Dual Analysis of DNA Microarrays

Cagatay Turkay
Department of Informatics,
University of Bergen
Cagatay.Turkay@uib.no

Julius Parulek
Department of Informatics,
University of Bergen
Julius.Parulek@uib.no

Helwig Hauser
Department of Informatics,
University of Bergen
Helwig.Hauser@uib.no

ABSTRACT

Microarray data represents the expression levels of genes for different samples and for different conditions. It has been a central topic in bioinformatics research for a long time already. Researchers try to discover groups of genes that are responsible for specific biological processes. Statistical analysis tools and visualizations have been widely used in the analysis of microarray data. Researchers try to build hypotheses on both the genes and the samples. Therefore, such analyses require the joint exploration of the genes and the samples. However, current methods in interactive visual analysis fail to provide the necessary mechanisms for this joint analysis. In this paper, we propose an interactive visual analysis framework that enables the dual analysis of the samples and the genes through the use of integrated statistical tools. We introduce a set of specialized views and a detailed analysis procedure to describe the utilization of our framework.

Categories and Subject Descriptors

I.3.m [**Computing Methodologies**]: Computer GraphicsMiscellaneous; J.3 [**Computer Applications**]: Life and Medical Sciences-Biology and genetics

General Terms

Design

Keywords

interactive visual analysis, visual analytics, microarray data

1. INTRODUCTION

Analysis of gene expression data has been one of the key analysis in molecular biology in the recent years. This type of datasets are produced by microarray experiments and represent the activity of genes under different conditions (different samples). The activity of the genes are referred to as expression levels. The analysis of expression data is used to find groups of genes that are responsible for certain biological processes. Such analysis have a wide application area from agriculture to pharmacy.

i-Know '12 September 05 - 07 2012, Graz, Austria

Statistical analysis methods are widely used in the analysis of microarray data where the data is handled as a table. Conventionally, statistical tools perform the analysis on either the rows or the columns of the data. An example of such a tool is hierarchical clustering where the analyst clusters either the genes or the samples of microarray data separately. However, in gene expression analysis, the analyst is interested in discovering the relations between the genes, as well as between the samples. Moreover, microarray data usually contains the expression levels of thousands of genes over a small number of samples. Such type of datasets is referred to as "large p small n" data and traditional statistical methods are known to be problematic when they are applied on such datasets [12].

Due to these problems in the utilization of statistical analysis methods in gene expression analysis, analysts need solutions where they can use these methods interactively on both the samples and the genes. The main motivation of this paper is to provide mechanisms that enables the expert to perform statistical analysis as a natural step of interactive visual analysis (IVA). In this paper, we build upon the ideas from our previous study where we investigated the dual analysis of items and dimensions in the analysis of high-dimensional datasets [25].

In this paper, we propose a framework to perform the dual analysis of genes and samples using an interactive visual analysis environment. In our framework, we integrate the most common statistical analysis tools that are used in microarray data analysis. All the visualizations and the statistical tools can be used on both the samples and the genes. We provide an iterative dual analysis scheme, where the analyst explores, compares, and, refines the results of statistical methods at each iteration.

We describe how the integration of statistical tools into IVA is achieved. We introduce three specialized views to use in the analyses. We firstly introduce a categorical data view, which enables the comparison of the results of different statistical tools, e.g., clustering, linear discriminant analysis, with available meta-data on the samples, e.g., gender of a sample. Secondly, we introduce the "Dynamic PCA Plot", which guides the user to explore more "useful" projections of the data interactively. Finally, we introduce the "deviation plot", which provides a mechanism to visualize certain statistics of the samples or the genes. This view computes and visualizes the deviations in these statistics that occur via interactive selections. All the three views are used in linking & brushing operations together with a set of traditional visualizations such as, scatterplots, heat maps or histograms. To achieve the duality in the analysis, all the views are capable of visualizing either the genes or the samples.

Additionally, we describe the dual analysis procedure where all the introduced views and integrated analysis tools are used in combination. We demonstrate a set of analysis procedures that brings

new opportunities in gene expression data analysis.

2. RELATED WORK

Gene expression matrix analysis aims to discover relations between genes and samples at the same time. This makes them a suitable domain for visual analysis. Saraiya et al. provides an evaluation of the currently available microarray visualization tools [19]. They evaluated these tools in terms of the insight gained through the analysis and they aim to provide a guide for analysts in selecting a suitable visualization tool for their data. Quackenbush provides an overview of the techniques and challenges in the computational analysis of microarray data [16]. Gehlenborg et al. [5] reviews an wider range of works on the visualization of omics data and provides valuable future perspectives.

Heat maps are direct visualizations of microrray data, so they are included in most of the visualization tools [20, 4, 18]. Due to the large sizes of these heat maps, they are ordered using mainly hierarchical clustering and annotated with the resulting dendograms [4].

