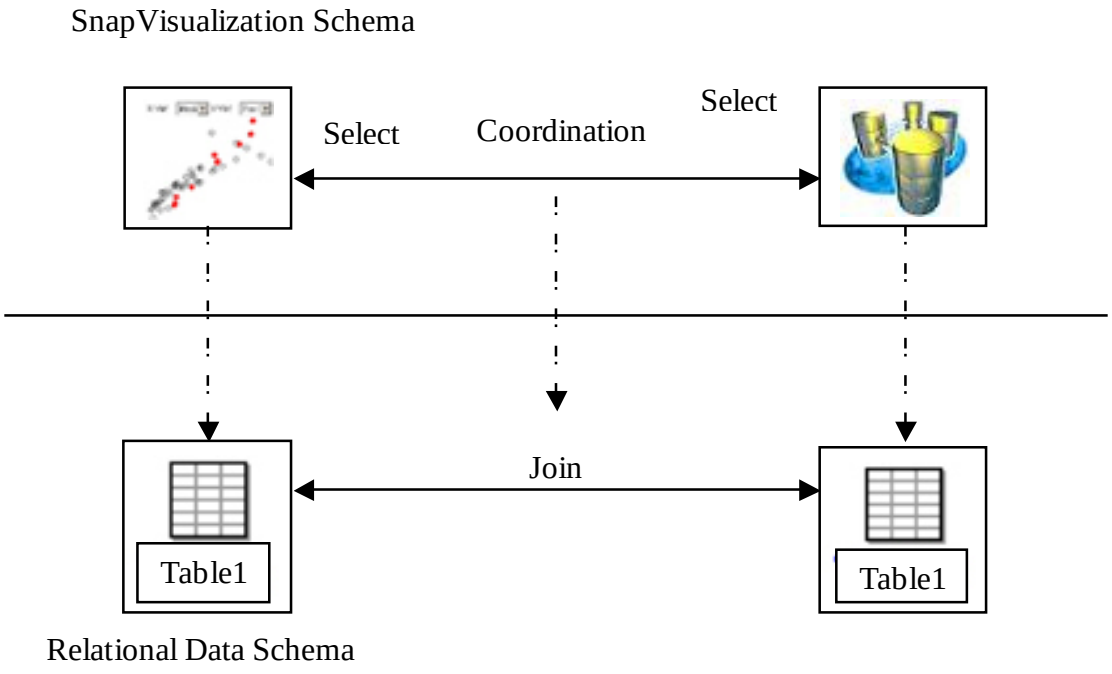

Figure 41: Snap Model

The Fusion model is an extension of the existing Snap model. The key idea of the extension is the bypassing of the normal key conversion for events overloaded to support bias specification and rule visualization. Fusion's capability to do normal key conversion for non-data mining events helps it to retain all the existing capabilities of Snap to perform interactive data exploration.

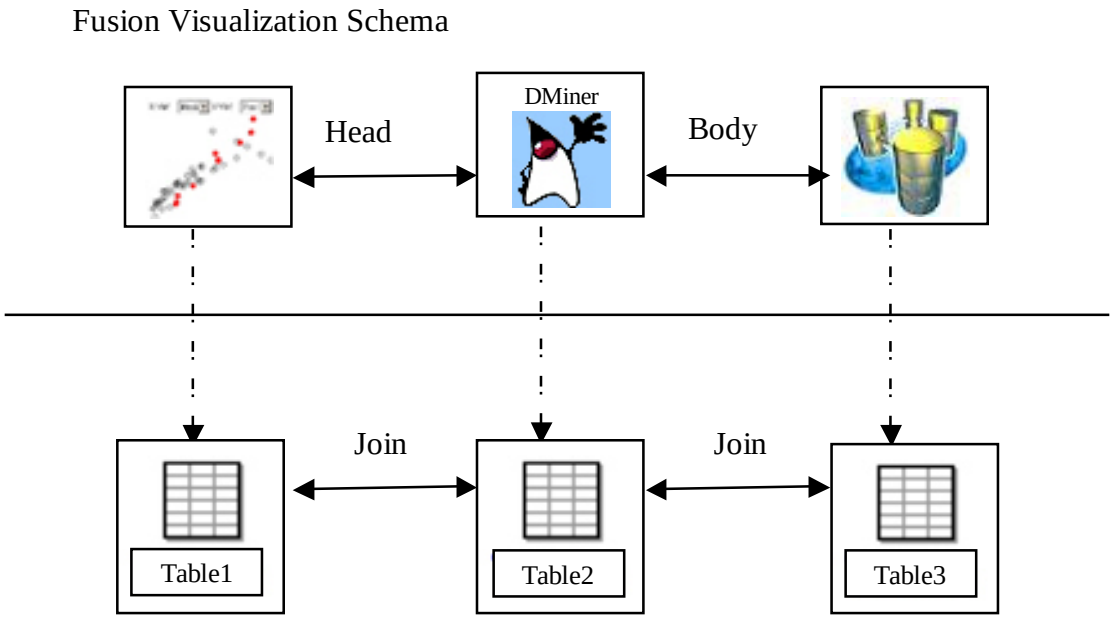

Figure 42: Fusion Model

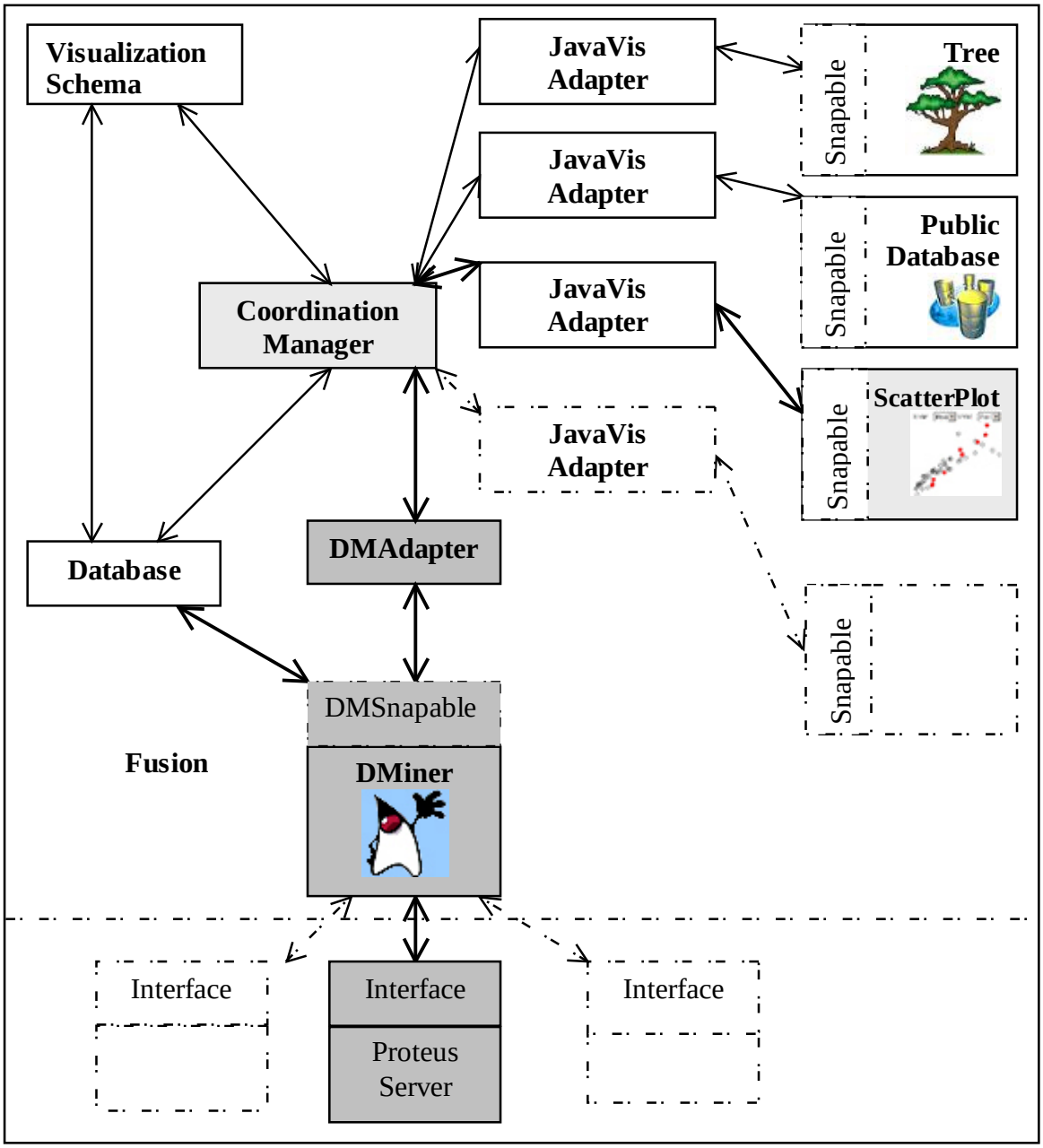

Figure 43: Fusion Architecture

Regions refer to the existing Snap architecture

Regions refer to the Fusion enhancements, made to the existing architecture

Regions refer to the Fusion additions, not present in the previous architecture

Existing communication in Snap

Enhanced communication in Fusion

5.1 Existing Architecture Summary

Snap has event-based, implicit invocation software architecture [12]. The visualization schema is used to coordinate events between the individual visualization components. When a view is added, Snap registers itself as a listener for the component's events. When users interact with a component, the component fires an event. Snap receives the event and propagates it to other coordinated visualizations. Snap acts as a mediator between each component [10]. Visualization components implement a "*Snapable*" programming interface exposing the component's capabilities to Snap.

The Snap architecture consists of three major layers for coordinating components (Figure 44). The first layer of the architecture includes the Database Manager and Database Schema. They provide connectivity to data sources and describe join associations between the separate data relations. The second layer contains the Visualization Schema and Coordination Graph [13]. This layer supports the assignment of data relations to visualization components, and the coordination of events between components. The Visualization Schema allows two components to be coordinated only when their encapsulated relations can be associated by joins in the data schema. The third layer includes the Coordination Manager. It handles all communication with the visualization components, and is responsible for the receiving and firing of events. The Coordination Manager utilizes the bottom two layers to propagate events to coordinated components and translate events as needed.

