**Abstract**

In the design of FilooT, some of the generated views are manipulatable. In addition to

the most common techniques for manipulating the views, which are scrolling and zooming,

There are two other techniques that I discuss in this section: the Overview+Detail, and the Focus+Context [24].

**1. Introduction**

Health investigators are interested in finding the relation of the gene substitutions to disease characteristics in a virus gene family of sequences. Aanalysts also need to understand how characteristics of the disease related to virus strain mutations.

Applying the principles of Information Visualization to design biology-specific systems is highly acknowledged in the literature [18, 25, 26, 30, 32]. Advanced hand-drawn pictures in scientific publications prior to existing computers shows that utilizing the human vision system was grounded in biology many years ago [32]. As the biological data-sets scales are increasing rapidly, custom software combined with manual intervention is replacing manual data analysis in biological sciences [18].

These computer-based visualization tools have enhanced our ability to communicate with the large amount of scientific data. Usually these tools are designed for a specific data-set/task-set in the domain. Advantages of these custom tools are twofold. First, they solve target analysts’ problems, which are part of the domain problems. Second, by analyzing the successful tools, researchers can eventually extract the target domain’s design guidelines and patterns. A special issue of Nature Methods gives biologists an overview of current computational methods and tools used for visualizing biological data [32]. Although classical visualization techniques are used in the field of biology, researchers define new and creative ways to meet the target domain visualization needs [15]. One such example is the work of Nielsen et al. in creating a novel graph representation for visualizing genome sequence assembly structures [30].

**2. Literature review**

Once a virus infects a host, it makes copies of itself, growing the population of virus within the same host and eventually spreading to other people. During the viral replication process, its gene sequence has to copy and transmit the exact same sequence (between 7000 to 500,000 nucleotide bases) to its child cells. During this process, typically some mistakes can be made and as a result, some substitutions appear in genetic sequence [31].

One way for characterizing DNA is to compare their sequences with each other [19].

In bioinformatics, aligning the sequences in rows helps finding the similar regions between them. In order to compare more than a pair of sequences, all sequences must be compared to each other in a heuristic optimization process called Multiple Sequence Alignment [19].

There are generally two types of visualizations for Multiple Sequence Alignment. Multiple Sequence Alignment viewers [4, 5] and the Sequence Logo [36]. Multiple Sequence Alignment view is a table representation, in which each row corresponds to a sequence and each column is a position in all the sequences. In the case of DNA sequences, each cell represent a DNA letter in each sequence.

An important characterization of this view is to show the variations in the sequences to the analyst.

IMAS [39] is a visual analysis tool for performing rapid analyses of DNA sequences.

This tool visualizes the output of common bioinformatics tools such as BLAST program for Pairwise Alignment and Clustal-W for Multi-alignment in a unified framework. The semantic zooming navigation in IMAS [39] provides this kind of features to the users.

Jalview is one of the most commonly installed tools The Multiple Alignment view is capable of handling the sequence hiding (row hiding)[48]. Another feature is that the user can select a number of sequences and make a group out of them. The group name will be shown when the user places the mouse over the sequences. The group making feature is supported only for sequences (rows) and not for positions (columns).

The sorting feature allows the user to sort the sequences with a number of different criteria, such as labels of groups. Another useful feature of Jalview is that the user can change colour choices.

Sequnce-Juxtaposer [42] is an example of applying Focus+Context in bioinformatic sequences alignment explorations.

Historically, the table view provides interactive features to allow the users to gain insight about the data [39, 48].

The following sections cover a number of these tools that primarily influenced my design.

There are also innovative tools for multiple alignment visualization. Nene et al. [29] built an innovative tool that helped them identifying mutations that worsen disease characteristics.

The multiple alignment views usually accompanied by another metadata matrix view are a kind of Table Lens where each column contains information about one metadata and each row represents the value of that metadata for each strain [33, 45, 11, 43, 23]. Freire et al. [11] used horizontal bars (the Length visual channel) for encoding each cells’data, as well as vertical bars on top of each column to show overall column distributions.

Others, such as Sopan et al. [43], used colour saturations in different cells to encode their values. The colour saturation usually is a better choice for visualizing the information in a cell as it has higher accuracy for encoding ordered data [23].

Some data has a hierarchical structure. One common visualization technique for representing this structure in the data is tree (or a special node-link diagram) visualization [5, 15].