In *Hierarchical Clustering Explorer* [20], Seo and Shneiderman use an interactive dendogram over heat maps and they provide an interactive framework that contains conventional visualizations. They include a cluster comparison view where the user can compare two clustering results. In a recent study, Lex et al. introduce MatchMaker [14], that is used to compare groups of columns. In their work, they provide a use-case where they use their methods to compare clusters of gene expression data. Treeview provides a focus+context visualization for the microarray heat map [18]. It tries to overcome problems of dense matrix displays, by highlighting the selected items in a zoomed view. Another important work is the SpRay application by Dietzsch et al. [1]. The authors perform an analysis of the raw gene expression data together with derived features related to the genes. We take a different approach in our paper and perform the analysis of the genes and the samples in parallel.

Caleydo provides a more general visualization solution by displaying gene expression data together with pathways [13]. It proposes joint and visually linked views of gene expressions and pathways in a viewing mechanism called “bucket". Rubel et al. [17] integrate clustering and visualization in the analysis of 3D gene expression data. Authors integrated the data clustering for 3D gene expression analysis into their PointCloudXplore visualization tool. Additionally, there are a number of commercial tools like Spotfire [22], GeneSpring [6] and JExpress [3] that provide a rich set of views and statistical analysis routines. Jeong et al. [8] demonstrated how their tool, iPCA, is utilized in performing PCA operations interactively. The results provide mechanisms to observe how PCA results update in response to several interactive modifications. The presented method is similar to our dynamic PCA view. However with our solution, we achieve the seamless integration of the selections on both the data items and the dimensions into the interactive PCA computations.

In our previous work, we introduced a dual analysis framework for the analysis of high-dimensional data [25]. In this general framework, the analysis is carried out over the items and the dimensions in a linked, parallel fashion. Conventional visualizations, such as scatterplots are used to visualize the dimensions as visual entities. The current paper, extends this framework with more specialized views, namely, the categorical data view and the deviation plot. We also demonstrate the application of the dual analysis idea in the analysis of microarray data.

In this work, we extend the current literature in gene expression data analysis with the dual interactive analysis of genes and samples. We propose a visual analysis scheme where the analysis of genes and samples is carried out in parallel. This analysis involves the integration of statistical tools into the visual analysis cycle and it utilizes special views for the evaluation and comparison of the results of these statistical tools.

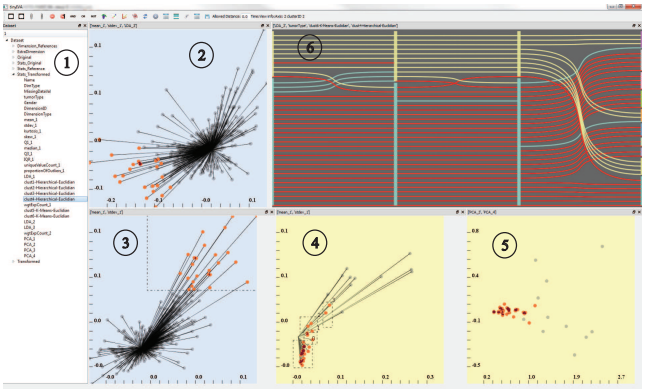

Figure 1: The dual analysis framework. 1) Data view for selecting different data variables and results 2,3,4) Deviation plots that depict changes in the statistics 5) Dynamic PCA view depicting PCA results applied on samples 6) Categorical data view showing different information on data, such as meta-data, clustering results. Notice the background color for views showing genes is blue and for samples it is yellow.

2.1 Statistical Analysis Tools

In this paper, we utilize a number of multivariate statistical analysis methods like principal component analysis (PCA), linear discriminant analysis (LDA), and, clustering. PCA is a widely used dimension reduction method [10]. The goal of PCA is to create a lower dimensional projection of the original data, while trying to preserve much of the variance. PCA creates principal components (*PC*) which are the axes of an orthogonal coordinate system. In microarray data analysis, PCA is usually used to filter genes prior to other statistical analysis operations like clustering or classification [28].

LDA is a supervised multivariate analysis method which tries again to find a linear combination of the original dimensions of the data [9]. Unlike PCA which is a supervised method, LDA considers class labels when computing the linear combination of dimensions. Therefore, LDA tries to maximize the class discrimination while reducing the dimensionality of the data. LDA is used as a classifier or as a dimension reduction method.

Clustering is the process of assignment of a set of similar data items into subsets — clusters. It is a commonly used method in microarray analysis to reveal the underlying structures and relations between genes and observations. In this paper, we use both hierarchical and k-means clustering using Euclidean distance for dissimilarity computations [23].