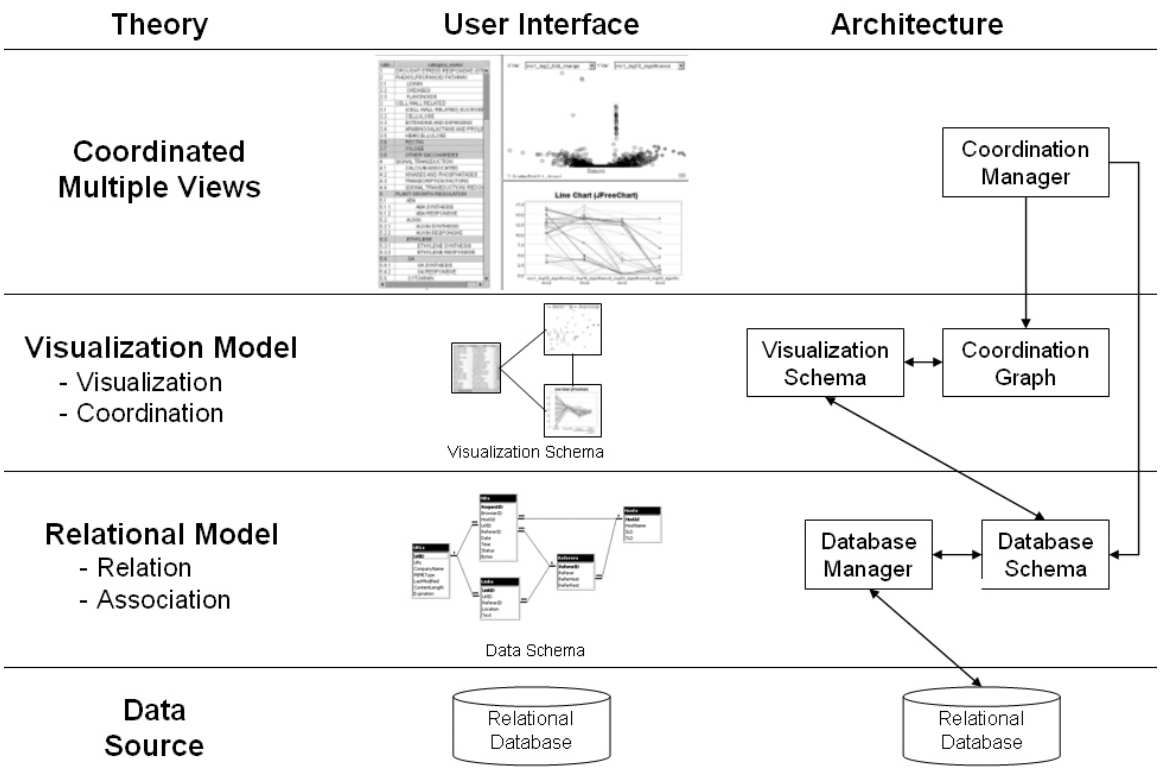

Figure 44: The Snap architecture integrates the data schema and visualization schema to enable automatic execution of coordinations in multiple-view visualization [13].

The Snap architecture requires visualization component developers to handle only firing and receiving events in their component with the Snap system. This shields developers from needing to communicate with any other visualization components directly. Visualization components must implement a standardized *Snapable* application programming interface (API). This API is designed to be very simple, and to minimize developers' required effort (Figure 45).

```
public interface Snapable
{
/** Called when Snap needs to load data into the visualization.
*
* @param rs the data to be loaded by the component
* @param primaryKeyColumnName identifies the column to be used when
throwing
* and receiving SnapEvents.
*/
public void load(ResultSet rs, String primaryKeyColumnName);

/** Method used by Snap to register as a listener for SnapEvents */
public void addSnapEventListener(SnapEventListener sel);
/** Method used by Snap to unregister as a listener for SnapEvents */
public void removeSnapEventListener(SnapEventListener sel);

/** Method called when a SnapEvent is to handled by the visualization */
public void performSnapEvent(SnapEvent e);

/** Returns a list of Actions (as Strings) supported by a visualization */
public Enumeration getSupportedActions();

/** Returns the Icon for the visualization */
public Icon getIcon();
}
```

Figure 45: Snapable interface

5.1.1 Event Translation

When coordinating events between two components that encapsulate different relations, event translation is needed to join the relations. In the scenario described in, events occurring in the tree-view visualization of Categories must be translated when coordinated to the components displaying Expression levels. When the tree-view of Categories fires a "select" action event, it sends Snap a list of the category numbers of the categories selected by the user. Snap then propagates the event to the scatter plot of Expression levels according to the coordination in the visualization schema. When firing

the event to the plot, Snap must first translate the category numbers to the associated clone_ids by performing a data join. The plot then receives the translated event, and highlights the appropriate genes in the scatterplot.

The Coordination Manager utilizes the visualization schema to propagate events and determine the relations encapsulated by each component. It then utilizes the data schema to translate events appropriately based on the underlying data join associations.

The JavaTechnologyAdapter manages the event handling thread and event queues. The adapter layer can also provide data transformation functionality for Snap. Typically, visualization components receive data relations from Snap in the form of a JDBC ResultSet. However, adapters can offer other data models by translating the data relations into TableModels, TreeModels, or other common data structures. This functionality can further simplify the requirements for integrating new components into Snap.

5.2 Fusion: Enhanced Architecture to support Data Mining tasks

The existing architecture has been enhanced to support bias specification, rule visualization and ILP rule mining.
The following changes were made to the existing architecture:

1) SnapEvent class was enhanced to carry biases.
2) Co-ordination Manager was enhanced to bypass the normal key conversion for bias communication.
3) DMSnapable interface was added to give database component to the Data Miner that implements it.
4) DMAdapter was added to preserve biases during the communication of the events.

Each of the changes made to the architecture is discussed in detail in the following sections.

5.2.1 SnapEvent

When the user specifies, a SnapEvent is fired which sends the bias information to the Data Miner. SnapEvent has been enhanced to encapsulate biases in the form of strings (Figure 46).

```
public class SnapEvent extends EventObject
{
protected String eventType;
protected Vector keys;
protected String biasString ="";

public SnapEvent(Object source, String eventType, Vector keys)
{
   super(source);
   this.keys = keys;
   this.eventType = eventType;
}
//overloaded SnapEvent to communicate biases
public SnapEvent(String eventType, String biasString, Object source)
{
 super(source);
 this.eventType = eventType;
 this.biasString = biasString;
}
}
```

Figure 46: SnapEvent enhanced to support bias specification and communication, the text shown in bold are enhancements over the previous version.

5.2.2 Coordination Manager

The Coordination Manager does the normal event translation for brush-and-link actions. But for data mining actions like, bias specification and rule visualization, Coordination Manager bypasses the event translation and key conversion and directly communicates the biases to the Data Miner. This is crucial for retaining previous capabilities of Snap, while supporting the data mining capabilities.

5.2.3 DMSnapable

To support rule visualization Data Miner needs access to the database. Data Miner extends the "DMSnapable" interface, which gives it access to the database.

```
public  interface DMSnapable extends Snapable
{

/** Called when Fusion needs to load data into the visualization.
* @param rs the data to be loaded by the component
* @param primaryKeyColumnName identifies the column to be used when throwing
* and receiving SnapEvents.
* @param dbm gives the component, access to the database
*/
public void load(ResultSet rs, String primaryKeyColumnName,DatabaseManager
dbm);

}
```

Figure 47: DMSnapable interface, overloaded to give database access to the component.

5.2.4 DMAdapter

DMAdapter is customized to preserve the biases during the communication of the SnapEvent to the DataMiner.

```
/** Enqueues the SnapEvent to be performed by Snap */
public void snapEventOccured(SnapEvent e)
{
if(!ignoreVisualizationEvents) // Used to ignore component talkback
{
SnapEvent event ;
 if (e.isDataMining())
{
event = new SnapEvent(e.getEventType(), e.getBiasString(),this );
event.setDataMining();
}
else {
event = new SnapEvent(this , e.getEventType(), e.getKeys());
}
```

Figure 48: DMAdapter, enabled to preserve biases. If it is a bias specification event, then its biases are preserved. If not, the keys are preserved (normal SnapEvent).

5.2.5 Customized Version of Fusion for Microarray data analysis

The Fusion system has been customized for microarray data analysis The DataMiner component is the most customized part of the Fusion system. The DataMiner Component collects the biases, sent to it by various components. Using the biases (constraints) and the database, the Data Miner composes a data table that satisfies the user biases. The user starts the data mining process by clicking the "Start" button. The Data Miner then opens a network socket and starts communicating with the Proteus server interface (Figure 43). The Data Miner sends the rule schema and the input data table to the Proteus server. Since the Proteus server need not have to be connected to the database, it gives the user flexibility to mine the data of choice by using Fusion.

Rule Schema

LHS*level(+clone_id,#comp,#expr)

RHS1*level(+clone_id,#comp,#expr)

RHS2*cat(+clone_id,#category)

depth*2

support*0.6

Sample Data Table sent to Proteus

clone_id	comp	expr	category
PINE2_CLONE_1007759495_101	Mc1	+	ACTIN
PINE2_CLONE_1007759495_102	Mc1	+	TRANSLATION
PINE2_CLONE_1007759495_103	Mc1	-	CELLULOSE
PINE2_CLONE_1007759495_103	Mc1	-	COLD
PINE2_CLONE_1007759495_103	Mc1	-	BIOTIC
…..	..	..	…

Proteus uses the Rule Schema and the Data table and computes the ILP rules. After the rule computation is over, the Proteus server interface sends the rules to the Data Miner. The Data Miner then notifies the user that the results have been received through a dialog box. It then displays the rules in its tabular component.

In the following section, we discuss how Fusion architecture supports the process of interactive data exploration and data mining with the help of the scenario that was discussed in 4.5.5 of the Fusion User Interface chapter.

5.3 Execution

5.3.1 Interactive data exploration

The user selects the category (COPPER) UPTAKE/TRANSPORT in the category tree. This fires a Select event, along with the category numbers (primary leys of the underlying table c*ategory*) of the selected categories. The Coordination Manager converts the keys to primary keys (*ccid*s) of the *belongs_to* table(the underlying table for the Data Miner).

The Co-ordination Manager then reconverts the *ccid*s to the primary keys of the *cv* table(the underlying table for the scatter plot). Since the event type is select, the selected genes (the converted-selected clone_ids) are highlighted in the scatterplot, as shown in Figure 50.

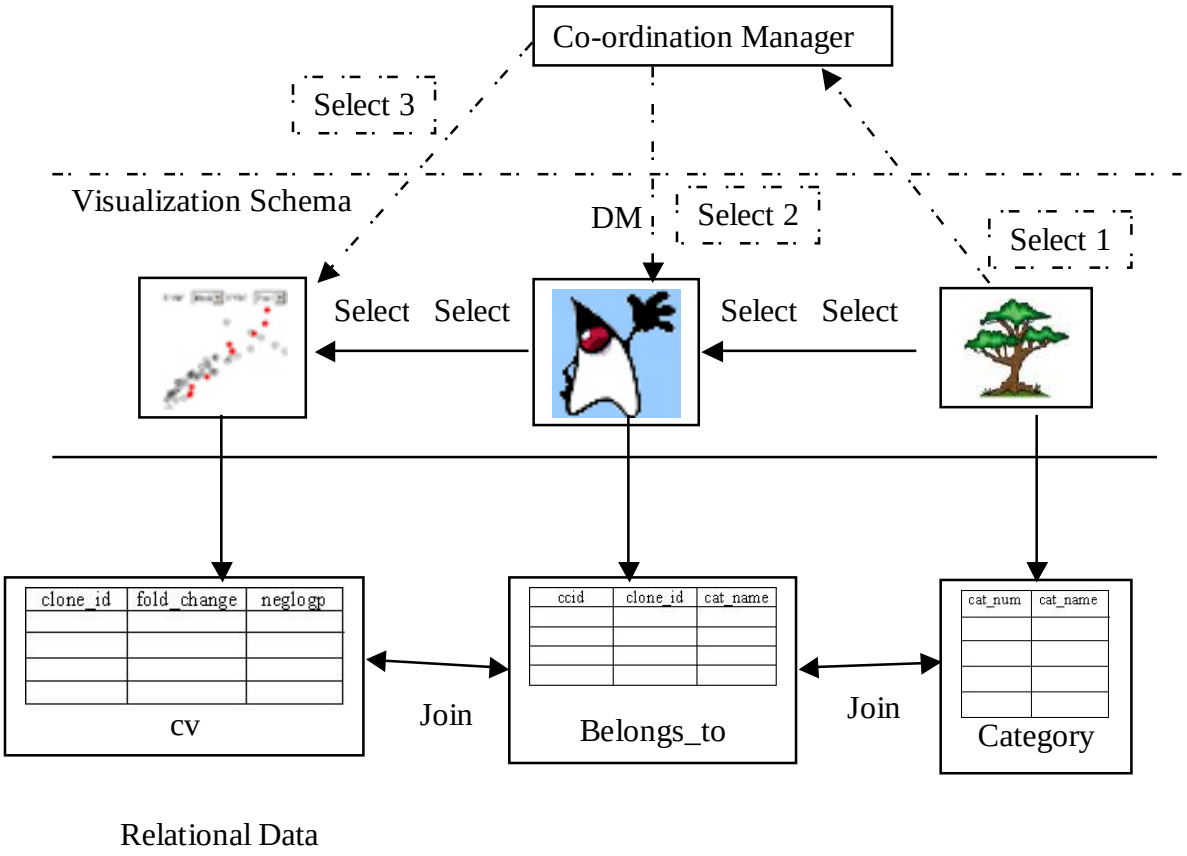

Figure 49: Underlying Event Firing and Key Conversion in Interactive Data Exploration. The numbering after the Selects is used to show the sequence of the events, the lowest number being the earliest.

