SAGENERATE – A Visualization Tool for Generating Hypotheses

Timothy Chan
University of British Columbia
2366 Main Mall
Vancouver, BC, Canada
timchan at cs.ubc.ca

Zsuzsanna Hollander
University of British Columbia
2366 Main Mall
Vancouver, BC, Canada
zsuzsa at cs.ubc.ca

Abstract

As new technology emerges, so does the growth of datasets. The past decade has been marked by an unprecedented growth in experimental and computational biomedical data, especially in the areas of Genomics and Proteonomics. This explosion has lead to many challenges in analyzing these large data sets. Consequently, this has sparked a surge of interests in both the Scientific Visualization and Information Visualization community to provide such tools to biologists specifically for their type of data.

This paper presents a software system SAGENERATE specifically designed to visualize high-dimensional data sets generated by a high-throughput gene expression technology known as SAGE (Serial Analysis of Gene Expression). This software system uses various information visualization techniques, which will significantly reduce the time it takes to search for interesting candidate genes with particular characteristics.

1 Introduction

The recent advent of high-throughput gene expression technologies, such as microarrays and SAGE, is one of the key reasons why there has been a phenomenal growth of biological data. These technologies have become highly popular. On top of their growing popularity, these technologies generate vast amounts of data in single experiments. Thus, the past few years have been marked by a surge in interest in developing tools able to analyze this kind of data. Unfortunately tools that are made for analyzing such data tend to be created by large commercial software companies and are often unaffordable. As far as we know, the freely available tools are inflexible and are far inferior to commercial software.

Analyzing this type of data can be quite cumbersome as the dimensionality of the data is high. For instance, a typical genome wide expression experiment involves several thousand genes often surpassing 20,000 rows of expression data. Consequently, using visualization tools could be highly beneficial by providing a means of showing the data in an understandable manner.

In a typical gene expression analysis of a disease, one often compares the expression of each gene between a diseased tissue and a normal tissue. The most differentially expressed genes (up or down regulation) are candidates for genetic markers or drug targets for a particular disease. However, there are cases where a researcher may want to look for a gene that is differentially expressed but has a low average "count" since those genes may actually be the genes that may have led to the onset of a particular disease. These slightly altered genetic expressions could mean earlier detection of a disease. This is beneficial in many diseases such as cancer where the earlier it is detected, the easier it is to treat, and thus, significantly improve patient survival. Furthermore, such information could give us better drug targets as targeting a gene at the beginning of a cascade of reactions or pathways could mean an increased chance to stop the onset of a disease and again, increase survival rates. Unfortunately, in order to isolate these potential candidate genes, a researcher must arbitrarily choose some statistical filters (ie. P-value, means, standard deviations, permutation scores, etc) and then print out the list. This is a time-consuming task since in a typical SAGE library there are over 30,000 genes to be analyzed.

Here we present a visualization tool we call SAGENERATE implemented in java specifically designed for SAGE to help relieve some of these frustrating monotonous tasks that most SAGE researchers must endure. Our tool implements many visualization techniques including filtering, the use of color, parallel coordinate plots, dynamic filtering and querying, and

fisheye lenses. All these techniques have the same goal and that is to ease the analysis of dense datasets.

This paper is organized as follows. In the next section, we will discuss the data we are using for our application in more detail. Section 3 will describe our solution in detail including how the infovis techniques help solve many problems. In section 4, we describe a few scenarios of use of our system. In section 5, we will briefly describe how we implemented it. In section 6, we describe a few related systems. In section 7, we evaluate our system describing its strengths, weaknesses, and lessons we have learned. In section 8, we briefly discuss our contributions and finally, in section 9 we end with some ideas we have for future work on this system.

2 SAGE Data (Serial Analysis of Gene Expression)

Here we describe briefly the SAGE technique and what type of data it produces and the data our application takes in.

SAGE is a high-throughput gene expression technology developed by Velculescu et al. in 1995 [1]. The approach has the capability of whole transcriptome characterization of a cell, which allows detection of variations of the cell's transcriptome in abnormal conditions (such as diseased cells). Unlike the microarray technique, which are based on relative expression levels, SAGE measures the level of gene expression based on the frequency of occurence of the 3` signature SAGE tags of 10 to 20 bases (depending on the restriction enzyme used) unique to each transcript. Although tags often map to more than one gene (and vice versa) due to tag length and a biological phenomenon known as alternative splicing, this technique does give us a lot of insight on the molecular workings of a diseased cell versus a normal cell and often hypotheses are derived form this type of data.

As mentioned before, SAGE libraries are generally very large in that in a typical SAGE experiment, 60,000 tags are generated (with ~18 to 20,00 unique tags) and when sequenced deeply, the number of tags can reach 150,000 to 200,000 tags. Thus, plotting one SAGE library versus another one could give a very messy and dense plot. Table 1 shows two very small SAGE libraries consisting of only 4 tags/genes where the counts column tells us how many instances of the TAG with the corresponding sequence there are in a given cell/tissue.

TAG	Library 1 Count	Library 2 Count
AAAAAAAAAT	5	35
AGTAAAACAA	6	17
ACCCCCCCAA	1	23
GAAAAAAAAT	3	50

Table 1: A small example of what SAGE data looks like.

Sample data will include data like the above table along with their corresponding statistical values such as means and standard deviations. Our tool can load a table with 12000 rows and 20 columns however, when loading large tables it does take a performance hit.

3 Description of solution

As mentioned in the introduction, SAGENERATE is designed to help researchers in the analysis of SAGE data sets as it provides the user with the visualization of the data through various types of graphs and by the means of infovis features and techniques such as filtering, querying and so on. As the name implies, SAGENERATE helps the user to generate meaningful hypotheses.

The user interface includes four graph panels where the table and/or graphs of choice are displayed (see fig. 1).

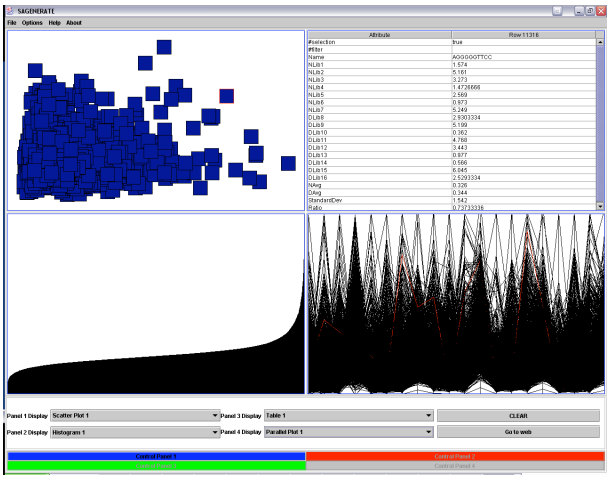

Figure 1. GUI with the three types of graphs and a table. The data shown data set contains 20 columns and over 12,000 rows.

Drop-down menus are part of the main graphical user interface and facilitate the selection of one of the 3 types of graphs and a table for each panel. The control panels on the bottom of the screen open a pop-up control panel for the corresponding graph from the panels, and allows the user to filter, zoom, resize, label, sort, query, and

color the data points on the graphs (see fig. 2), although coloring, sorting, and fisheye technique can only be applied to scatter plots at this point. Multiple file analysis is also supported (up to four in the current version). For each file the user has a choice to view the data on a graph that is more suitable for the task at hand. There is a control panel associated with each file that allows for a number of information visualization techniques to be performed.

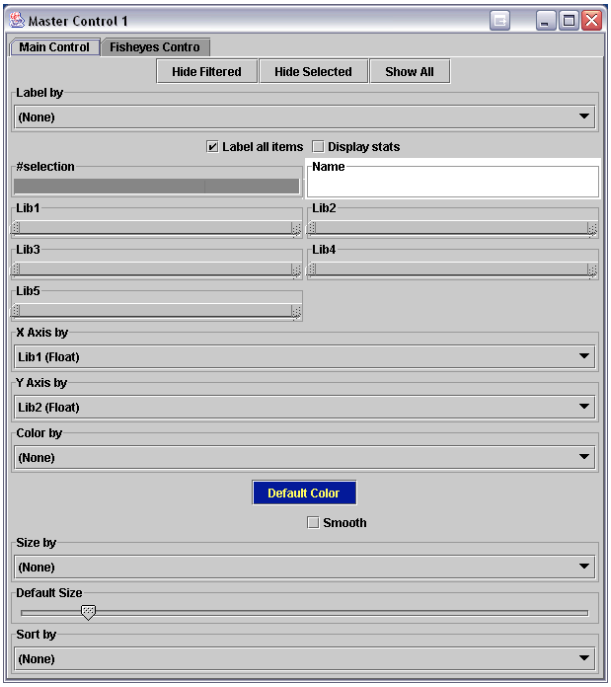

Figure 2. Control Panel that controls the fisheye, labeling, querying, filtering, coloring, sizing, and sorting

3.1 Graphs

The data points can be seen through the use of three types of graphs: scatter plots, parallel coordinates, and histograms. The scatter plots are well-known data displaying techniques and are very useful for showing the relationships between two variables [2]. For example scatter plots can aid the user in finding similarities and/or differences between the control and disease counts. The limitation of this type of visualization of the data is that it can only represent two dimensions. To compensate for this drawback the parallel coordinates [3] were added to enable the user to view the long and skinny nature of SAGE data (that is few columns, with thousands of rows). Parallel coordinates provide a good overview of displaying the entire table all at once and allows one to do pair-wise analysis of SAGE libraries with ease. This allows a researcher to pick out interesting trends between pairs of libraries with ease (ie. similar expression levels between pairs of libraries). In addition, for experiments involving different stages of a disease (such as early to late stages of cancer), parallel coordinates allows easy detection of these trends also (ie. The decreased expression of a key gene over a period of time – could lead to the possibility of gene therapy targets) and could potentially give valuable insight on the expression behavior of key genes corresponding to the progression of the particular disease.