The Noblis Team [4] used sunburst layout to represent the evolutionary tree of the current outbreak sequences. The tree visualization from Noblis team submission for the VAST challenge 2010, mini challenge 3. The diagram uses the sunburst layout to show the evolutionary relationship between the sequences. The colours represent the degree of the overall danger level of the sequences.

In another example, Freire et al. [11] used the basic Node-Link layout for the evolutionary tree information.

In FilooT, I adopted the same representation.

In some cases there are other types of relationships between data items, which could be considered a general form of hierarchal relationship the Network representations is used [15].

ManyNets [11], a network visualization tool with tabular interface.

The tabular view was a kind of Table Lens that the disease characteristics were shown in columns. There was a feature of creating a new column with the existing characteristics.

Clicking on a column header sorted all the rows according to the values of the particular characteristics associated to the column.

GeneTracer [22] It had three views: Gene Sequence view, Disease Characteristic view, and Graph view. In Disease Characteristic view, they had a Table Lens where each column had a different colour and each cell had different saturations of that colour. The Graph View visualized the relations among the sequences via a Minimum Spanning Tee representation where the weight of an edge between two sequences is the Hamming distance between the two [19].

Wood et al. [50] Their Table Lens representation utilized a heat map. An interesting feature int their Table Lens was two levels of sorting of the rows (Sorting the rows according to one characteristic, then sorting them again within each category of the first sort, according to the second characteristic).

**3. design**  
Figure 1 FilooT visualization system. (a) An interactive visualization table to represent  
the genetic sequence information. (b) A matrix visualization for interacting with the  
disease characteristics data. (c) The P-Value bars to show a metric (reverse of pvalue in  
Mann- Whitney U test 3.2.4.3 ) about each column. (d) The Group View containing the  
user created groups along with an overview of each group. (e) A graph visualization for  
representing row (or column) relationships depending on the system mode (Row based or  
Column based). (f) Two buttons enable the user to choose between the Column and Row mode. (g) The Statusbar is being updated after each action that the user makes.

**3.1. Interactive Tabular view**

The tabular view is an interactive visualization for exploring genetic sequences. The first row represents the genetic information about the original sequence. The second row shows position numbers and numbers start from one and end with the length of the sequences.

Each of the subsequent rows indicates one sequence. Each cell contains the result of the comparison of each sequence with the original sequence appeared in the first row. The purple color is used to represent those cells that did not change in comparison with the original sequence and the yellow color highlights cells with a change in a particular row and column. The letter indicates a change in the information of the specific cell in comparison to the corresponding column in the original sequence.

This view supports the following user interactions. These interactions affect the other views linked to Main View.

**Navigation**: The horizontal and vertical scroll bars at the bottom and on the right are so that the user can explore more of the sequences and the positions’ data

**Zoom:** The “+” and “-” buttons allow user to zoom in and out.

**Reset:** In order to compare different columns with each other, placing the columns close to each other frees up the cognitive load of the users and enables them to use their memory to focus on their desired task [40]. One way of putting columns close to each other is to allow the user to drag and drop the columns next to each other. However, enabling this feature admits that the user can change the natural order of nucleotides in a sequence. One must realize that the natural order is meaningful in the original domain. In order to keep the natural position orders, the “reset” button returns the columns to their original sequence from one to the length. This feature is used whenever the user previously changed the column positions, and wants to reset the position numbers.

**Filter:** Basic Filtering:  the user can separate out a group of columns (or one column). The transition between hidden/ unhidden state is animated so that the view does not jump to a new state.

Augmented Filtering: While having basic filtering seems useful for exploring the data, finding relevant columns still requires manual work (exploring all the columns to find relevant ones). Moreover, a small number of substitutions in a column may occur randomly and do not reveal any valuable information to the analyzers.

Therefore, an augmented filtering excludes the columns that have fewer yellow cells than the filter number.

GeneTracer [22] to reduce redundancy, they built a button to remove all those columns that contain the same gene bases across all the rows.

Wood et al. [50] provided a button to hide common regions across all sequences. This feature was similar to the GeneTracer redundancy button.

**3.2. Matrix View**

The matrix view enables the user to sort the rows according to the values of different characteristics (for example a disease characteristic such as severity). Design of this view is inspired by the Table Lens [11]. In table lens, the levels are shown by the length of horizontal bars or colour saturation per cell [43]. However, we utilize position and redundantly colour saturations to encode the same property of the data. Each column is divided by the number of its characteristics levels. On top of each column, there is a coloured label that shows the different levels in that particular column. The darker the colour, the higher the level of the characteristics. The coloured labels are placed from  right to left respective to color saturation level.