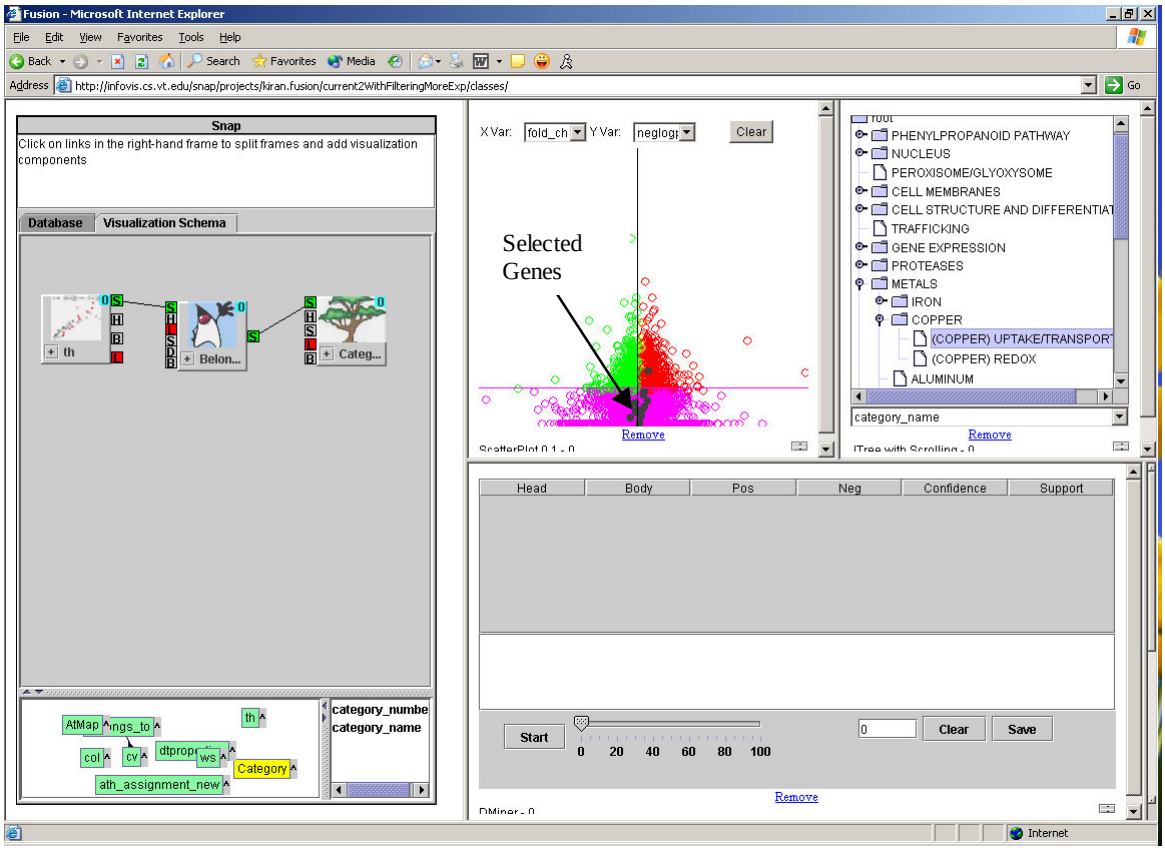

Figure 50: Interactive data exploration

5.3.2 Bias Specification, Batch Data Mining and Rule Visualization

The user adjusts the bias (threshold) by dragging the parallel line along the y-axis of the scatter plot as shown in Figure 51. The scatter plot fires a Head event(with the bias information). The Co-ordination Manager finds that it is a Bias event, by passes the normal key conversion and sends the Head event to the Data Miner as shown in Figure 52. The user starts the batch data mining by clicking the “Start” button on the Data Miner. The user is notified of the reception of the results by a dialog box and the rules are listed in a table. The user selects a rule of interest. The Data Miner fires a Head event (with the information of the genes that satisfy and those that do not satisfy the rule) . The Data Miner then fires the Body event (with the information on the categories that comprise the selected rule). The Coordination Manager directs the Head event (again by passing the normal key conversion) to the scatter plot. The scatterplot receives the event

and highlights the genes that satisfy the rule in dark, and those that do-not in gray. Similarly, the Coordination Manager directs the Body event to the Tree component. The Tree component then highlights the corresponding categories in the hierarchical category tree.

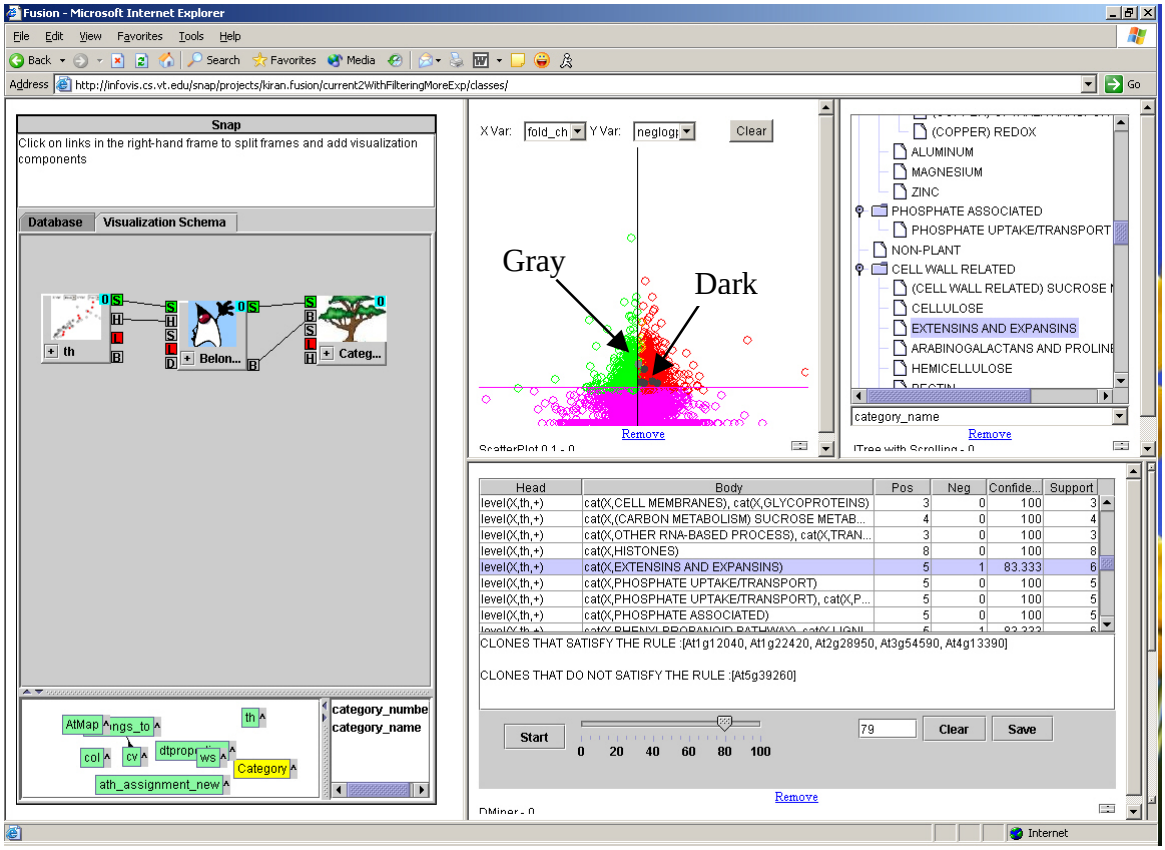

Figure 51: Bias Specification, Batch Data Mining and Rule Visualization

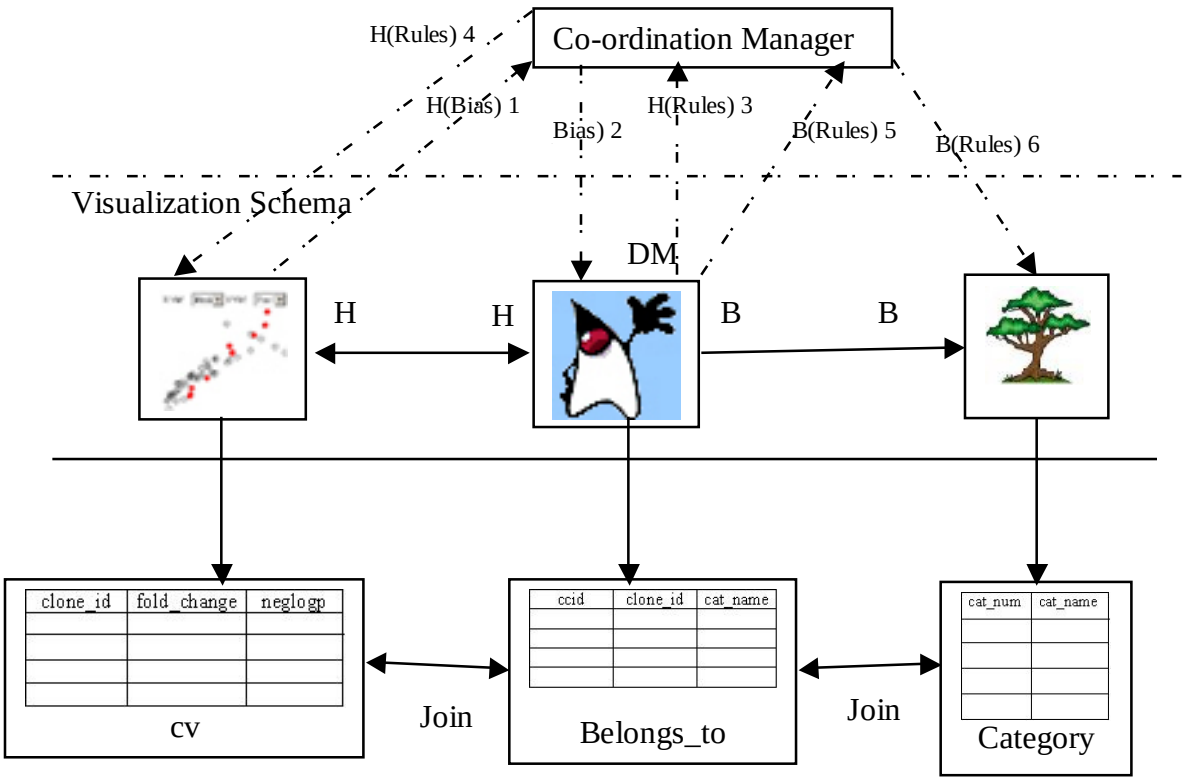

Figure 52: Underlying Event Firing and bias communication in bias specification and rule visualization.