On the parallel coordinate plot the data dimensions are represented by vertical lines and the data points on two adjacent coordinates are connected with a line. This type of graph makes the sighting of gene expression trends realizable among the different dimensions. The histogram, on the other hand, illustrates the range and distribution of values for a single column (a count or a statistical measure). This is useful in the discovery of outliers, skews, and spread of the data. Biologically, it can help us find genes with differing characteristic quickly such genes that are highly expressed, similarly expressed or lowly expressed.

If a data point is highlighted in any of the graphs (this can be achieved by a mouse click on the point), it is also automatically highlighted on all of the other graphs that contain the specific data value. This way the user can have a great overview by seeing the expression pattern of the tag on all of the graph types. Also, the table that can be selected from the drop-down menus on the GUI shows the values of each row for that tag. As you pick points from any of the graphs, the table can be populated by holding shift and selecting multiple points.

3.2 Dynamic Filtering

As it was mentioned earlier, the regular SAGE libraries have tens of thousands of tags; as a result the graphs illustrating these points can be very crowded. When analyzing a big data set, researchers usually want to scale down the data to fewer points, to focus their attention to the points of interest and/or outliers. The obvious solution to this problem is filtering. Therefore, double range sliders were added that allow the user to choose an interval of numerical values and to focus only on a small part of data, making the isolation of candidate genes possible. Filtering of data can be done based on any and all of the numerical attributes of the data table and can be performed rapidly with the provided sliders. As a result, the respective graphs are automatically updated to reflect the filtering. Furthermore, the effect of filtering is global, meaning that if a user changes the range of a variable the filtering will be reflected on all of the graphs corresponding to the same data table. Our application also allows multiple analysis of the same table. What we mean here is that we can pick 4 scatter plots (and show them simultaneously) and choose 2 different columns for each plot (ie. Permutation score vs average count, permutation vs standard

deviation etc.) and use the central master control to control the thresholds we impose on the table and observe the behavior of all four graphs simultaneously. Again, trends can easily be observed and help a researcher pick genes of interests easily.

3.3 Occlusion

The overlapping of points cannot be avoided when dealing with very big data sets. Filtering of data, by using the range sliders on the control panel, can help avoid some of the occlusion. If the filtering is not desired or there are some more overlapping points on the graph at the end of the filtering process then the fisheye lens [4] can be used (by selecting the "Fisheye Control" tab on the control panel). The fisheye lens allows the user to look at points that are overlapped on the scatter plot and also, the user can get a feel for the number of overlapping points. Even as the fisheye tool contorts the dense scatter plot for easier viewing (works like a magnifying glass with some zooming), our tool still allows a user to pick points as it is in its contorted state. Thus, this allows one to keep track of interesting points that are difficult to view in a non-contorted scatter plot. Another feature of SAGENERATE tool that helps in eliminating occlusion is the "Size by" option on the control panel. This functionality resizes the squares corresponding to the points on the scatter plot according to the crowdedness of the area they reside in. The points in the most crowded areas get very small and the outliers are the biggest. The sizing can be done by any of the numerical columns (attributes) from the table.

3.4 Color

Color is used for the differentiation of graphs corresponding to different data files. For example, the graph panels that contain the various graphs corresponding to different data tables have different border colors where the graphs corresponding to the same table are bordered with the same color. (ie. Red bordered graphs all correspond to the same table) This feature allows the users to easily see which graphs illustrate the same data table and which show different tables. Also, the data points on the scatter plot can be colored using a variety of colors. Almost any color imaginable can be generated as the entire range of colors is available in its color control panel (RGB, ALPHA, HSB). Such flexibility would even appeal to color-blind individuals as the shade/lightness can also be adjusted.

Another useful feature that helps visualization is the coloring by column, which uses shading to differentiate between the values of the numerical column selected. So, even if, for example, a scatter plot shows only two dimensions of the data, via coloring by another column, one can have comprehension of three dimensions of the data.

3.5 Comparative analysis

To facilitate the comparison of multiple data sets our tool allows the user to visualize up to four files at a time. This feature is useful when the researcher needs to know what the similarities and differences are between the gene expressions of various diseases or tissues. For example by loading four SAGE libraries simultaneously the user can perform comparative analysis of brain, prostate, breast, and lung cancer data. SAGENERATE also features convenient shortcuts. For instance the option list in the "Options" menu provides the user with the possibility of loading all scatter plots, or histograms, or parallel coordinates at once, without having to select them one by one from the Panel Display menus from the bottom of the GUI.

4 Scenarios of use

In the following sections two possible scenarios of use of SAGENERATE are described.

4.1 User Analyses SAGE Libraries to Find Points of Interest and to Generate Hypothesis

Suppose Researcher *A* has the following hypothesis : Genes that are consistently expressed across all lung cancer libraries (ie. high permutation score and low standard deviation within the lung cancer library group) but not expressed at all in the normal libraries are interesting candidate genes because they could very well be the key genes that may have lead to the onset of the disease.

Researcher *A* has a file with lung cancer SAGE libraries, normal lung SAGE libraries, and their corresponding statistical parameters of interests (average of cancer libraries, average of normal libraries, ratio of cancer mean/normal mean, standard deviation between cancer libraries, and permutation score between cancer and normal libraries). According to his hypothesis, he needs to find genes with differential expression in the normal versus lung cancer tissues. At the same time, he needs a set of candidate cancer causing genes with a low average and low cancer library standard deviation. Since his SAGE library has over 20,000 tags, manual analysis of the data would be time consuming and inefficient. Thus, in order to reduce the dimensionality of the data, he needs to filter the data according to some arbitrary set of statistical parameters. The goal is to end up with a small manageable list with interesting statistical properties and thus allow the generation of candidate genes for biological analysis and thus generate several hy-

potheses from this list of candidate cancer causing genes.

A decides to use SAGENERATE to help him along with the generation of hypotheses. First, he loads his file then chooses one scatter plot, one histogram, one parallel coordinate and one table to be displayed on the four graph panels. Since the data set is so big the displayed graph is extremely dense and difficult to read. *A* opens the control panel by clicking on the "Control Panel 1" button and via the range sliders filters he starts to play around with the statistical parameter sliders for permutation scores, cancer library standard deviation, and cancer library mean. Recall, that his goal is to generate a manageable list of genes with a high permutation score but with a low average and standard deviation. The problem is that it is difficult to decide what is considered high and what is considered low. So *A* decides that 100 is a manageable list so he begins by taking roughly the top 100 permutation scores by sliding the slider until he sees around 100 points. He then slides down the average mean until he sees about half as many points on the screen and finally, does the same with the standard deviation. He now finds 25 candidate genes on the scatter plot and observes an interesting trend. There are about 15 points that have very similar permutation scores and averages and standard deviations but they seem to be lying on top of each other. *A* wants to select them but he has troubles. Luckily for him, he is using SAGENERATE and his frustrations are quickly subdued as he clicks the tab on the master control panel to the fisheye lens feature and adjusts the lens size. As he mouse-overs the scatter plot, he is able to see some of these hidden points and is able to click on some of them. However, he still has trouble for a couple points. He then clicks back to the main menu and adjusts the size of the points. As he does this, the points move farther and farther part from each other until *A* has room to click on the points of interests. After clicking on all the points he now has a list of 25 genes in his table and he quickly glances at it. There are still too many points for him to sift through as each group consists of 10 libraries (10 cancer and 10 normal SAGE libraries). He looks at the parallel coordinate plot and sees that the library points can be divided into roughly two groups (two dense plots that are similar to each other across the cancer libraries). He decides he should split these genes up and examine them separately with some biological experimentation. Specifically, what would happen if he introduced this set of genes via cloned vectors to a normal cell line of lung tissue. However, he first has to verify that these genes are not artifacts (noise) of the SAGE technique and thus, needs to do some Real Time PCR experiments first. At this point, he has 2 lists of genes that may play a crucial role in lung cancer development.

4.2 User Analyses Four Files Containing SAGE Libraries to Find Similarities between Four Diseases

Researcher *B* has the following hypothesis:
Different cancers maybe different at the macromolecular level, but may in fact be similar at the sub-molecular level and thus have implications on how the diseases are treated.

Researcher *B* has four files each containing different SAGE libraries corresponding to four different types of cancer (brain, breast, lung, and prostate). *B* wants to find the genes that have similar expression in all four libraries and have similar expression pattern with the genes known to be involved in some of the four types of cancers.

B loads the four files, corresponding to the four types of cancer libraries, into the SAGENERATE. She opens a histogram for each file by clicking on the "Options" menu and selecting "Load all Histograms". She chooses the permutation score as coordinate for each of the graphs. The histograms give her an overview of the distribution of the permutation score for each file. Since her libraries contain over 15,000 tags there are a lot of data points and *B* is only interested in those that have high permutation score. She opens the control panels one at a time and filters out the tags with low permutation score (the points with low differential expression between normal and cancer). Next, she changes the histograms to parallel coordinates to have an overview of the remaining data points.

B notices that there are a few lines with the same expression pattern across the coordinates between prostate and breast cancer data. By opening the control panel she chooses to label the graph by the tags. After the labels are placed on the graphs she detects a few tags that are present on all of the parallel coordinates and have the same general expression pattern. She clicks on the tag of interest then clicks on the "Go to web" button on the bottom of the screen that takes her to the NCBI public database [9] and maps the selected tag to the corresponding gene(s). She takes the list of genes and checks them against the literature to see if there are any articles relating these genes to these types of cancers. If there are any, this strengthens the hypothesis that the two are related and thus, the remaining genes are candidates that require further biological experimentation.

5 Implementation

SAGENERATE was implemented in the Java programming language with the help of the InfoVis Toolkit [4]. The InfoVis Toolkit was developed at the French

National Institute for Research in Computer Science and Control (INRIA), in France, with the purpose of helping developers create Java Swing applications containing advanced two dimensional information visualization components.