We exploit position channel for representing discrete ordered data-type because it is the most powerful visual property for encoding all kinds of data [23]. In addition, the colour saturation is a better alternative for the length channel for encoding this ordered information [23].

We also used hue to separate different characteristics that are nominal data and the hue channel is appropriate for separating different categories [23].

The user can perform the following list of interactions in Matrix View:

**Sort:** If the user selects a column header, the rows will be sorted according to the values of that column. Besides, the user can choose between ascending and descending sorting.

**Aggregation:** The “add” button enables the user to make a new column by combining the existing ones with a simple mathematic function in between them

**Zoom:** The user can zoom in and out to the view using “+” and “-” buttons from Main View

**Overview:** At the bottom of each column in Matrix View, an overview of that specific column is provided so that the user can see the pattern of the change for all the row values for that specific disease characteristics column, without the need to zoom. When the Row mode is activated, and a sequence header is highlighted to show the mouse position, it also highlights a row in the overview of Matrix View.

Matrix view and tabular view are linked together by shared row labels. Consequently, when the rows’ positions are changed in one view (for example if the user sorts the rows), their vertical positions will be changed in the other view accordingly.

**3.3. The P-Value View**

# 

There is a pattern [4] within some of the columns that makes them interesting candidates to form new hypothesis.

This pattern suggests a relationship between substitutions in a particular column and one of the characteristic of the rows.

As humans do not complete pattern-detection tasks very well [28], we cannot rely on them to find this pattern in columns.

Commonly biologists use metrics detect interesting patterns. Mann-Whitney U test’s p-value is one of the metrics used for finding relevant positions [4].

Using the Mann-Whitney U test, the severe rows can be separated from others by splitting all the rows into two groups based on the existence of a substitutions in them.

The negative of the logarithm of the P-Value suggests likeliness of the significant difference between the two groups.

This value is shown by the bar lengths in P-value View to help users find relevant columns.

The length channel is the second most powerful channel for encoding the ordinal values [reference]. Therefore we used length to represent the p-value metric.

The P-value view also provides the filtering feature.

The filtering feature enables the user to filter out any column where the length of the bar is smaller than the filter number. This view also lets the user sort the positions based on the bar length. The columns will be sorted from high to low and placed from right to left.

In general, sorting all the rows according to one of the characteristics from top to bottom, a significantly larger proportion of substitutions appear at the top rather than the bottom. As the user might want to focus on those columns with the higher bar length, merely hide/unhide all the other columns is not efficient. Instead, it would be more productive to sort the columns based on the reverse of the p-value (length of bars. keep the bars on top of the columns in Main View, so that the user could go over the bars while observing the columns’ pattern.

The Tabular view and the P–Value view are linked so that if the user re-orders the positions in one view, the corresponding column’s order will be changed in the other. Also they can use the reset button to go back to the original domain ordering.

**3.4. Group View**

The group view helps Users to find related columns (or rows) and group them together to focus on fewer rows (less data dimensions) for future analysis.

grouping feature is defined for both columns and rows. the user is allowed to click on rows and add them to a newly created group. When the user is in

row mode, the user can select different rows to make a group of them.

The user also can separately load each group into the views in order to investigate the group information and to focus on the relationships between the columns. It is more likely that they will make these groups from the relevant columns. Column Grouping The idea is to let the user make different groups from a combination

of different columns. The user can see an overview of the group and its general pattern.

This view supports two categories of interactions with groups: basic grouping and augmented Grouping.

In the Basic Grouping, groups are created by user based on their prior knowledge or observations.

In contrast to the Basic Grouping, in the Augmented Grouping, the users can more effectively detect the columns (or rows) of the same group because the system highlights the relationships between columns(or rows) to guide group creation.

The augmented grouping motivates the graph view.

An overview of the distributions should allow analysts to see the big picture, and identify clusters, trends and outliers that may be candidates for detailed inspection [43]. Therefore, an overview of a group consists of a larger window than Main View information (prior to zooming).

If the user clicks on the overview the content of that specific group is loaded in all other Views. There is a predefined group that contains the entire data-set for the user to be able to go back to the original data (with latest changes).

In order to guide the users to find related columns or related row, an augmented grouping feature is designed. This feature is different for rows and column because they could have different kind of relationships. The visualization of augmented feature is supported in Group View.

There is linkage between Main View and the selected group. When the user selects a group among the previously created groups from Group View, the chosen group’s data will be uploaded into the system. Therefore, the data in all Views matches the data in the selected group. when the user’s mouse hovers a column, the corresponding column is highlighted in the overview of the currently selected group.