Fusion has the following visualization design flexibility features:

- User can choose a visualization component based on its suitability for the data. For example, the user can choose a scatter plot for visualizing gene-expression data, and a hierarchical tree component for visualizing functional categories.
- User can dynamically establish coordinations between visualizations by drawing an edge from a port on one component, to a part on the other component. For example, in Figure, the biologist joins the “Select” ports for interactive data

exploration. Later the biologist decides to find ILP rules, so she connects the data mining ports, enabling her to switch smoothly between visualization and batch data mining. This smooth switching works in the following manner:

- When the biologist does a 'Selection' event in one of the components, then Fusion operates in a ''interactive visualization mode' and the biologist can see the corresponding translated event (selection/load/highlight) in the other components.
- When the biologist clicks on the 'Start' button on the Data Miner component, Fusion operates in the 'batch data mining' mode. Fusion sends the inputs to the Proteus, waits for the results and displays them in the Data Miner, the results are received.

- Users can edit the visualization schema to control the inputs to the data miner. For example, in Figure 51, the user directs the data mining algorithm to operate on table *cv*, by connecting the data mining port (H) on Scatterplot (whose underlying table is *cv*) to the Data Miner.

Thus, the Fusion architecture supports a flexible visualization framework for interactive data exploration, bias specification, ILP rule mining and rule visualization.

Chapter 6

6 Scenario and Analytic Evaluation

This scenario describes how a biologist uses Fusion to relate gene-expression patterns to functional categories and derive biological interpretations from such relationships. This scenario also shows how the user can switch between the interactive data exploration and batch data mining modes.

The biologist is analyzing the gene expression data from three different varieties of Arabidopsis thaliana; Cape Verde (cv), Columbia (col) Wassilewskija (ws) and Thellungiella (th), a relative of Arabidopsis thaliana. The goal is to understand the mechanisms underlying the acclimation to ozone stress, by studying the gene expression patterns. Figure 53 shows the underlying data base schema.

She loads the *fold_change* vs *neglop* data from *cv* into a scatterplot. She loads the *fold_change* vs *neglop* data from *th* into another scatterplot. She loads the functional category data into a tree view. She loads the data miner with the data table that maps genes to their functional categories.

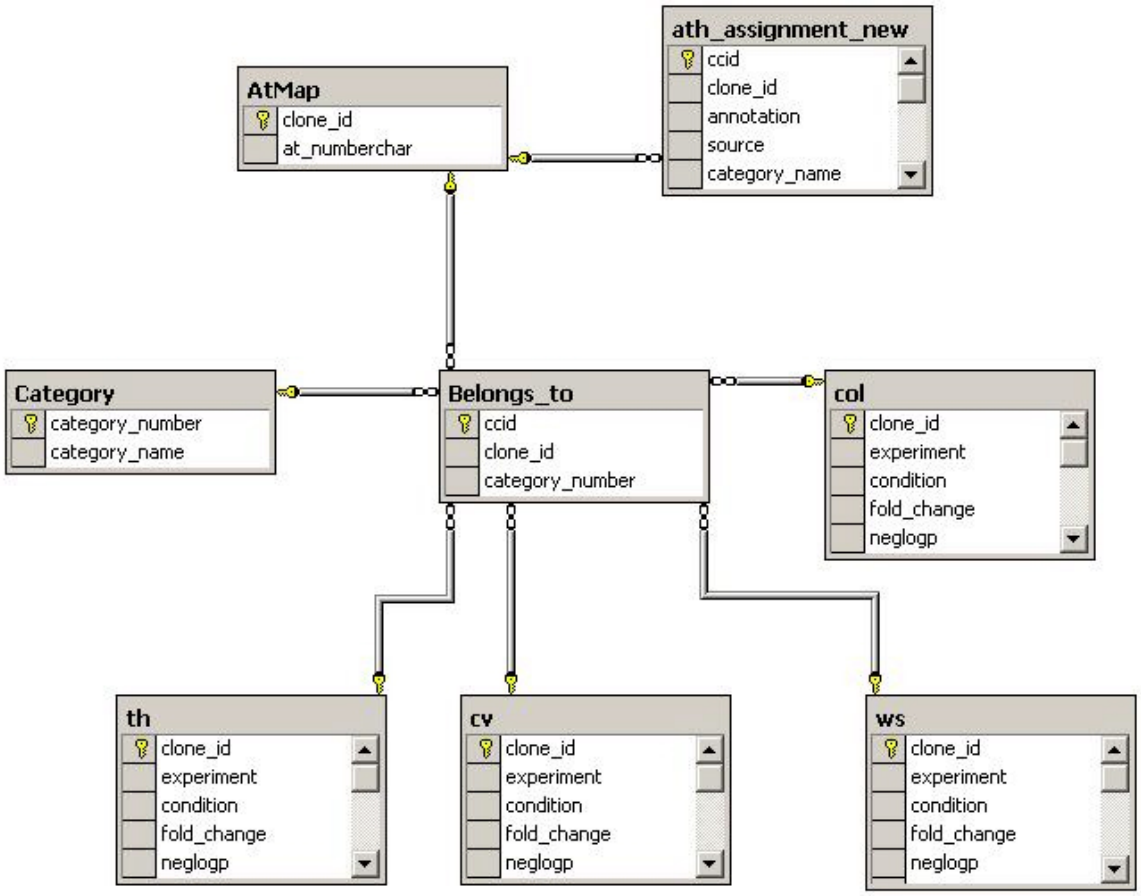

Figure 53: Microarray database schema

6.1 Interactive Data Exploration

The biologist is curious to compare the general behavior of genes in experiments *th* and *cv*. So she links the 'Select' ports on the scatterplot, data mining and tree components in the visualization schema.

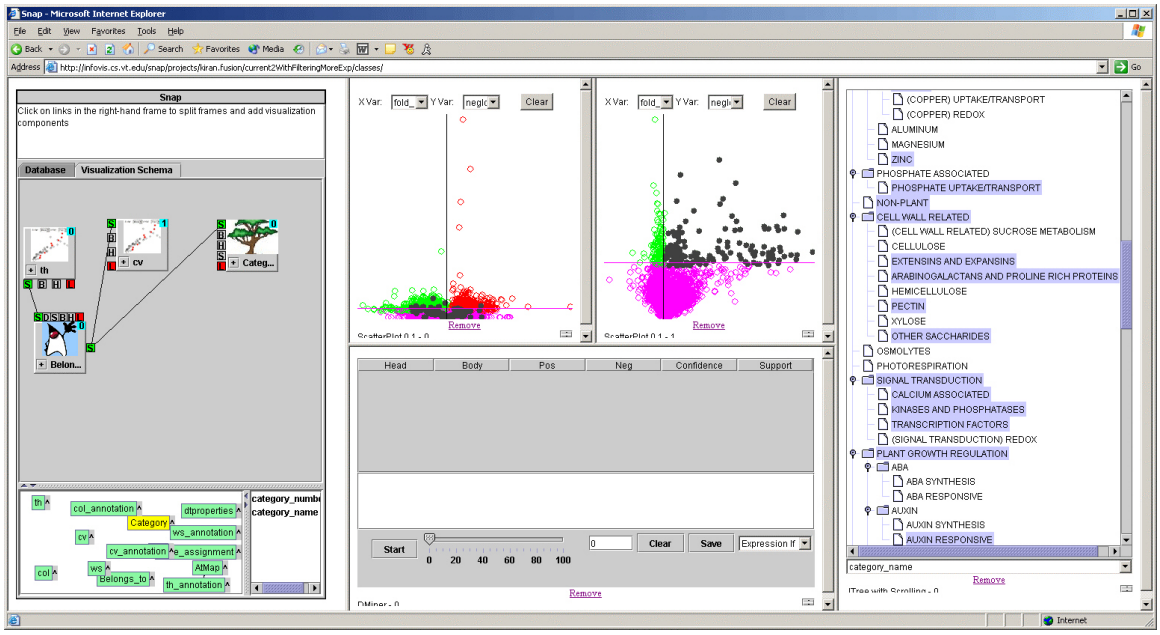

Figure 54: Interactive data exploration by normal brushing and linking

She then selects the expressed genes in the *th* scatterplot and she finds that most of the expressed genes in *th* were not expressed in *cv*. She then selects the expressed genes in *cv* and finds that most of the expressed genes in *cv* were not expressed in *th*. She observes the markedly different behavior of the genes of the two varieties. This confirms her prior biological knowledge about the difference in resistance of the two varieties to stress; cape verdae (cv) being the more resistant variety.

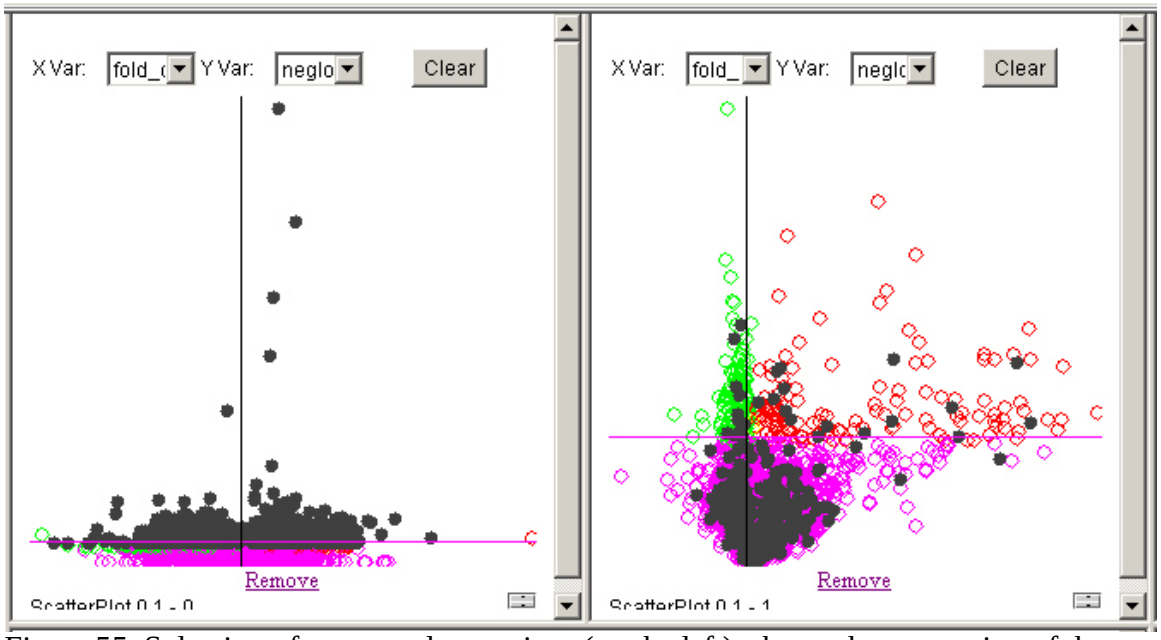

Figure 55: Selection of expressed genes in *cv*(on the left) shows the expression of the corresponding genes in *th*