The graphical user interface was implemented with Java Swing. The histogram was implemented in such way that it conforms to the data structures used for the scatter plots and parallel coordinates that were created using the InfoVis Toolkit. The same toolkit was used to write the functions/classes for the filtering, querying, labeling, coloring, sorting, and resizing of data points. The code for the control panel from the toolkit was modified to make it more suitable for our application.

6 Related SAGE or Gene Expression Visualization Systems

One of the more useful gene expression tools we have come across is designed specifically for SAGE data is known as DISCOVERYspace, which was presented recently in a poster session at the Canadian Gene Expression Conference [6]. This tool is quite comprehensive and is linked to various SAGE databases and provides some basic statistical functions (ie. P-values, ratios). Expression comparisons can be made visually in scatter plots (see fig. 3) or Venn Tables. The plot is dense but there is a zoom feature. However, the application has no histogram or parallel coordinate capabilities and no dynamic sliders for filtering. All of these features were desirable by the researcher we interviewed. In addition, the application has limited statistical capabilities and does not allow one to import custom tables with custom statistical values (ie. Permutation-scores). Lastly, the tool is not even available to the general public, with a user base of only around 30.

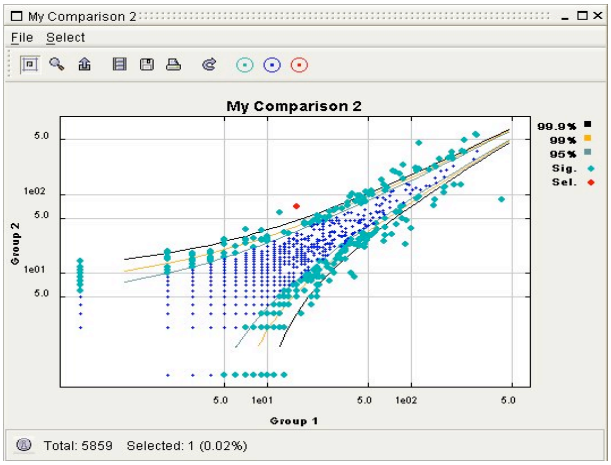

Figure 3: DISCOVERYspace screen shot of its scatter plot feature.

Another powerful gene expression visualization tool is developed by Spotfire [7]. Spotfire's Functional Genomics tool is based on its flexible analytic application environment known as DecistionSite.(see figure 4) Their visualization tool is quite flexible tool and allows various plots including parallel coordinate plots, scatter plots, and 3d plots. It is quite powerful and even includes sliders and its features go way beyond the scope of our project. (see fig. 4). However, this tool costs several thousand dollars and obviously unaffordable by most academic institutions.

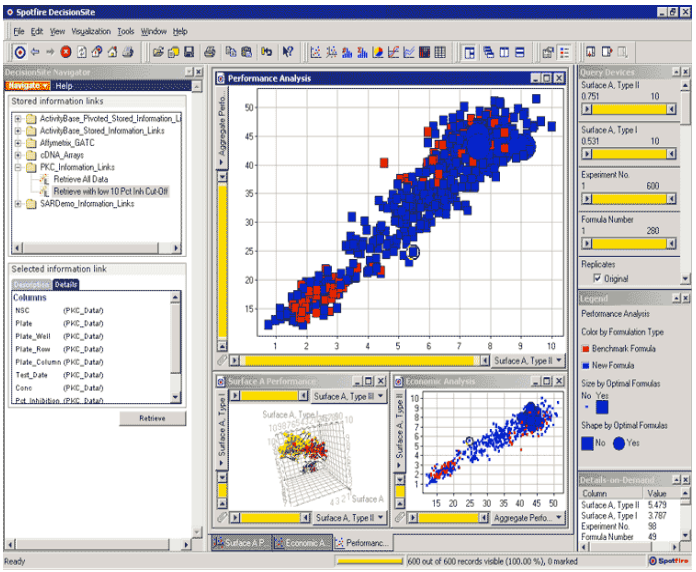

Figure 4: Screen shot of Spotfire's Genome Analysis Program Designed for Microarrays

Some publicly available tools to examine gene expression are available from the National Cancer Institute [8]. Their site, known as CGAP (Cancer Genome Anatomy Project) is quite useful for retrieving publicly available SAGE libraries and it contains some useful tools for examining gene expression including its SAGE Absolute Level Lister, SAGE Anatomic Viewer, and SAGE Digital Gene Expression Displayer. The SAGE Absolute Level Lister simply shows the distribution of the transcript expression levels of a user-defined SAGE library. The SAGE Digital Gene Expression Displayer distinguishes significant differences in gene expression profiles between 2 pools of SAGE libraries. Both of these tools though are non-graphical and are thus, difficult to read. However, their SAGE Anatomic Viewer, which displays the relative expression of a given gene in normal and malignant tissues throughout the body, gives a nice graphical representation and can be quite useful [5].

7 Evaluation

In the following two sections, we will discuss the key strengths and weaknesses that SAGENERATE possesses. Following these two sections, we will discuss the lessons we have learned from this project.

7.1 Strengths

Many of SAGENERATE's strengths are based on functionality that other system just do not have.

Flexibility and Plenty of Useful Incorporated Information Visualization Functionalities

Recall in section 3, we describe several infovis techniques that SAGENERATE possesses from dynamic filtering sliders, color control, fisheye functionalities, parallel plots and so on. Many of these techniques are not available in currently easily accessible software. (see Related Works Section).

Freely Downloadable

SAGENERATE is open source and freely available. We consider this a strength because most tools for this dataset domain are not free. Bioinformatics is considered the next big economic wave and thus there are several companies that are pushing for software development in this field (ie. Spotfire). These companies create very nice tools for bioinformatics research however, for many research institutions, it is just not an affordable solution. Furthermore, because bioinformatics is still in its infancy stage, many consider it strictly a research field of study. Thus, even companies that can afford the software find it a difficult decision to invest in expensive software when the turn over rate of products being produced via bioinformatics research range from 5 to 20 years. This leads to the main motivation of our project, and that is to create a freely downloadable tool to researchers in this field to increase their productivity. In fact, currently, we already have some interests in our software.

Multiplatform

Since SAGENERATE is coded in java, which automatically makes it multi-platform capable piece of software. That is SAGENERATE should run on all types of operating systems. In fact, much of our coding and testing was done on both a Microsoft Windows PC and a Macintosh computer (see sample screen shots of Mac in figures 14 and 15).

7.2 Weaknesses

Most SAGENERATE weaknesses can be attributed to the lack of time to implement the functionalities. Most of these weaknesses can easily be remedied in future works. Below we discuss a few of these weaknesses and later in the Future Directions section of the paper, discuss how we will deal with these weaknesses.

Limited to Four Tables

SAGENERATE is limited to importing 4 tables of data. Our original proposed plan was to implement a system to handle one large table of data containing a set of SAGE libraries for one specific normal tissue and its corresponding diseased pair (see section 2). However, upon further discussion and evaluation with our client additional loading of tables was requested. Additional tables allow one to compare multiple diseased/normal experiments and possibly derive similar genetic information from different diseases and thus, may suggest that the two diseases may react similarly to the same treatment. For instance, suppose that brain cancer is in fact similar to breast cancer at the sub molecular level. Such knowledge would have major implications in the way the two diseases are treated. Thus, with the time restrictions we were under, we decided to allow a user to add 4 tables since there are 4 panels in our initial design and thus, allow a user to compare up to 4 diseased/normal pairs at the same time.

Although we already went beyond our original proposed plan of handling one large table of data, this is still a bottleneck of the system that hinders it from comparing more diseases simultaneously. Thus, we address this issue in section 6.

Missing Controls

Another weakness of the work is the absence of separate controls for the parallel coordinate plot and the histogram plot. Even though dynamic filtering affects all 3 graphs, our application's control panel's focus is the scatter plot. That is, the color control, size control, and fisheye controls only affect the scatter plot.

Performance and Scalability

In terms of performance, SAGENERATE has the same bottlenecks that plague all java applications. In spite of the strides that java has made in the past few years, it is still considered one of the slower languages. In fact, the current implementation of SAGENERATE requires a fairly powerful machine with plenty of memory. With small data sets (rows not going beyond 1500), SAGENERATE performs well, however, when you

import tables with high dimensions, it slows down a lot. Our pilot tests show a considerable slow down in the application where when you import a table with > 2500 rows and 20 columns, the performance of the system is noticeably compromised. Although we were able to load in tables with 20 columns and 5000 rows, the system slowed down so much that it crashed the SAGENERATE in some cases. However, even the performance of applications such as Microsoft Excel would be affected by data sets this large especially when things such as sorting and graphing are conducted. Thus, in spite of being tested on fairly high-end machines (1GHz G4 powerbook and ~2GHz PCs with >= 512 mb of RAM), SAGENERATE can only handle fairly small datasets. Being that this is a project course, it is not something we can easily fix in the span of a few weeks and is a problem that goes beyond the scope of this project. However we will discuss some ways to boost the performance of our system in the future works section.

7.3 Lessons Learned

There were a number of key lessons learned during this project and course. Some of which include the following:

Information visualization techniques can be applied to any type of domain and significantly benefit productivity.

A lot of people in these different domains are unaware of the possible benefits these visualization techniques can bring them.

There are many free useful visualization toolkits that can make these infovis techniques realizable in a matter of weeks without purchasing an expensive piece of software as long as you know how to program.

The use of colors is great for helping organize our graphs (ie. blue borders all refer to the same table that is being modified). However, careful thought must be taken into account when dealing with colors and is in fact a very important HCI principle: *The Color As a Supplement Principle*. The idea of this principle is to minimize the use of color since a significant percentage of the population is color-blind (7%) and often too many colors create clutter. Thus, we made sure that our application is still useful even if we had no colors. Careful planning on laying out the panels to correspond with the combo boxes were done purposely (that is the combo box in the left corner corresponds to the left corner panel and so on). In addition labeling was done too as each control frame is labeled at the top to show which table it is modifying.