**3.5. Graph View**

A node-link representation visualizes the relationship between the columns (or rows).

Column relationship:

Two kinds of relationships between any pair of columns are supported: Complementary patterns and Correlation.

Given that the substitutions in this data-set are represented by values 0 and 1, one may define the similarity between two columns with measures such as Pearson’s correlation calculation for any pair of column. This is however not optimal because many zeros (no substitution)in the columns result in a correlation close to 1 indicating they are highly correlated however is not true.

To alleviate this problem, we propose to use a new measure which ignores the common zeros between columns. Assuming that two columns, X and Y, each have n members,

X = {x1, . . . ,xn} and Y = {y1, . . . , yn}. The measure is defined as follows:

M(X,Y) = n

i=1

xiyi −

n

i=1

xi ⊕ yi, (3.2)

where ⊕ is the logical XOR operation and results in 1 when one of the side equals to 1 the other side equals to 0. This measure, ignores entries with no substitution in both columns, increases when entries with substitutions occurs together and decreases when substitution complements each other. Given that, both positive and negative values are expected.

Row Relation:

The relations between rows are hierarchical. The already designed

Graph View tis used to make a Tree for the representation of this relationship.

Some of the submissions [22, 12] used the Minimum Spanning Tree for constructing the evolutionary tree. The weight of the edges was the Hamming distance between the two nodes [19]. The Hamming distance was the number of positions that differed in any two rows that implied the number of changes is needed to transform one sequence to the other. The Minimum Spanning Tree is a tree in a graph that connects all nodes (rows) and its total edge weight is the minimum of total edge weights of all the possible trees

One alternative representation for relationship between a pair of columns is the matrix visualization [15]. One benefit of using this matrix is, by re-arranging the rows and columns, some interesting patterns would be revealed. However, this option requires a large screen space. One drawback is that we cannot eliminate the cells with 0 correlation from the space.

The second option is using a node-link graph, where there is a link between a pair of columns only if their correlation is non-zero. The link is coloured blue for correlation (numbers greater than 0) and red for complementary (numbers less than 0). Colour saturations and line weights are also redundantly used to encode the same information.

As there are a considerable number of columns with zero correlations, this option conserves the space better than the table representation.

Assuming we know the degree of the relationship between a pair of columns with numbers from 0 to 1 for correlated ones and -1 to 0 for complementary ones, my initial suggestion to encode this information was using a graph or a table representation.

Interaction:

There is also another filter mechanism built into the view that removes the different levels of correlation links.

the user’s mouse position highlights the corresponding row label in Main View (in as well as the equivalent node label in Graph View the user deletes a row from the matching node in Graph View will be deleted.

Two filters are placed to enable the user to sort out the columns based on the strengths of their connection.

**4. Evaluation**

Choosing an evaluation methodology for InfoVis tools is a challenge due to diverse intersections with many other fields such as psychology, art, etc. [51].

For Visualization Design of our tool, we used the Nested Process Model and our work falls into data/operation abstraction, and encoding/interaction technique design layers of this framework. [27]. this model provides two recommendations for evaluation at the visual encoding/interaction layer:

a) The design needs to follow perceptual and cognitive principles [27]. To follow this recommendation in design, we used heuristics for information visualization [51].

b) The design should be able to communicate with the analyzer and be useful towards their problem solving [27]. Therefore we need to assess the initial design idea with domain experts And our general research question was: “How the design of FilooT could help the domain users (Bioinformatics users) solve the tasks problems”. understating of strengths and weaknesses of the design helps to make it better iteratively (Summative Research [10]). Considering the formative nature of our research question, we took User Experience scenario (UE) for InfoVis evaluation [21].

Therefore we came up with a range of specific questions to understand “what do our target users think of the visualization?” [21].

However, the evaluation methodology that is used in this study is a user-based method in which problems are found through the observation of and interaction with users while they use or comment on the interfaces [21, 41].

**4.1. Method**

We conduct a qualitative, User Experience study for system evaluations [21] where users played with the system to answer some tasks with the goal of understanding the design flaws and potential usefulness .

The study follows by an informal open-ended interview to understand user’s opinion about the tool

Although we had a set of pre-defined tasks we explained the users that the process of solving the tasks is more important than reaching to answers. The participants were asked to talk freely about anything comes to their mind about the tool during and after the process.

The followings are the interview guide questions.