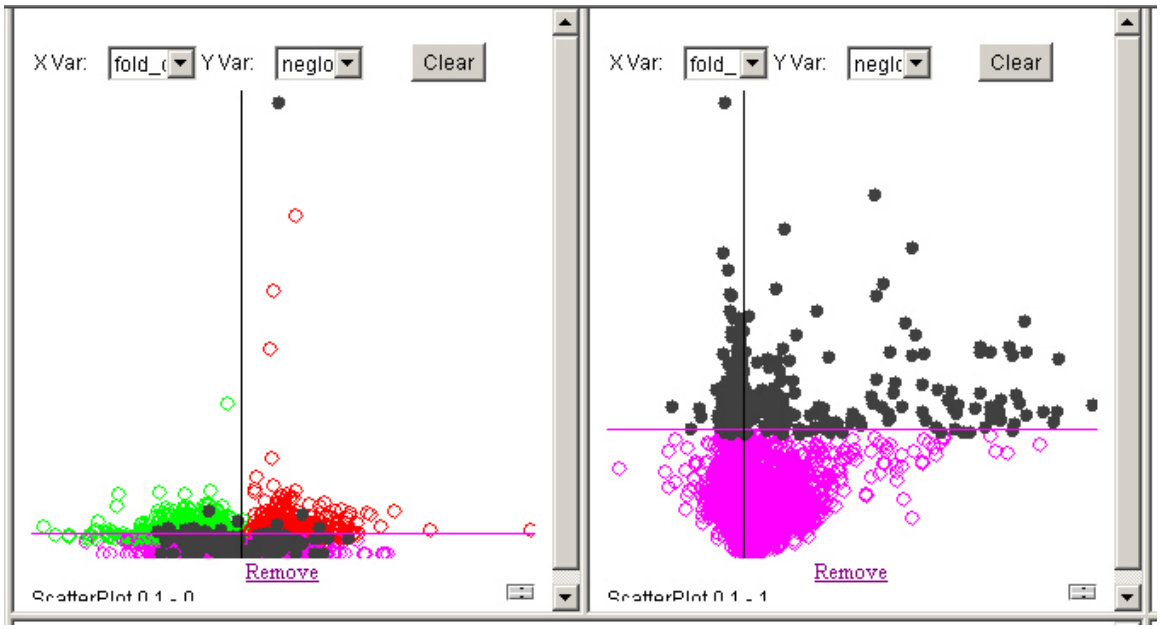

Figure 56: Selection of expressed genes in *th*(on the right) shows the expression of the corresponding genes in *th*

6.2 Data Exploration guided by Data Mining

The biologist sets up the visualization schema and starts the Data Mining process by clicking the 'Start' button on the Data Miner (Figure 57), the Data Miner sends the input data tables and rule schema to the Proteus server, and waits for the results from the server. The results received are displayed in the tabular component within the Data Miner.

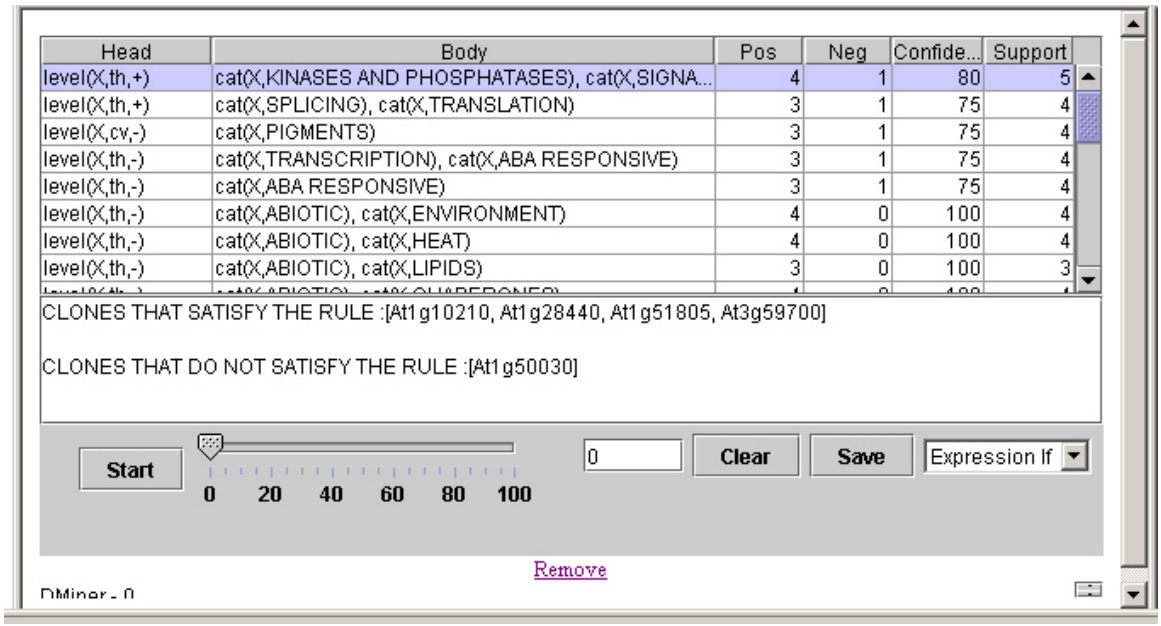

Figure 57: Data Miner Component

She notices the rule

level(X, th, +) :- cat(X, KINASES AND PHOSPHATASES), cat(X, SIGNAL TRANSDUCTION).

She notices that four genes satisfy the rule, shown under the Pos column.

She notices that one gene does NOT satisfy the rule, shown under the Neg column. She infers that out of the five significantly expressed genes belonging to the categories KINASES AND PHOSPHATASES and SIGNAL TRANSDUCTION, four of them were

also positively expressed (on the right side of the scatter plot) in the variety *th* and one of them is not. She then selects the rule to see the candidate genes in scatter plots(Figure 58) and their categories in the hierarchical tree structure . She can check the validity of the rule, as she can see the genes that satisfy the rule(shown in black) being on the right side of the scatterplot in variety *th* and one gene shown on the left side of the scatterplot in light grey.

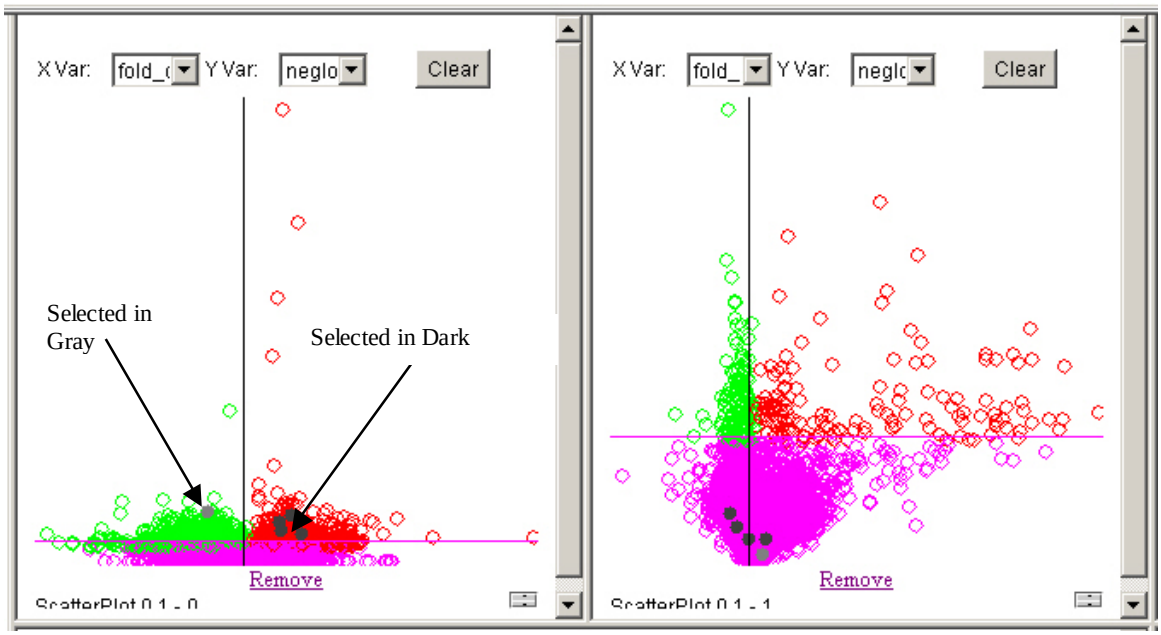

Figure 58: Scatter plots of varieties *th* (left) and *cv* (right) showing the genes that satisfy the rule (shown in black) and the genes that do not satisfy the rule shown in grey.

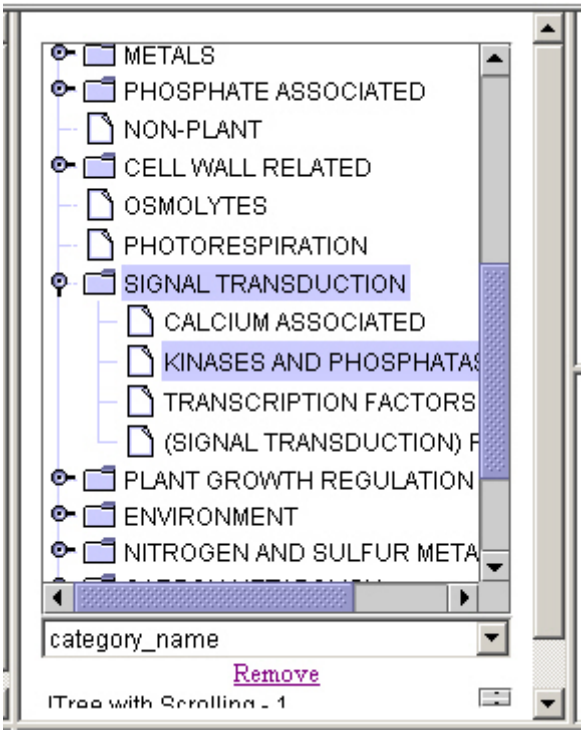

Figure 59: Hierarchical tree structure showing the constituent categories

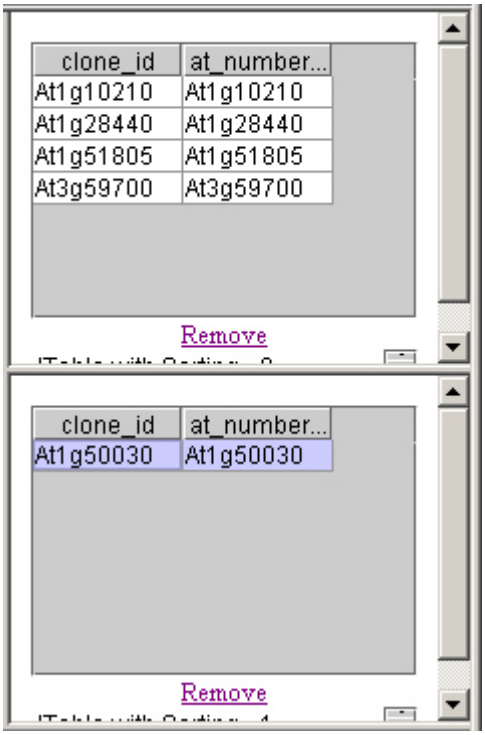

Figure 60: Table components, the top one showing the names of the genes that satisfy the selected rule, the bottom table showing the genes that do not satisfy the selected rule

She finds the names of the genes in the table components Figure 60. She is curious to find more about the gene that did not satisfy the rule, that is, the gene that belonged to both the categories KINASES AND PHOSPHATASES and SIGNAL TRANSDUCTION but was not positively expressed in variety *th*. So she selects the gene(At1g50030) in the bottom table. She then finds the results of the public database search on that gene in the public database component (Figure 61). So she hypothesizes that the gene At1g50030 is different form the other genes that belong to KINASES AND PHOSPHATASES and plans to conduct further analyses to find more about its behavior. Figure 62 summarizes the entire process of data exploration guided by Data mining results.