When you have a lot of controls, it is difficult to decide on what controls are the most important and should be on the main interface without some user input and user experience. Some decisions were made based on time restrictions. That is, we tried it one way but did not have time to evaluate how useful it is to have those controls the most visible (*The Visibility Reflects Usefulness Principle*). We wanted to try tabbed panels but we did not have time for it (see future work).

Dynamic filtering (via sliders) is a very useful technique for providing focus on a few sets of data points in a high dimensional data set. However, having too many sliders can be cumbersome and cause the control panel to be too big and cluttered. Due to time constraints, our application does not deal with this problem.

Performance issues are difficult to deal with and can easily be overlooked when designing a system.

8 Conclusion and Results

This paper presents a visualization system, SEGENERATE that can help scientists in the process of generating hypothesis based on SAGE libraries. Here we have developed a strong prototype for visualizing SAGE data with the following features:

1.Graphing scatter plots, histograms, and parallel coordinates.

2. Selectable scatter plots with corresponding tables.

3. Double sliders allowing for dynamic filtering of data points, decreasing dense plots.

4. Vast color control and usage – helps in labeling and organization.

5. Fisheye controller that helps reduce occlusion.

6. Size controller that also helps with occlusion problems.

7. Selected tags can be connected to NCBI database and mapped to the corresponding genes.

8. Has a capacity of four tables and allows simultaneous viewing all four plots in the same panel.

Considering the time and the limited experience we have in java (especially swing), we strongly believe we have done a good job for this project as we have surpassed our original proposal by quite a large margin. Much of this success is due to Fekete's Infovis toolkit

which allowed us to add a lot of functionality we didn't think was possible (ie. Fisheye lenses, dynamic querying and filtering, etc). In addition, our client was quite happy and surprised with our first prototype. Lastly, as we will also be using this program in future work ourselves, we will continue to develop it until we have implemented and integrated all features we would like in it as described in section 9.

9 Future Work

Even though we believe we had a successful project, time constraints have hindered the vast potential of this project. That is, we just could not implement everything we wanted in the system. Thus, there are a number of improvements and useful components/features that could be added to the system to make it a very useful complete tool for biologists and bioinformaticians. Below we describe a few minor implementations that would take only a couple weeks to implement (and would have been implemented given more time) and a few major long-term goals of the project.

9.1 Minor Work

As mentioned in section 7.2.1, it would be useful if the application could handle more than four graphs. However, we need to make sure that the system does not become too cluttered as a result of these additions. To do this, we would modify the system to be able to contain frames and these frames can be minimized inside the frame (much like how Microsoft Products such as Excel handles multiple instances of worksheets). Each of these frames would contain one of two types of panels. One type would look exactly like our current application where up to 4 graphs/tables can be put into a single frame and can be compared with ease. The second type of frame would contain only one type of graph or table and will grow (with the use of tabbed panes) as we add more and more tables. Such a design promotes flexibility and thus appeal to more people.

Another improvement (which was mentioned in the weaknesses section) that would not take long is the addition of separate control panels specifically designed for the parallel coordinates and histogram graphs. To reduce the number of free panels, these control panels can be added as tabbed panes in the current master control panels for the scatter plots.

One last minor improvement that we would like to do is the implementation of a gray out filter. That is to give the user the option to eliminate or fade to a gray-like color the points that are filtered-out. This would be a useful feature because in some cases we want to be able to see the points that were filtered-out but have them sort of in the background as points of reference.

9.2 Major Work

In spite of the name of our project, it is not made specifically for SAGE data. That is, one may import any type of table even if it is not SAGE data and view it. This makes the tool flexible and allows researchers to import custom tables with their own statistical values. However, it would be nice if we created some built in processing tools such as programs that calculate permutation scores and other statistical values so that one may simply import raw SAGE data and analyze it on the same application. Another key improvement would the addition of various other diagrams, graphs, and tables. These could include Venn diagrams (like those provided in DISCOVERYspace), zoomable pie charts, and scatter plots that allow trend lines to be drawn through them with formulas. Each type of graph/diagram will include control panels specific for it.

An ambitious follow-up to the mapping of SAGE tags to current genes is to integrate an information extraction tool such as BioRAT [10]. BioRAT is an open source information extraction tool built specifically for extracting specified information from biological abstracts and papers. Specified information is defined in BioRAT templates which can be customized to suit the users' needs. Thus, BioRAT essentially does the reading for the researcher and plays the role of a "virtual assistant". For example, if you wanted to know what genes bind or interact with the candidate gene you have isolated in your analysis in SAGENERATE, BioRAT could help you do this task automatically. BioRAT queries databases such as PubMed [11] and automatically extracts wanted information (defined in the templates) from papers. Such an integration of these tools could potentially save researchers hundreds if not thousands of hours of tedious and frustrating tasks.

Adding all the above features are all useful and beneficial from the user's stand-point but as mentioned in the weaknesses section, we must not overlook the performance issues. Hardware-wise, making use of hardware-accelerated graphics cards, multiple processors and more RAM can improve performance significantly. But in order to make use of this hardware, software must be written for it. For instance, to make use of multiple cpus, software must be threaded. And to make use of hardware-accelerated graphic cards, software must make use of the available hardware instructions. Another option to the performance issue is use of a low-level language such as ***c***. However, there is a major trade-off in using OS dependant languages and that is the loss of portability. Bioinformatics software is consistently running into such bottlenecks (ie. BLAST) and thus, often require the aid of server farms. It is crucial

that all of the above issues are taken into consideration for any future work.

10 Credits

We would like to give special thanks to Tamara Munzner for her guidance in this project and in our Information Visualization course. In addition, we would like to thank our client Raj Chari for his input and evaluating comments of our product.

References

1. V.E. Velculescu, L. Zhang, B. Vogelstein, et al. Serial Analysis of Gene Expression. Science 1995 Oct; 270(5235): 484-7.
2. J. Beritn. Semiology of Graphics. University of Wisconsin Press. 1983.
3. A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multidimensional geometry. In Proc. IEEE Conf. on Visualization 1990: 361-78.
4. G.W. Furnas. Generalized Fisheye Views. Proceedings of Human Factors in Computer Systems CHI. 1986; 16-23.
5. J.D. Fekete. The InfoVis Toolkit. Research Report RR-4818, INRIA Futurs, May 2003. URL: http://www.lri.fr/~fekete/InfovisToolkit/
6. Liang, P. SAGE Genie: A suite with panoramic view of gene expression. 2002. PNAS. 99(18): 11547-11548.
7. Varhol, R., Zuyderduyn, S., Oveisi-Fordoei, M., Fjell, C., Robertson, N., Siddiqui, A., Jones, S. *Discoveryspace: A Gene Expression Analysis Tool.* 2004. Candian Gene Expression Conference
8. URL: http://www.spotfire.com
9. http://cgap.nci.nih.gov/SAGE/
10. A.E. Lash, C.M. Tolstoshev, L. Wagner, et al. SAGEmap: a public gene expression resource. Genome Res. 2000 Jul; 10(7): 1051-60.
11. D.P.A. Corney, B.F. Buxton, W.B. Langdon, et al. Extracting Biological Information from Full-length Papers. UCL-CS Technical Report: RN/03/17. 2003. URL: http://bioinf.cs.ucl.ac.uk/biorat/
12. URL: http://ncbi.nlm.nih.gov