* How each view/features helped you to find the answers? Any of them were more or less useful?
* Did you have any difficulties understanding any of the presented information in any of the views?
* Please explain your process and steps of finding the answers.
* Please give me your feedbacks to make each of the views better. Your suggestions may include your ideas of new interactions or refinement of some parts of the design.
* Are there limitations of the current system that would hinder its adoption?

**4.2. Setting**

A laptop computer was used. The studies took place in lab environment.

**4.3. Participants**

Participants are live subjects in the lower mainland. The subjects are undergraduate/ graduate students or postdoctoral researchers. All the participants are over 19 years old. The requirement for the study is that the participants need to be familiar with

DNA multiple sequence alignment concepts in bioinformatics, and they should be interested to participate in this study. No information was withheld from participants.

## 4.4. Procedures

The experiment took less than 1.5 hours.

Participants were asked to read and sign the consent form. Then they were asked to fill out a pre-study with a few questions about their familiarity with the domain and their experience with similar tools.

They were given the Study Task/Data Description. After they read it, they were trained for 10 minutes to learn to use the basic features of the software.

They used the system for 30 minutes. They were encouraged to write down their findings, think aloud and express their thoughts, concerns and their questions at any time during the study.

Experimenter used pen and paper and took notes from observations of participants’ use of the tool.

After the 30 minutes passed, we had a semi-structured interview about their experience with the tool. There were open-ended questions. At the end experimenter thanked them for their participation in this study and they received the compensation for participation, and signed the compensation form.

**4.6. Data**

We use a synthetic data-set from the VAST Challenge 2010, Mini Challenge 3 created by Konecni [20]. VAST Mini Challenge 3 is about illegal arms dealing scenario in which one of the dealers called Nicolai died in a hospital with symptoms consistent with Drafa Fever. In order to develop pandemic response plans, public health organizations need to get more information about the disease. The data-set consists of 56 strains of a particular original virus, which are the result of spreading of a disease over time to different infected people. Each of these strains has a gene sequence of 1400 nucleotides with one or more nucleotide changes from the original virus’s sequence. There is also information about some characteristics for each of the evolved viral strains.

**4.7. Tasks [?]**

In this research, we have used a benchmark data-set/task-set so we are assuming that the tasks are already validated and they reflect the target domain users’ work.

**Task 1**: Identify mutations that lead to an increase in symptom severity (a disease characteristic). Each mutation provides the base substitutions and their position in the sequence, where the base substitutions occur. Report findings in the order of importance. For example,

C → G, 456 (C changed to G at position 456);

Note that in some cases, a combination of the mutations explains a characteristic’s severity. These mutations should be reported together (e.g. second item above).

• **Task 2**: Identify mutations that lead to the most dangerous viral strains.

• **Task 3**: Nicolai has a strain identified by sequence 583. One patient has a strain identified by sequence 123 and the other has a strain identified by sequence 51. Which patient contracted the illness from Nicolai and why?

**4.5 limitations**

The given time can interfere with participants’ solving task. However, the accuracy of the answers is not studies in this research, and only the process of the problem solving using the tool is considered.

**Results and Discussion**

The results of the user study

led me to create a list of requirements as well as develop a guideline for future work. These

results informed the initial design of the tool [6].

During the study session, the participants used the tool to find the answers for three

tasks (see Section 1.3.1). I used paper and pen to write their comments about the system

features. They were also asked to share their findings about the system. There was an

interview with open-ended questions to capture their comments at the completion.

Main View  
Visualization Comments

• The colour choice for mouse hovers (changing from gray to pink) is not discernible for

individuals with colour-blindeness.

• One user suggested we choose different colours for each nucleotide.

• The same user suggested that we choose similar colours for T and A, and similar

colours for C and G, but that the two categories should be differentiated.

• The labels must be readable, regardless of the zoom level.

Interaction Comments

• According to users suggestions, the sliderbar was the most usable feature.

• Another user asked if it would be possible to sort the rows on their ID, and to search

for rows by typing their labels directly into a textbox.

• If the fonts is small the labels could pop up in response to a mouseover.

• One user wanted to be able to type a motif and have the system highlight it with

its local alignment score. This would be useful in order to know if a column is in a conserved region or not (conserved sequences are similar sequences that occur within

DNA sequences). Another user mentioned that the system could be linked to the

UCSC Genome Browser 1 to coordinate data so as to be able to highlight information

about adjacent columns.  
Main View Observations

• The sliderbar was the first and most common interactions with the tool.