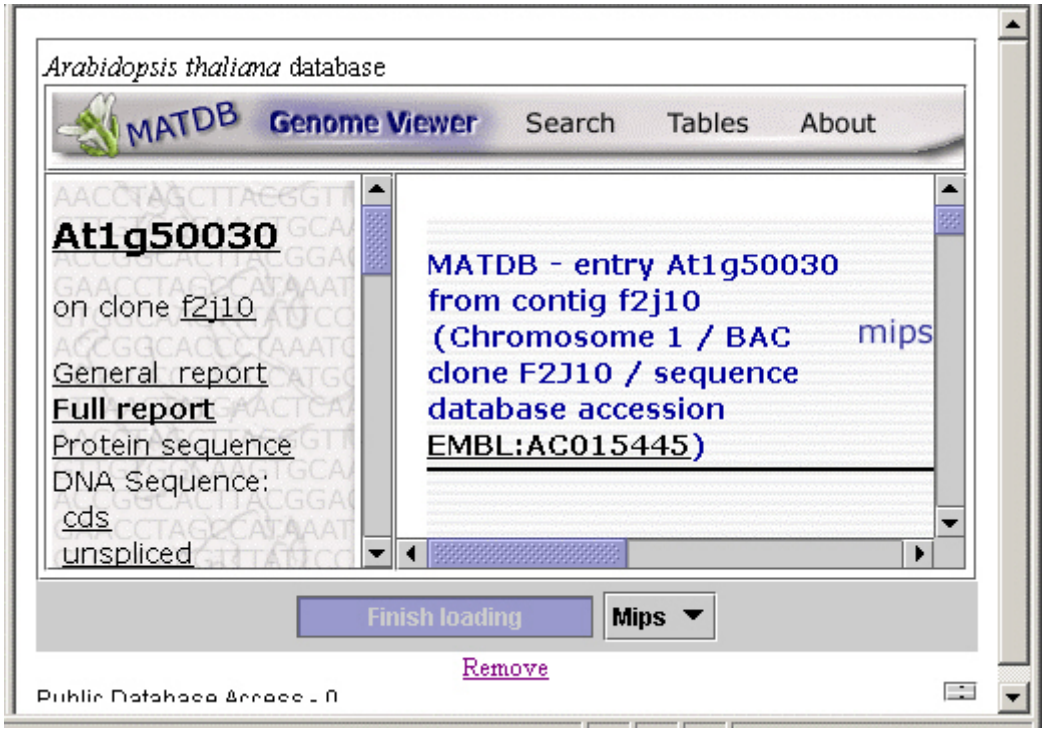

Figure 61 : Public database component that shows the results of the public database search on the selected gene

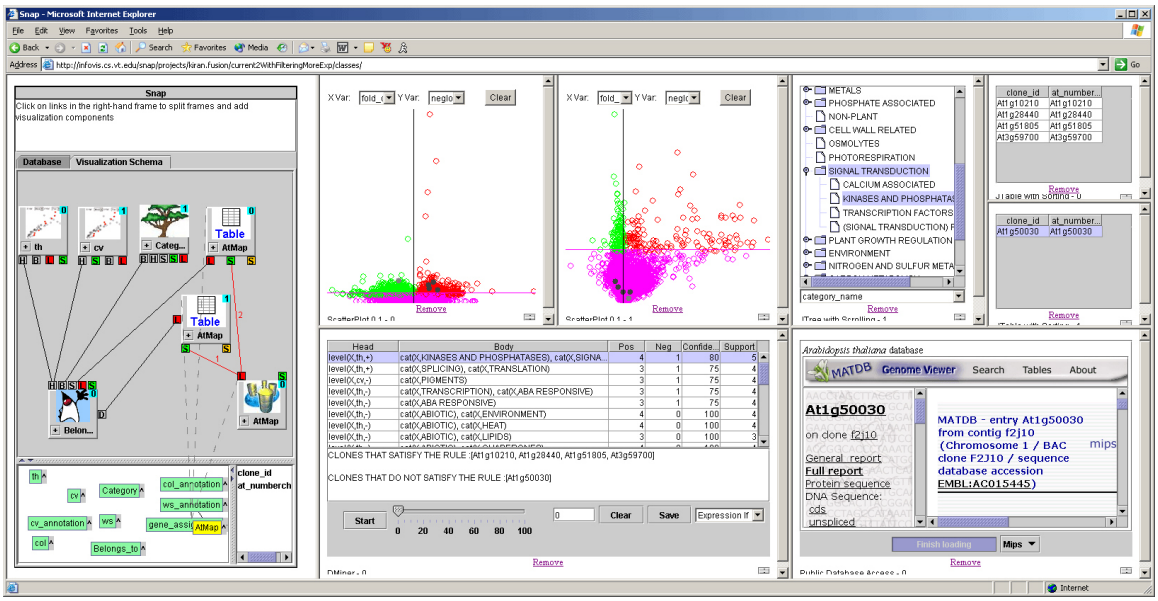

Figure 62 : Data Exploration guided by ILP Rules

6.3 Analytic Evaluation

Figure 64 shows the summary diagram of the positively responding gene categories in each plant variety. The rules generated by Proteus (using Fusion, in batch mode) , were used to compose the diagram. For example, rules shown below lead to the part of the diagram as shown in Figure 63.

Level(X,CV, +) :- cat(X,TranscriptionFactors)

Level(X,CV, +) :- cat(X, Nitrogen and Sulphur Metabolism), cat(X, Amino acid Metabolism)

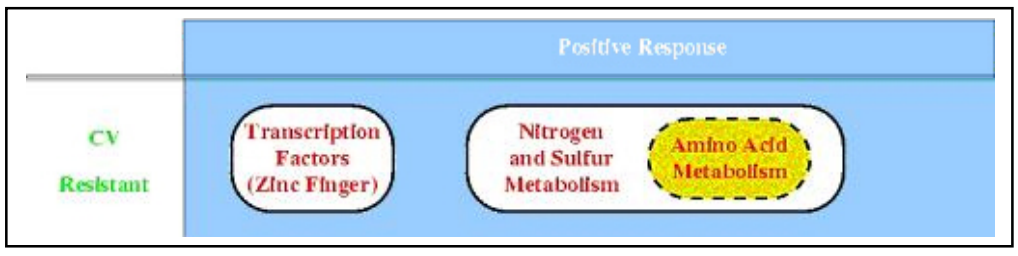

Figure 63: Part of the summary diagram showing the *cape verde* gene categories positively responding to ozone stress.

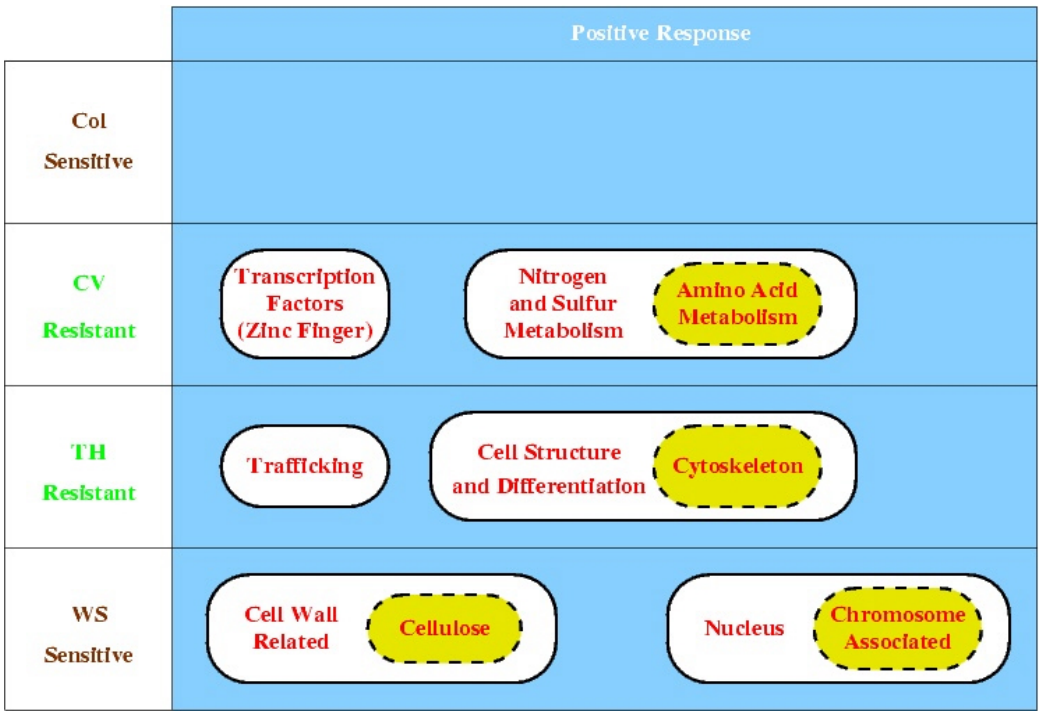

Figure 64: Summary diagram of the results found by the analysis of ozone-stress microarray data, using Fusion. Diagram courtesy: Dr.Ruth Grene. Data provided by Dr.Bohnert and Pinghua Li of the University of Illinois.