Appendix A. Screen Snapshots of SAGENERATE

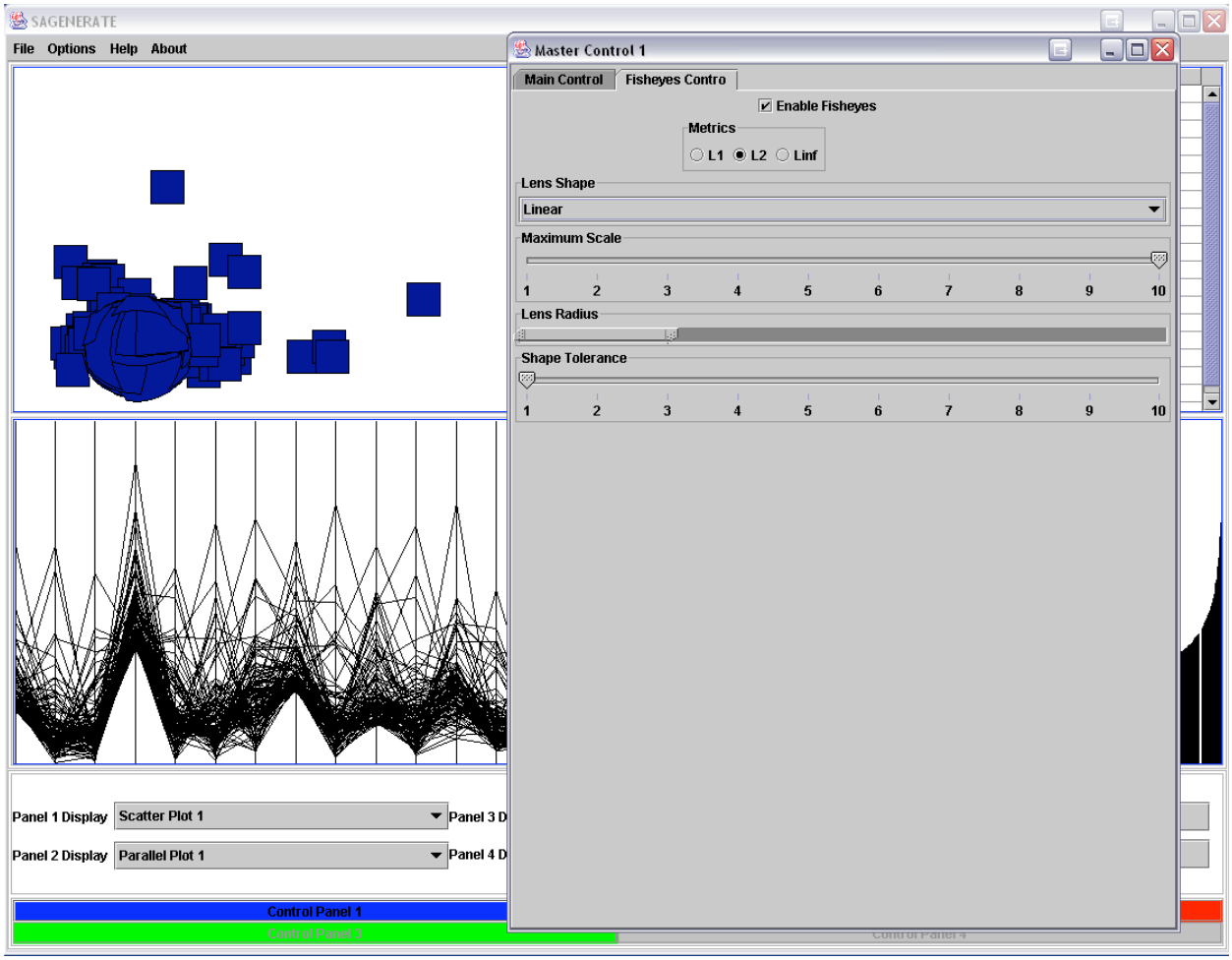

Figure 5. Fisheye technique applied to scatter plot

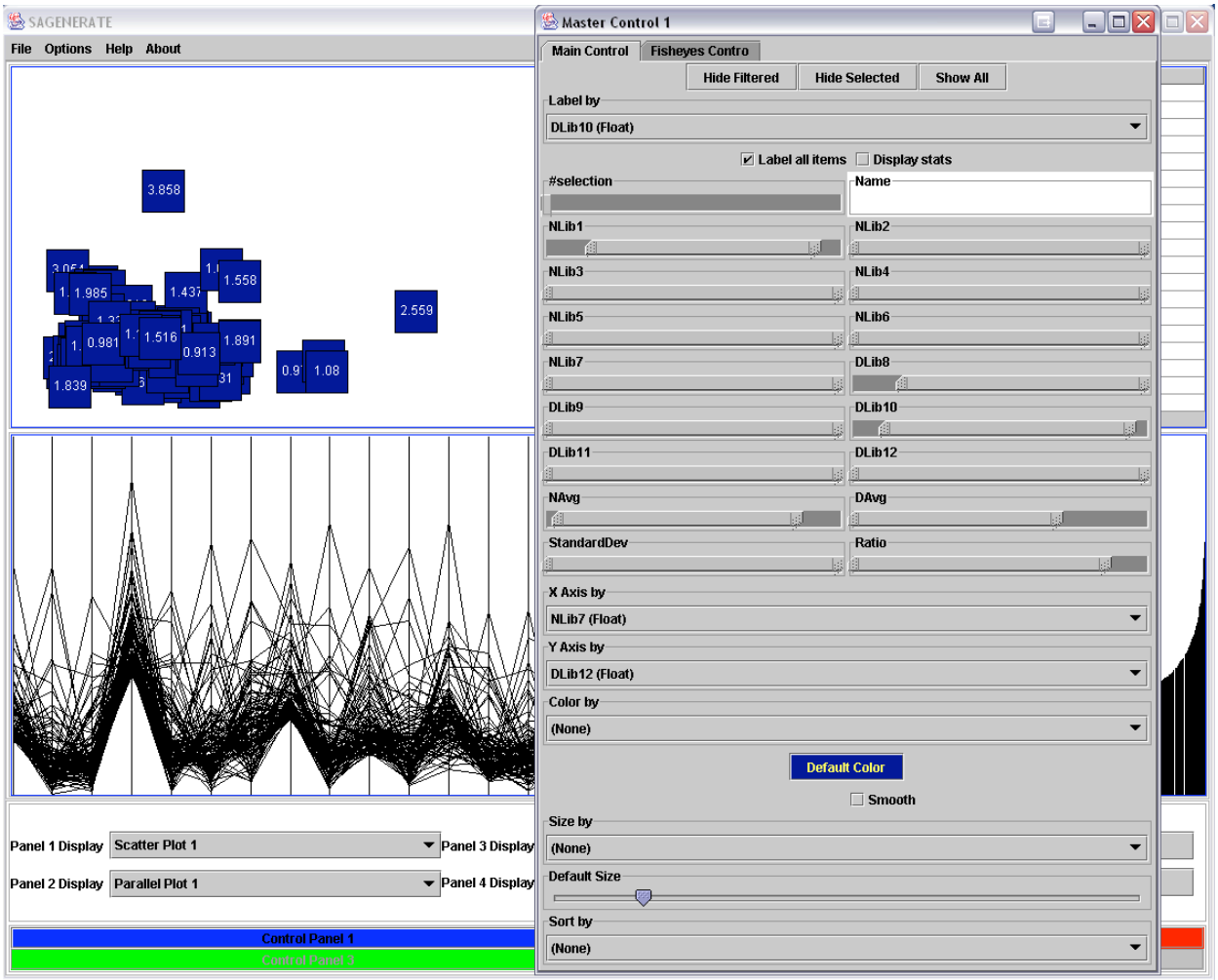

Figure 6. Labeling of points on the scatter plot by one of the columns in the table

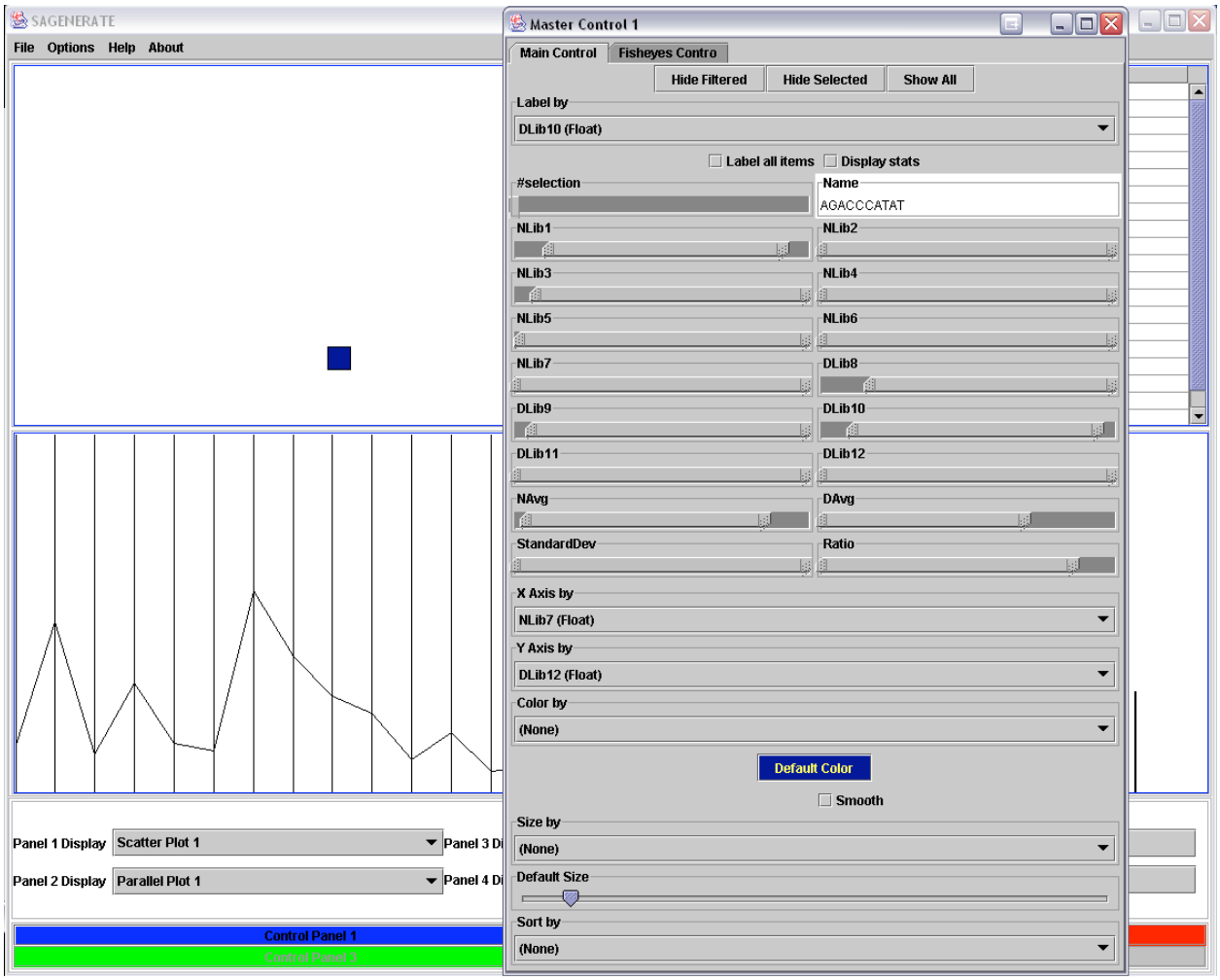

Figure 7 Querying the data set for the tag “AGACCCTAT” (see Name field) will result in the illustration of the data point, corresponding to the tag, on the graphs