5.1.1.3 Main View Discussion

• Regarding different substations in a column, the user pointed out that in a real dataset

different kind of substitutions usually happen in a column, yet there is only one

kind of substitution in a column in this study’s data-set. For example, if an ”A”

changes to ”C” in one row in one column, then all other substitutions in that particular

column,will also be a ”C”. By contrast, in a real data-sets, there are the four kinds

of substitutions in a column. To make the tool adaptable to a new data-set, as some

of the users already suggested, I could use different hues to show different nucleotides

(Figure 5.1).

One potential drawback for this suggestion is that it increases the number of hues in

the tool. Depending on the number of columns in Matrix View, this could interfere

with the hue uses in that view, because using more than eight hue colours on screen is

not recommended (see Section 3.1.1). Instead, I suggest using only one hue per column

for Matrix View and distinguishing between columns by adding extra space between

them. Figure 5.2 shows this idea. More specifically I suggest using Proximity to show

the organization of inter and intra columns (see Section 3.1.1 ). Also the designer

must prevent hue overloading. For instance if they want to use red for showing one of

the nucleotides, they should use a different hue for edge colours in Graph View.

• The data-set carries no information about adjacent columns. According to the participants’

comments, authentic data-sets contain these information. Figure 5.3 shows an

example of neighbouring columns which do have some relationship with each other.

To solve the tasks, researchers need to know if a column is related to its neighbours and what the nature of that relationship is. Examples of these relationships include

codon and motif information in data-sets. However, because the VAST Challenge

stated that the DNA is non-coding, codon analysis and AA sequence analysis cannot

be considered.

• All of the users were familiar with at least one Visual Analytics tool similar to Main

View. Although some of the interactions that they used to see in similar tools were

not relevant to the study tasks, the users desired to see all of the familiar features in Main View.  
Matrix View Comments

Below are users’s opinions about Matrix View. Matrix View (see Section 3.3.1.4) has

an overview to represent the overall trend on each column. A row of this overview will be

highlighted when the user is in Row mode and the user’s mouse hovers over that particular

row.

Visualization Comments

• A user thought that the as yet unimplemented “add” button could be extremely

helpful.

InteractionComments

• A suggestion was to have the system show the row label in response to a mouseover.

• When the user clicks on the coloured label at the top of each column, all the rows will

be sorted according to the values of that column’s characteristics 3.3.1.4. However,

when the user clicks again, the rows order might be changed because the sorting

algorithm allows for sequences that are different but still correct. Therefore, although

the rows will be sorted each time a user clicks on a coloured label, the rows’ order

within a level might be changed. The user did not like this and preferred consistency

and predictability in the column sorting behaviour.

• Another user liked that the coloured labels at the top of each column were clickable

and the clicking on them would sort the rows (see Section 3.24). (He said, “It’s

pretty cool”. Moreover, he thought that these labels could be made more useful if the

different levels were clickable separately so that the system would jump to a state in

which Main View and Matrix View contained the rows with the selected level.

• One user suggested that Matrix View could have a built-in option to keep track, and sort the next column based on previous selections. He mentioned that could be

particularly useful if the analyzer had certain priorities of columns and want to see a 5.1.2.2 Matrix View Observations

• One user sorted the rows based on different characteristics and used the Overview and

P-value bars to see which characteristics contributed more to overall danger.

5.1.2.3 Matrix View Discussion

The compilation of the user’s comments on Matrix View shows that this is one of the

most frequently used views. The users’ suggestions could be used as a guideline to make

this view more accessible and useful for completing the study tasks. Below is my extension

of how one of the suggestions should be considered for further implementation:

• For the user comment about clicking the coloured label in order to update the view

with new information, I suggest that instead of Main View suddenly switching to new

information, the system automatically move the vertical scrollbar which shifts Main

View until the new, selected column is reached. Through this process, the user sees

that the change in Main View is fluid, while keeping track of how the information

changes.

• The above strategy could be used to link Graph View and Main View. When the user

clicks on a node in Graph View, Main View needs to contain the matching row. In

the case of a row needing to appear in Main View window, vertical scroll bars could

be used. In the case of a column, horizontal scroll bars could be used to move Main

View columns.

• It seems implementing the ”add” button will be useful. Levels need to be associated

with a particular number, and this number should be editable by the users.

5.1.3 P-value View

5.1.3.1 P-value View Comments

P-value View elicited a few comments. This view has a section that filters out the

columns based on the reverse of the p-value of the Mann- Whitney U statistical test that

suggest a pattern in the columns (see Section 3.2.4.3).