Figure 65 shows the list of genes that satisfy the rule

Level(X,CV, +) :- cat(X,TranscriptionFactors)

Gene	Annotation	Comments
At1g72050	C2H2-type zinc finger Homologous to TF IIIA	DNA repair (animal protein)
At2g18380	GATA zinc finger protein	Nitrogen regulatory protein (fungi)
At2g45050	GATA zinc finger protein	Four cysteines coordinate a zinc ion.
At3g262250	CHP-rich zinc finger protein	DC1 domain rich in cysteines and histidines

Figure 65: Transcription Factors (Zinc Finger) proteins associated with acclimation to Ozone (Upregulated in Cape Verde) (Courtesy: Dr.Ruth Grene)

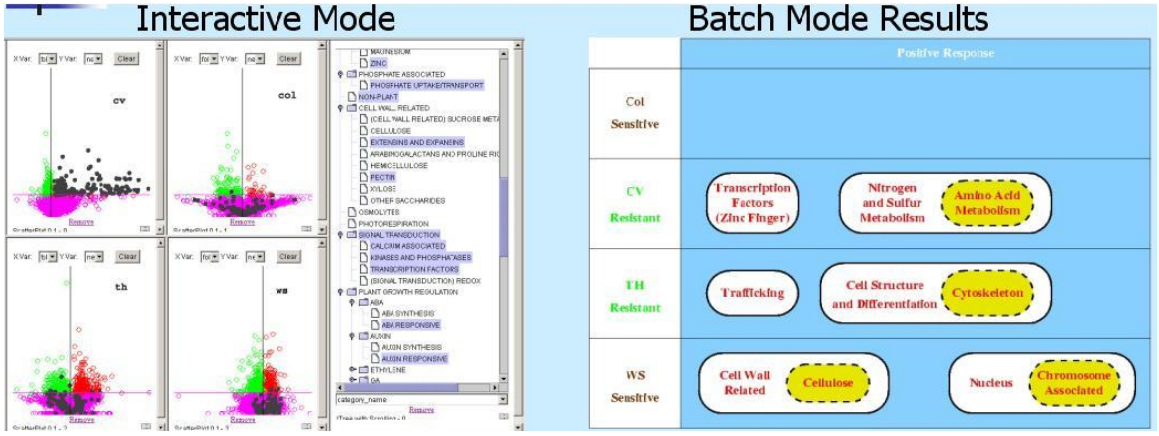

Figure 66: Typical results of the interactive and batch modes of Fusion

Interactive data exploration in Fusion offers the following features:

- Generates testable hypotheses. For example, Figure 69, illustrates a scenario where the user finds a gene that is positively expressed in both *cv* and *th*. This leads to a hypothesis that this gene might be responsible for stress resistance.
- Validates Existing Knowledge and gives good overview of the overall behavior. For example, in Figure 66, the user compares gene expression values across multiple varieties, by loading each in a scatter plot. The user selects the positively expressed genes in *cv*, and finds they do not exhibit similar behavior in any of the other varieties. This also validates the existing knowledge about the high stress resistance of *cv*, compared to the other varieties.
- Facilitates rule visualization and allows the users to filter rules based on their confidence measures, as shown in Figure 67. It also allows users to investigate rules of interest.

Batch Mode of usage of Fusion helps the user in summarizing the biological mechanisms as shown in Figure 64 and also in making specific observations as shown in Figure 65.

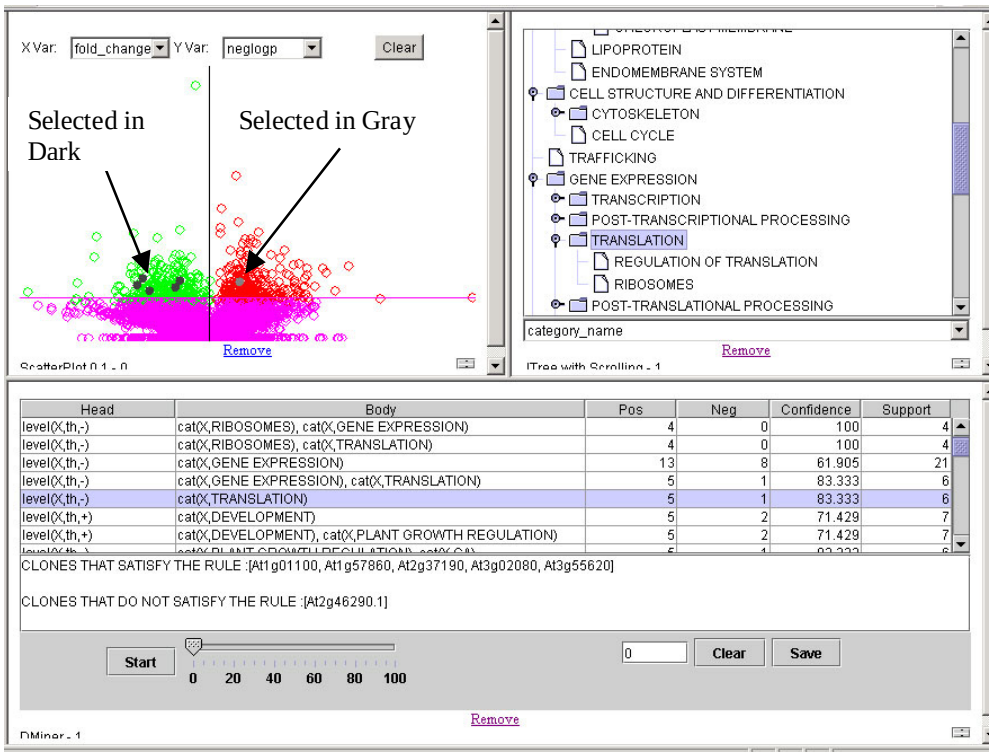

Figure 67: Rule Visualization

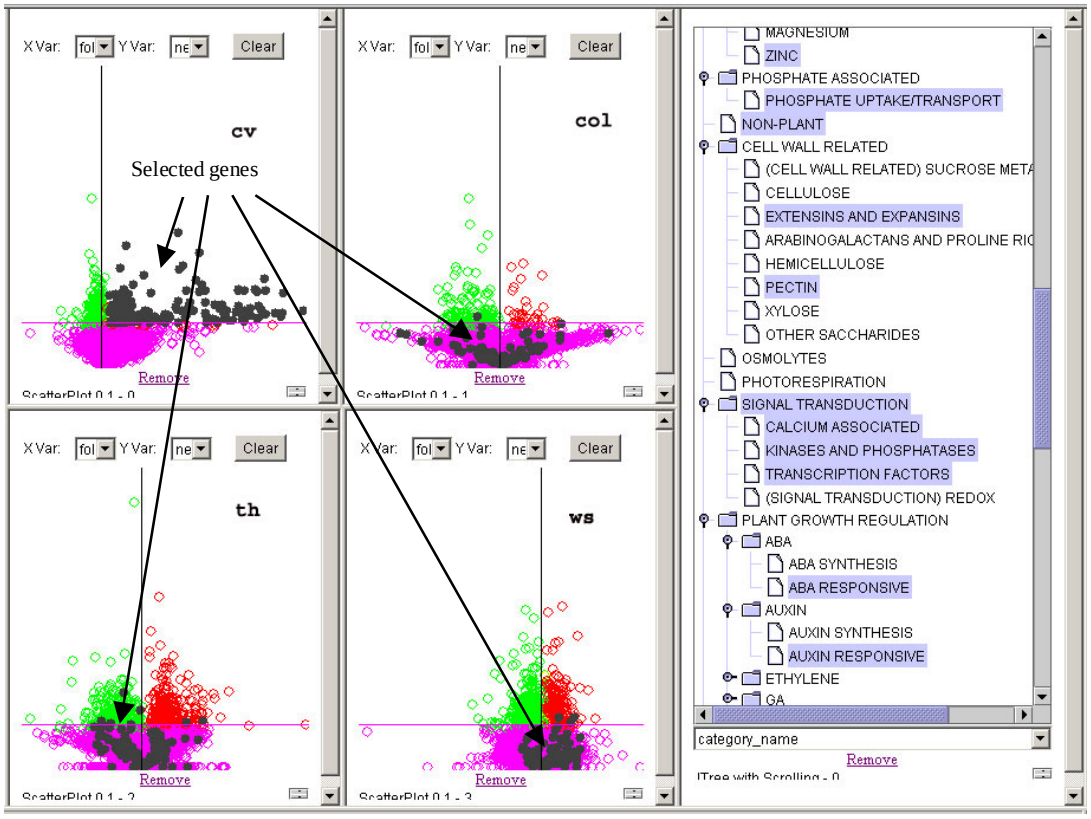

Figure 68: Overall behavior of genes that are positively expressed in *cv*

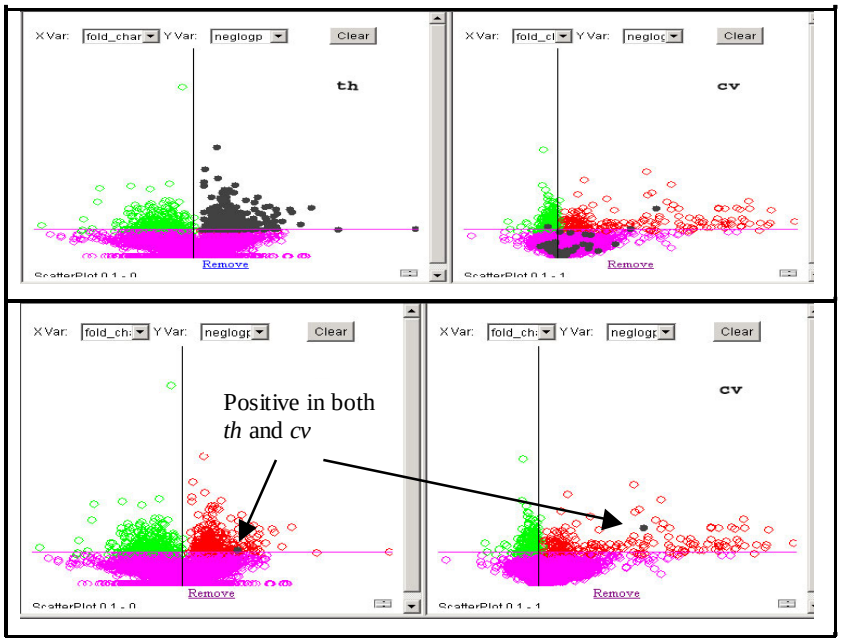

Figure 69: Identifying specialized subsets interactively, as in this example, a gene that is positively expressed in both *cv* and *th* generates hypotheses for further investigation.

Thus, Fusion helps biologists perform interactive data exploration, batch data mining and rule visualization. Fusion's capability to smoothly switch between the interactive and batch modes allows the user to optimize the discovery process by taking advantage of the strengths of both visualization and data mining.

Chapter 7

7 Conclusions

The goal of this thesis is to develop a theory, user interface and architecture to provide the user with the ability to perform both interactive data exploration and data mining to optimize the knowledge discovery process.

7.1 Contributions

7.1.1 Theory

The underlying theory of the Fusion system is based on the analogy between visualization and data mining concepts. The data mining concepts like descriptor, bias, rule and evaluation measure were associated with corresponding visualization concepts. The advantage of visual data exploration is that the user is directly involved in the analysis process by visualization and immediate feedback. Multiple view visualizations provide the user with the flexibility to choose different views for different types of data. Data mining algorithms find complex patterns in data. Fusion facilitates synergy between the interactivity and usability of visualization process with the pattern-finding abilities of ILP mining algorithms. Fusion helps the user in striking a balance between generalization and specialization by supporting a variety of bias choices, algorithms and evaluation criteria. Fusion enables an interactive discovery feedback loop where the users can adjust their biases and re-run the data mining algorithm, based on the quality and the number of the results that were found in the previous search.

7.1.2 Architecture

Fusion incorporates a software architecture that supports the integration of interactive data exploration and batch ILP rule mining with possible extensibility over descriptors

(bias selection operators), views (visualizations), data mining algorithms and evaluation criteria.