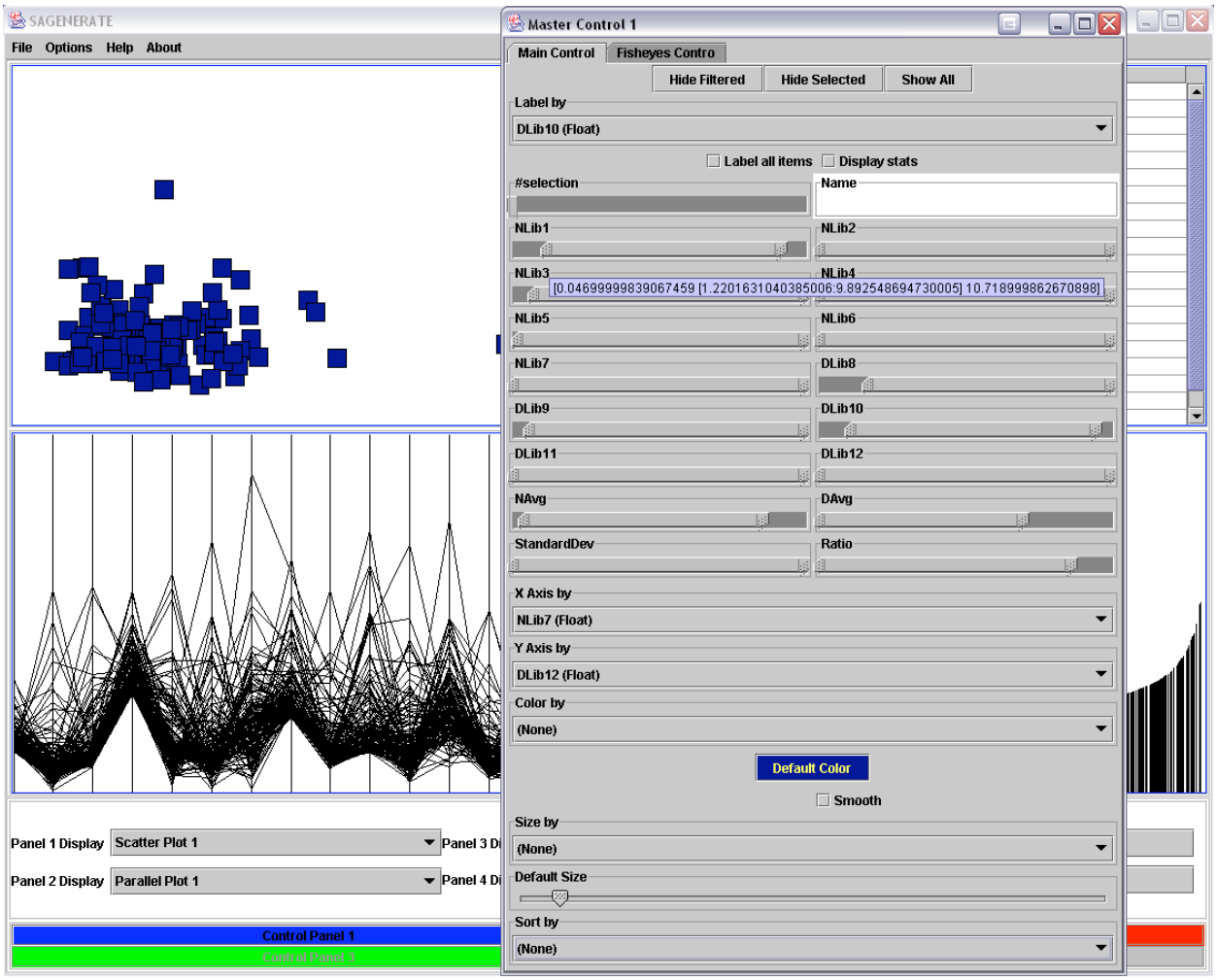

Figure 8. Filtering of the data points can be performed by dragging the range sliders. The range of the specific column will show when the mouse is held over the slider.

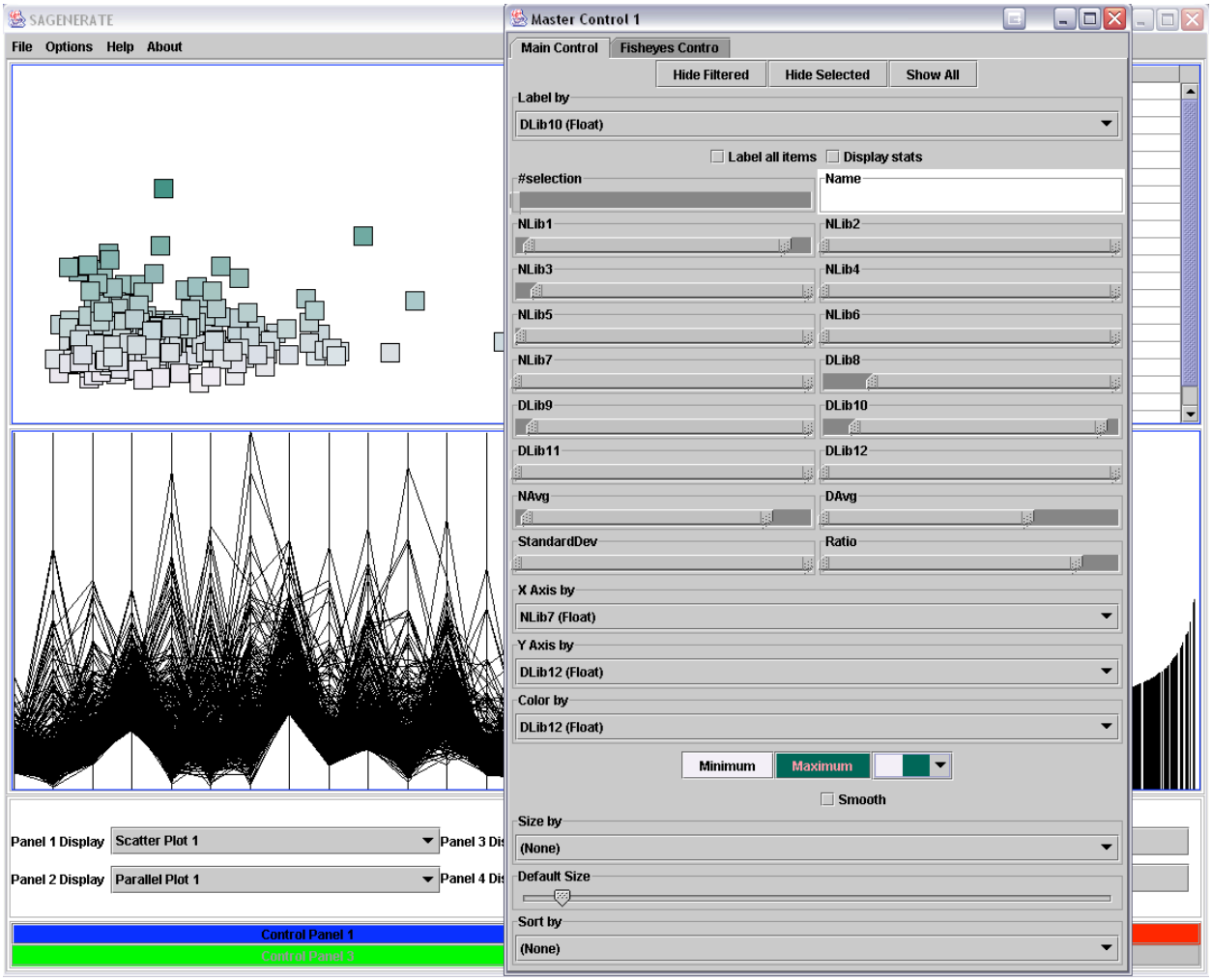

Figure 9. The points on the scatter plot can be colored using the "Color by" Option on the Control Panel. There is also an option to change the shading according to the values of the column selected from the drop-down menu.

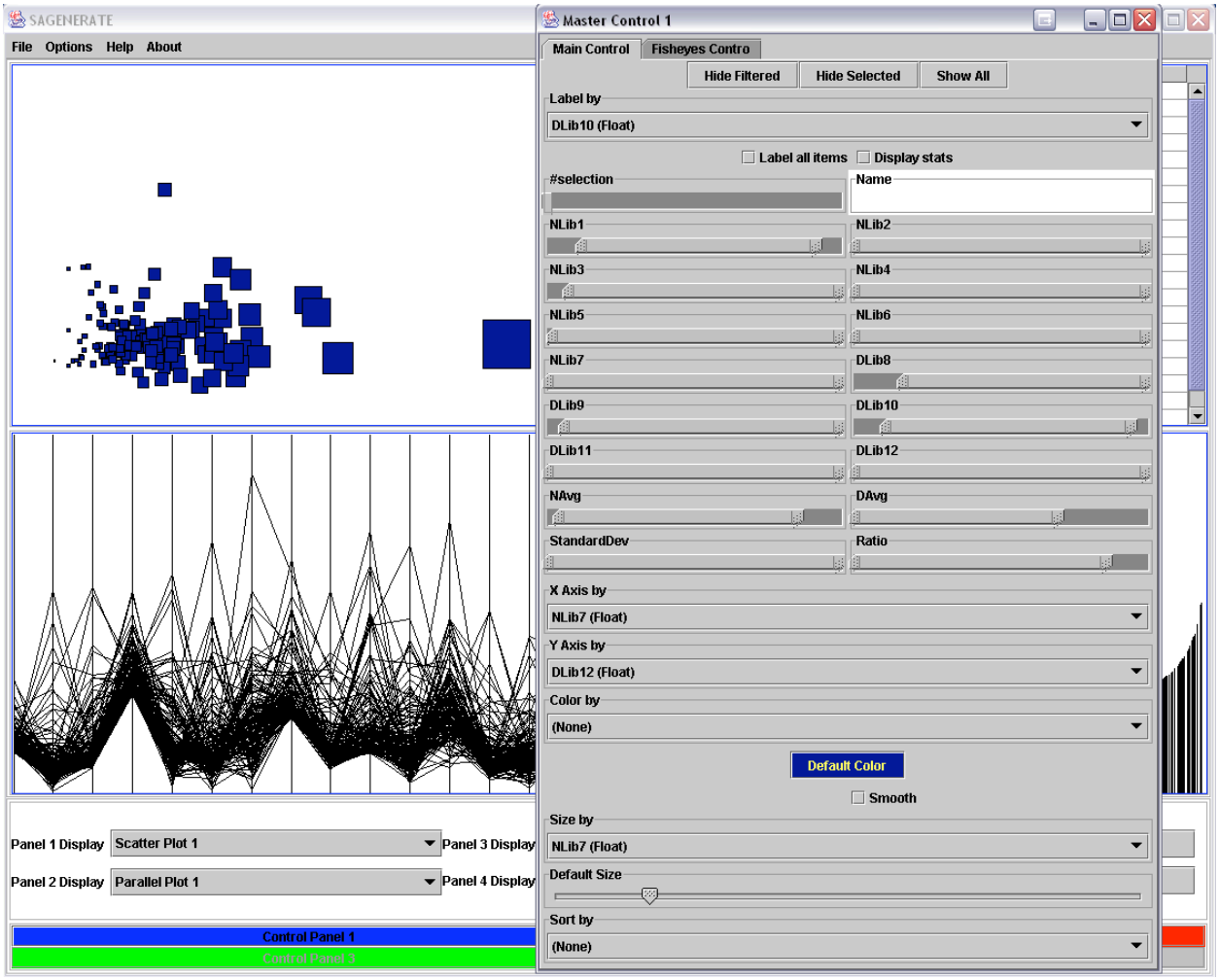

Figure 10. The size of the data points can be set by a column of the table. This way the more crowded regions will have less occlusion.

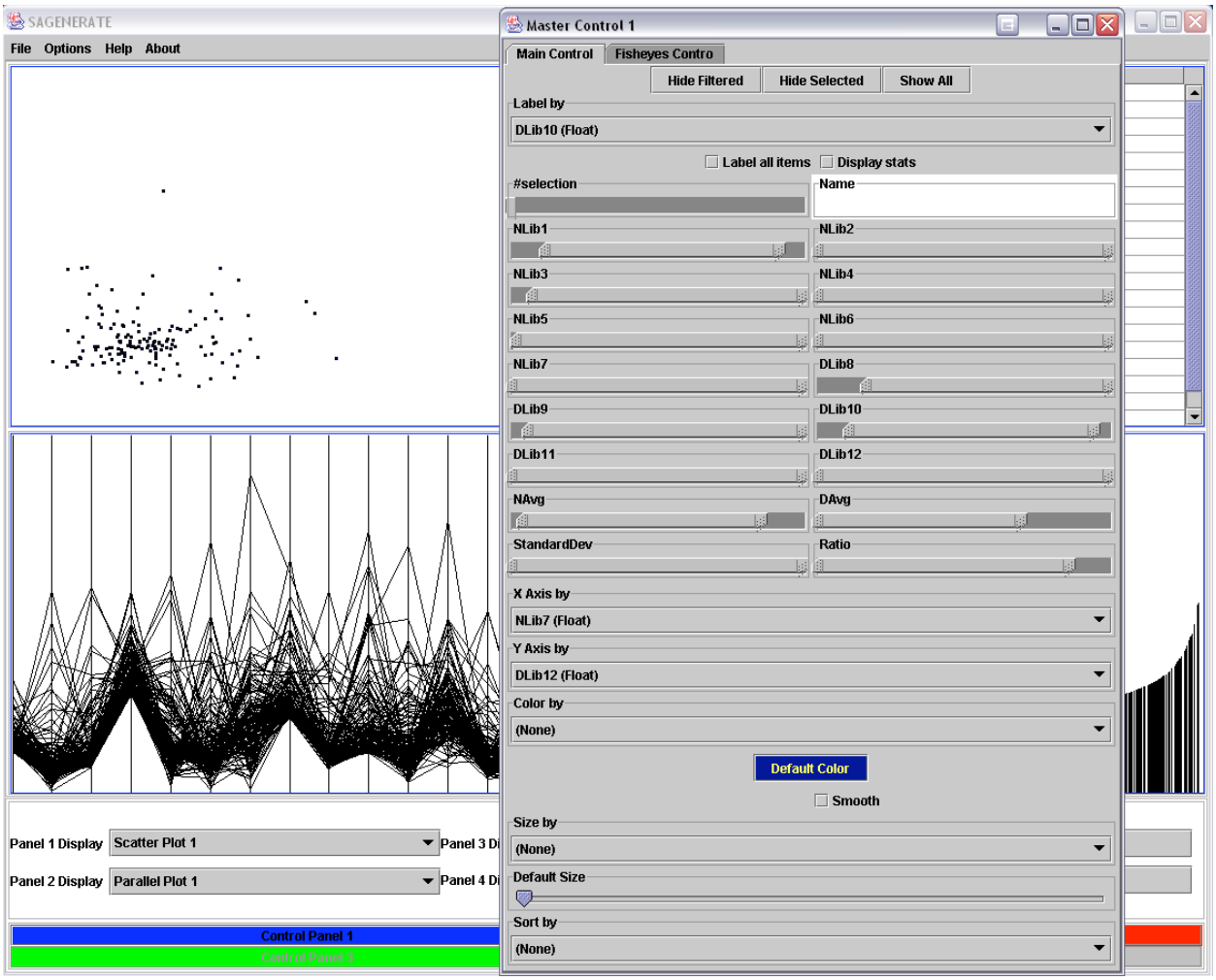

Figure 11. The size of the points on the graph can be modified to avoid occlusion

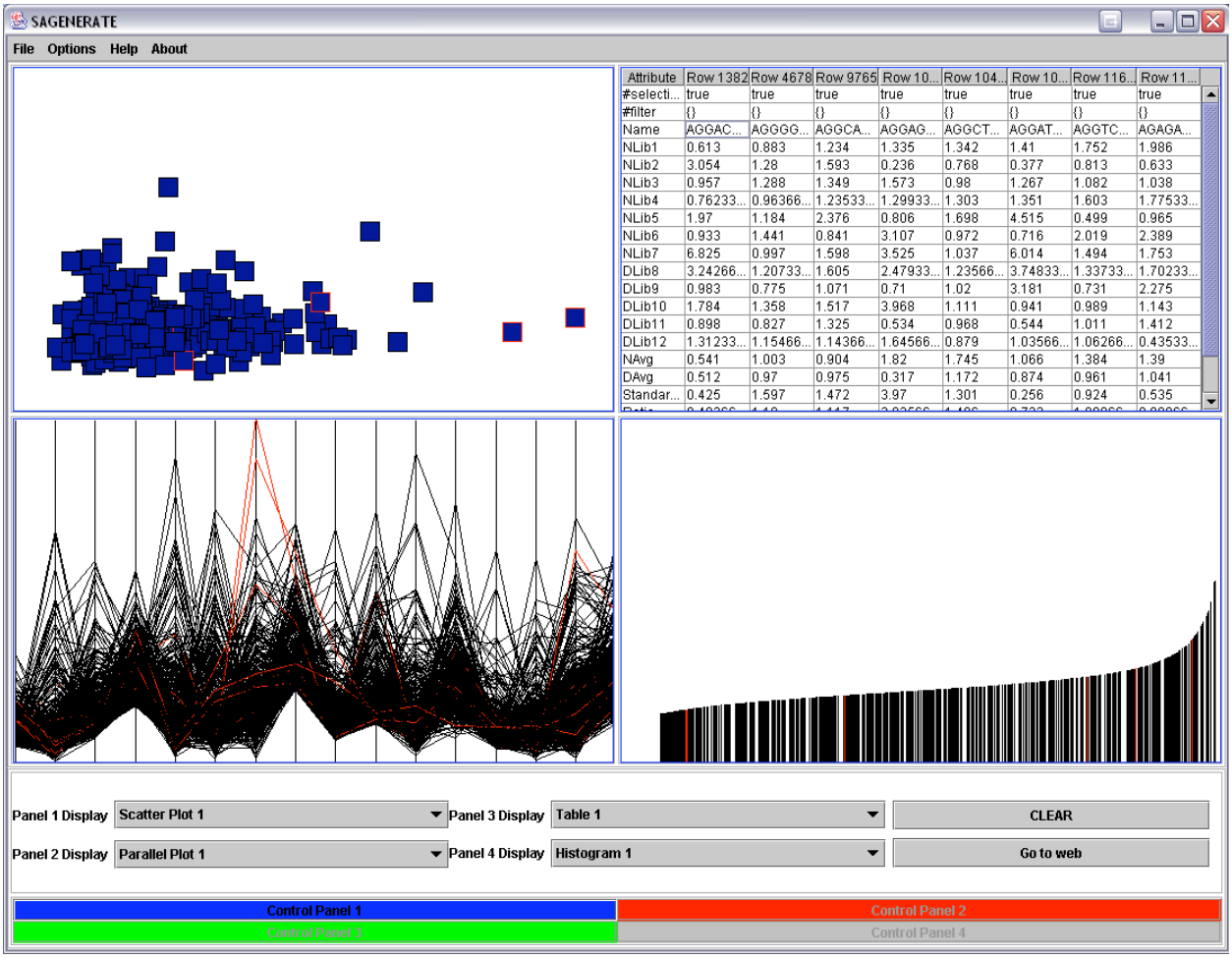

Figure 12. The points on the graphs can be selected by clicking on them. To select multiple points the Shift has to be pressed while doing the selection. The values of all the selected points will be show in the table.