7.1.3 User interface

Fusion user interface supports a two-mode exploratory process; the interactive mode where the user gets a feel of the data and chooses the biases; the batch mode in which the data mining algorithm is run, followed by the visualization of discovered rules and associated confidence measures in the data mining component. Fusion also helps the user validate the discovered rules, visually by showing the user their constituent data and attributes. Fusion facilitates an improvement in the quality of the results by better choice of the biases based on the feedback. The architecture and user interface of the current Fusion system supports the analogy between fundamental concepts in visualization and ILP rule mining.

7.2 Future Work

Our future work on Fusion is aimed at further generalizing the user interface and architecture to realize the full theory of the analogy between visualization and data mining.

Views: Fusion can be extended further to support a wide variety of views. Fusion's extensible architecture makes it easier to add new views to the system. Existing views like parallel co-ordinates plot, tree maps could be extended to support bias specification.

Biases: The data mining process can be further enriched by offering users a variety of bias selection operators. Thereby, we help the user in being more expressive in choosing the biases, like union, intersection etc. For example, user could specify biases like (if X belongs to Catgeory1) OR (if X belongs to Category2), ((if X belongs to Catgeory1) AND (if X belongs to Category2)) AND (if X belongs to Category3) etc.

Algorithms: Fusion can be extended to support a variety of data mining algorithms. This gives the user an option to choose an algorithm based on the needs. A variety of classification and association rule algorithms could be supported.

Rule Schema: Fusion can be improved to provide support for the specification and manipulation of the rule schema, by establishing and editing co-ordinations in the visualization schema.

Evaluation Measures: Fusion can be extended to support more evaluation measures. This gives the user feedback on the quality of the knowledge that was derived during the data mining process. This can help the user in fine-tuning the biases to get the results of the desired quality.

Rule Visualization: The Data Miner can be enhanced further by adding the capabilities to group together similar rules. Rules can be grouped together based on similarity of different attributes like functional categories, expression levels etc. The ILP Rule Visualization can be improved by presenting an overview of all the rules and color code the rules based on their attributes. This would enable the biologist to narrow down on the desired subset of the rules.

Usability: Usability studies can be conducted to improve the user experience at different stages of the usage of the Fusion system. This can help in better design of the views, descriptors, algorithms and evaluation measures.

With these enhancements, Fusion can evolve into a very useful tool for users in enhancing their knowledge discovery process.

Chapter 8

References

[1] Seo J. and Shneiderman B., Understanding Hierarchical Clustering Algorithms by Interactive Exploration of Dendrograms: *A Case Study with Genomic Microarray Data*. IEEE Computer Special Issue on Bioinformatics, November 2001.

[2] Berchtold S., Jagadish H., Ross K., Independence Diagrams: A Technique for Visual Data Mining. Proc. 4th Intl. Conf. on Knowledge Discovery and Data Mining, New York City, 1998, pp. 139-143.

[3] Keim D., Information Visualization and Visual Data Mining. IEEE Transactions on Visualization and Computer Graphics, Vol. 8, No. 1, January-March 2002.

[4] Kreuseler M. and Schumann H., A Flexible Approach for Visual Data Mining. IEEE Transactions on Visualization and Computer Graphics. Vol. 8, No. 1, January-March 2002.

[5] North, C., Conklin, N., and Saini, V., Visualization Schemas for Flexible Information Visualization, Proc. IEEE InfoVis 2002 Symposium, October 2002.

[6] Dysvik B., and Jonassen I., J-Express: Exploring gene expression data using Java. Bionformatics, Vol. 17, No. 4, 2001, pp. 369-370.

[7] Han J., Cercone N., RuleViZ: A Model for visualizing knowledge discovery process, ACM 2000, pp.244 - 253.

[8] Hochheiser H., Shneiderman B., Visual Queries for Finding Patterns in Time Series Data University of Maryland, Computer Science Dept. Tech Report #CS-TR-4365, UMIACS-TR-2002-45.

[9] B. Shneiderman, Inventing discovery tools: Combining information visualization with data mining, Information Visualization journal, 1(1), 2002.

[10] Gamma, E., Helm, R., Johnson, R., Vlissides, J., Design Patterns: Elements of Resuable Object-Oriented Software, Addison-Wesley, 1995.

[11] Sun Microsystems, Inc. Java 2 Platform, Standard Edition, v1.4.0API Specification [WWW document] http://java.sun.com/j2se/1.4/docs/api/ .

[12] Shaw M. and Garlan D., Software Architecture: Perspectives on an Emerging Discipline, Prentice Hall, Inc., 1996.

[13] Conklin, N., "A web-based, run-time extensible architecture for interactive visualization and exploration of diverse data", Virginia Polytechnic Institute and State University, Department of Computer Science, Master Thesis, December 2002.

[14] Lenwood S.Heath, Naren Ramakrishnan, Ronald R. Sederoff, Ross W. Whetten, Boris I. Chevone, Craig A. Struble, Vincent Y. Jounenne, Dawei Chen, Leonel Merwe avan Zyl, and Ruth Grene, "Studying the Functional Genomics of Stress Responses in Loblolly Pine with the Expresso Microarray Experiment Management System, *Comparative and Functional Genomics*, Vol. 3, No. 3, June 2002, pp. 226-243.

[15] Naren Ramakrishnan, A.Y. Grama, Data Mining: From Serendipity to Science, IEEE Computer, Vol. 32, No. 8, August 1999, pp. 34-37.

[16] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein, Cluster analysis and display of genome-wide expression patterns, Vol. 95, Issue 25, December 8, 1998, pp. 14863-14868.

[17] http://www.ornl.gov/sci/techresources/Human_Genome/publicat/primer2001/1.shtml

[18] http://industry.ebi.ac.uk/~alan/MicroArray/IntroMicroArrayTalk/

[19] http://www.genomicglossaries.com/content/RNA.asp

[20] Cummings, Craig A. Relman, David A, Using DNA Microarrays to Study Host-Microbe Interactions. Emerging Infectious Diseases; September 01, 2000.

[21] Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, et al., Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl. Acad. Sci. U S A 1999;96:6745-50.

[22] Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, et al., Interpreting patterns of gene expression with self-organizing maps: methods and

application to hematopoietic differentiation, Proceedings of the National Academy of Sciences, USA 1999;96:2907-12.

[23] Eisen MB, Spellman PT, Brown PO, Botstein D., Cluster analysis and display of genome-wide expression patterns, Proceedings of the National Academy of Sciences, USA, 1998,95:14863-8.

[24] Ben-Dor A, Shamir R, Yakhini Z., Clustering gene expression patterns, Journal of Computational Biology, 1999,6:281-97.

[25] Scherf U, Ross DT, Waltham M, Smith LH, Lee JK, Tanabe L, et al., A gene expression database for the molecular pharmacology of cancer, Nature Genetics, 2000,24:236-244.

[26] Ross DT, Scherf U, Eisen MB, Perou CM, Rees C, Spellman P, et al., Systematic variation in gene expression patterns in human cancer cell lines, Nature Genetics, 2000,24:227-35.

[27] Khan J, Simon R, Bittner M, Chen Y, Leighton SB, Pohida T, et al., Gene expression profiling of alveolar rhabdomyosarcoma with cDNA microarrays. Cancer Research, 1998; 58:5009-13.

[28] Perou CM, Jeffrey SS, van de Rijn M, Rees CA, Eisen MB, Ross DT, et al. Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. Proc Natl Acad Sci U S A 1999;96:9212-7.

[29] Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 2000; 403:503-11.

[30] Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 1999;286:531-7.

[31] Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proceedings of the National Academy of Sciences, USA, 1999;96:6745-50.

[32] Wang K, Gan L, Jeffery E, Gayle M, Gown AM, Skelly M, *et al.* Monitoring gene expression profile changes in ovarian carcinomas using cDNA microarray. Gene 1999; 229:101-8.

[33] Brown MP, Grundy WN, Lin D, Cristianini N, Sugnet CW, Furey TS, *et al.* Knowledge-based analysis of microarray gene expression data by using support vector machines, Proceedings of the National Academy of Sciences, USA, 2000;97:262-7.

[34] Gaasterland T. and Bekiranov S., Making the most of microarray data. Nat Genet 2000; 24:204-6.

[35] Schena M, Shalon D, Davis RW, Brown PO., Quantitative monitoring of gene expression patterns with a complementary DNA microarray, Science, 1995;270:467-70.

[36] Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, Eisen MB, *et al.* Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol Biol Cell 1998;9:3273-97.

[37] Cho RJ., Campbell MJ., Winzeler EA., Steinmetz L., Conway A., Wodicka L., *et al.* A genome-wide transcriptional analysis of the mitotic cell cycle. Mol Cell 1998;2:65-73.

[38] Chu S, DeRisi J, Eisen M, Mulholland J, Botstein D, Brown PO, *et al.* The transcriptional program of sporulation in budding yeast. Science 1998;282:699-705.

[39] DeRisi JL, Iyer VR, Brown PO. Exploring the metabolic and genetic control of gene expression on a genomic scale. Science 1997;278:680-6.

[40] Tao H, Bausch C, Richmond C, Blattner FR, Conway T., Functional genomics: expression analysis of Escherichia coli growing on minimal and rich media, Journal of Bacteriology, 1999;181:6425-40.

[41] Richmond CS, Glasner JD, Mau R, Jin H, Blattner FR. Genome-wide expression profiling in Escherichia coli K-12. Nucleic Acids Research, 1999;27:3821-35.

[42] Marton MJ, DeRisi JL, Bennett HA, Iyer VR, Meyer MR, Roberts CJ, et al. Drug target validation and identification of secondary drug target effects using DNA microarrays, Nature Medicine, 1998;4:1293-301.

[43] Cho, R. J., Campbell, J. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockart, D. J., *et al.*, 1998, Molecular *Cell* 2, 65-73.

[44] Nameeta Shah, D.A, St.Clair, C. Dodsworth, Bernd Hamann, Kenneth I. Joy, GeneBox: Visualizing Gene Expession Data Resulting from Microarray Experiments, Proceedings of the 2002 UC Davis Student Workshop on Computing, TR CSE-2002-28.

[45] Daniel B. Carr, Roland Somogyi, George Michaels., Templates for Looking at Gene Expression Clustering, Statistical Computing & Statistical Graphics Newsletter, April 1997.

[46] Purvi Saraiya, Chris North, and Karen Duca., An Evaluation of Microarray Visualization Tools for Biological Insight, The IEEE Symposium on Information Visualization, October 2004.

[47] Ahlberg C. and Wistrand E., IVEE: An information visualization and exploration environment, Proceedings of the International Symposium on Information Visualization, Atlanta, GA, 1995, pp. 66–73.

[48] Jonathan I. Watkinson, Allan A. Sioson, Cecilia Vasquez-Robinet, Maulik Shukla, Deept Kumar, Margaret Ellis, Lenwood S. Heath, Naren Ramakrishnan, Boris Chevone, Layne T. Watson, Leonel van Zyl, Ulrika Egertsdotter, Ronald R. Sederoff, and Ruth Grene, Photosynthetic Acclimation Is Reflected in Specific Patterns of Gene Expression in Drought-Stressed Loblolly Pine, Plant Physiology, December 2003, Vol. 133, pp. 1702–1716.