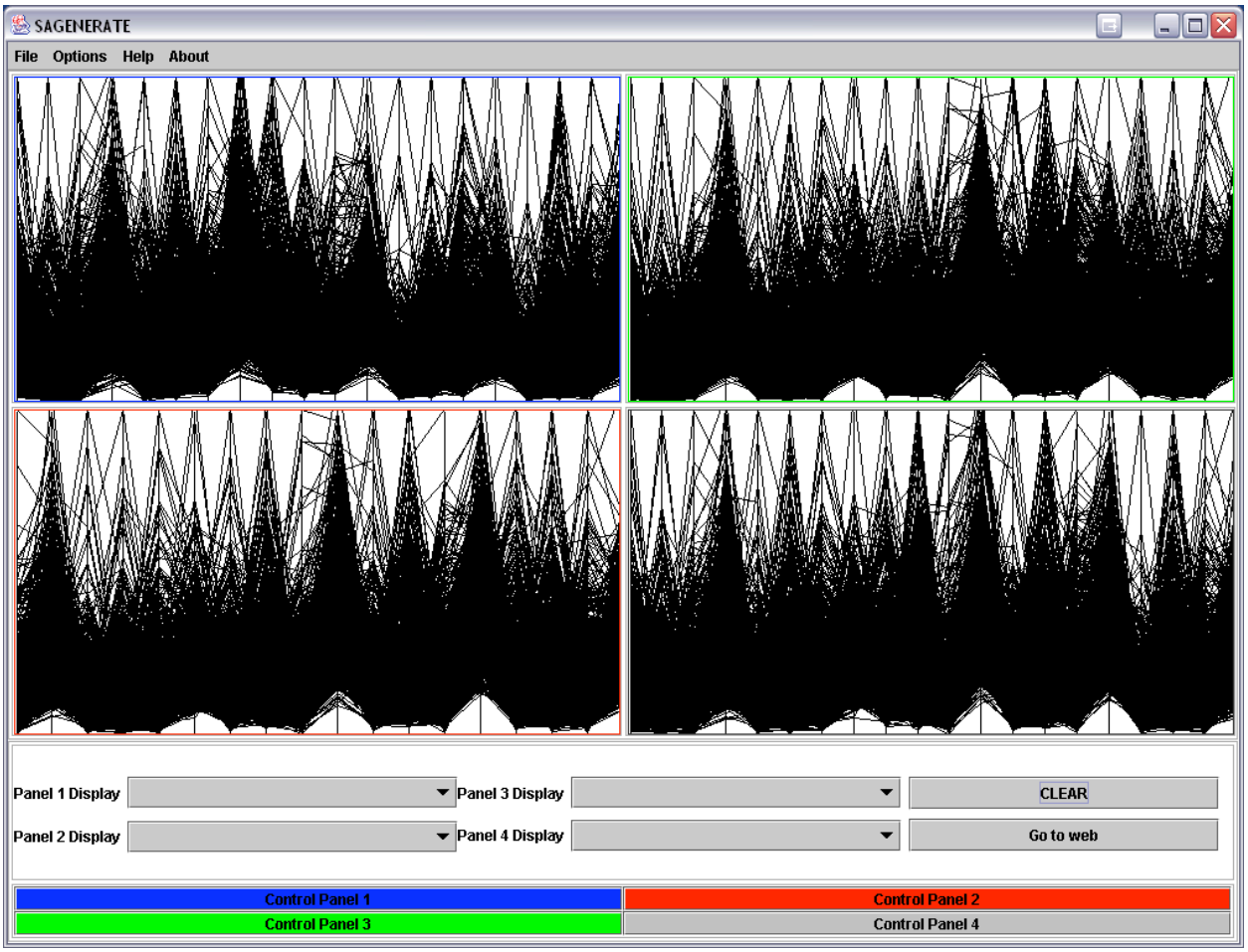

Figure 13. GUI with 4 parallel coordinates corresponding to 4 different files (the different colors of the border indicate that the graphs are not showing the same files).

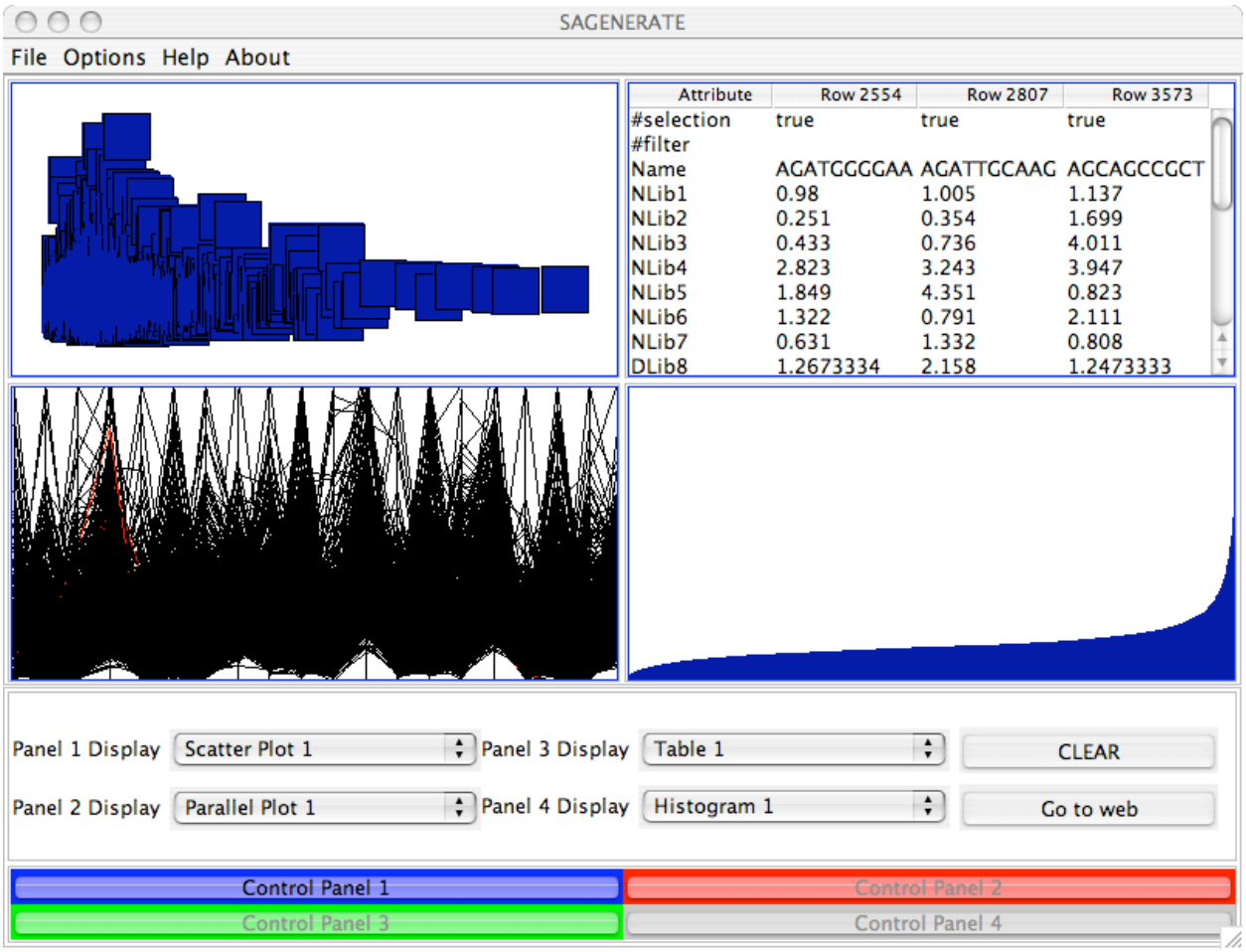

Figure 14. Sample GUI shot of SAGENERATE running on a Macintosh Computer

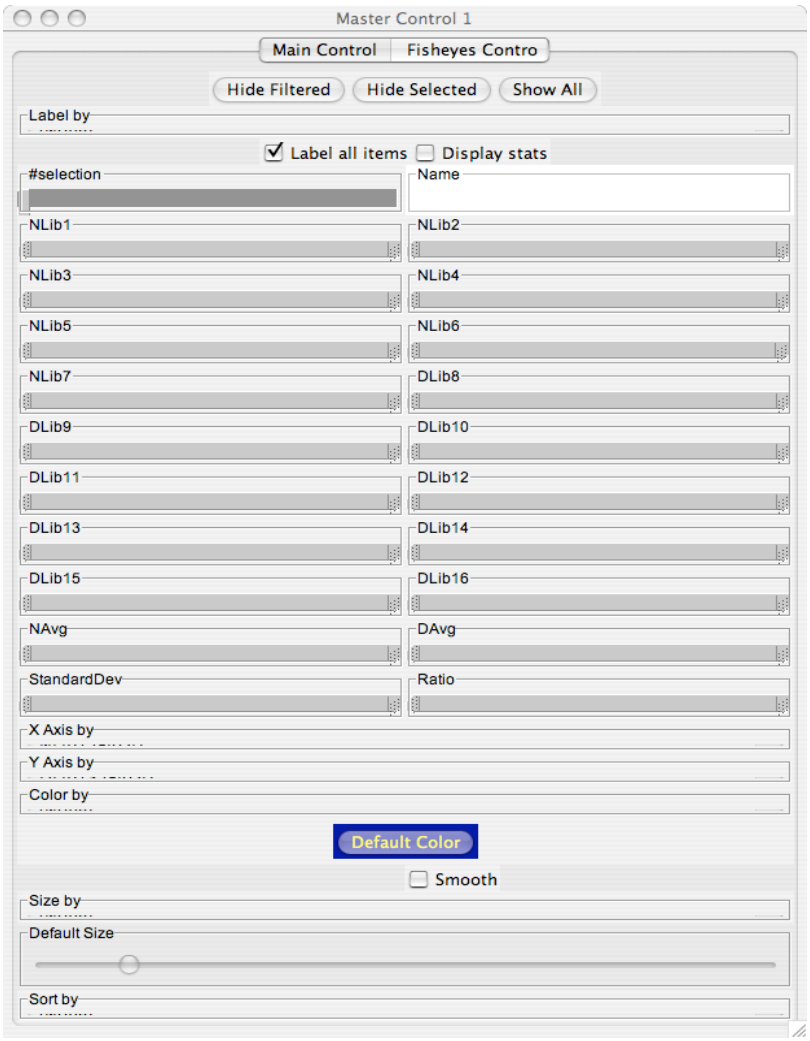

Figure 15. Sample GUI shot of control panel on a Macintosh Computer - OSX