3. THE DUAL ANALYSIS FRAMEWORK

Our proposed analysis framework involves the iterative and parallel analysis of the genes and the samples. At each iteration, the user modifies the selections, observes the relations visually and runs statistical analysis tools on the current selections. The results of the tools are observed and analyzed together with the already available meta-data. The analyst continues the same procedure in a loop until sufficient insight is achieved. The proposed frame-

work contains conventional IVA views like scatterplots, histograms and three special views, namely the categorical data view, the dynamic PCA plot and the deviation plot. Additionally, we include a microarray heatmap to visualize the expression values. In order to achieve the parallel and linked analysis of genes and samples, all the views are capable of visualizing genes and/or samples. For demonstration, assume two scatterplots, where the first one visualizes genes and the second one the samples. In this setting, while a point in the first scatterplot represents a gene, a point in the second scatterplot represents a sample. In order to make the distinction between these views easier, we color the background of the views that visualize the genes with blue (Figure 1 - 2,3) and the views that depict the samples with yellow (Figure 1 - 4,5). The user decides interactively to visualize either the genes and/or the samples in any of the available views. All the views are linked and the selected genes and/or samples are highlighted in all the views. In Figure 1, a screenshot of the realization of our framework can be seen.

In the context of this work, we refer to the microarray (or gene expression) dataset as $n \times p$ matrix M, where $m_{i,j}$ represents the $i-$th gene expression level for the $j-$th sample. We assume that M contains expression levels of n genes taken from p samples. We denote the set of genes as $G = \{g_1, \ldots, g_n\}$ and the set of samples as $S = \{s_1, \ldots, s_p\}$. Also note that $|G| = n$ and $|S| = p$. Additionally, such microarray datasets comes with information about the samples and/or the genes that are referred to as meta-data. Examples of such additional information can be the known classification of the samples, diseases associated with the sample, the gender of the sample, etc. We denote such meta-data regarding samples with D^S, where each sample has a number of meta-data D^S_i associated. Similarly, we denote each single meta-data regarding the genes with D^G_i. Figure 2 illustrates the structure of the microarray data and the associated meta-data.

In our framework, we utilize a brushing mechanism which is based on *composite brushing* [15]. As we have two types of views in terms of the visualized data, i.e., genes and samples, we handle the brushes on genes and samples separately. A brush b represents a subset of items (genes or samples) that are selected through a view. Each brush is combined with the already performed brushes using a Boolean operator Op with $Op \in \{\cup, \cap, \neg\}$, where $\cup$ represents the union, $\cap$ represents the intersection and $\neg$ represents the not operator. As a result, a composite brush B is produced, which is computed automatically as the user continues to make brushes. Individual brushes b_i are merged into composite brushes B_i using the selected Op by $B_i = Op(B_{i-1}, b_i)$ starting with $B_1 = Op(b_0, b_1)$. For simplicity, we denote the final set of brushed genes with G_L and samples with S_L. It is important to mention that, the type of the operator can be selected interactively by the user.

3.1 Integrated Statistical Analysis

Statistical analysis tools play a crucial role in the explorative analysis of gene expression data. However, the results of these tools become harder to interpret and less reliable due to their black-box structure and their dependence on the initial parameters. In order to overcome these drawbacks, visual analysis methods aim to integrate the explorative capabilities of experts with the computational power of algorithmic tools on the basis of visualization and interaction [11].

We integrated the most widely used statistical analysis tools into the proposed visual analysis framework. These statistical tools are principal component analysis (PCA), clustering and linear discriminant analysis (LDA). Although, we demonstrate our visual analysis procedures using these tools, our framework is general enough to include different types of statistical analysis tools like support

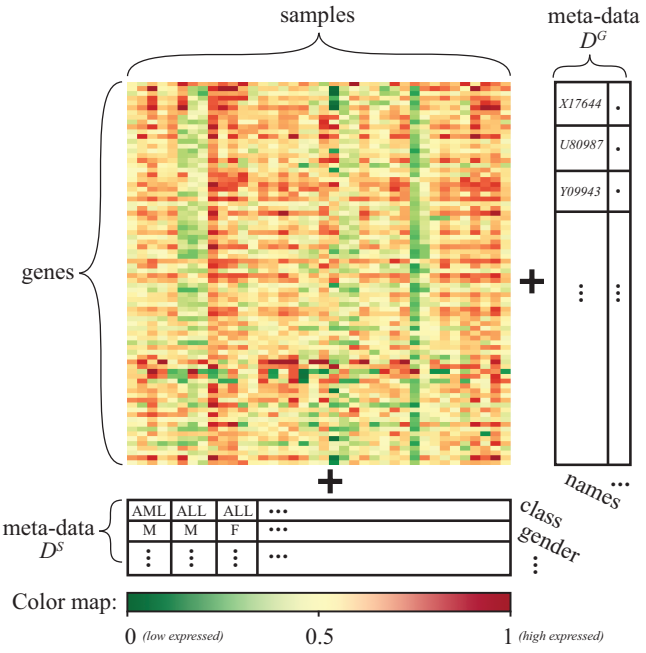

Figure 2: An illustration to depict the structure of the microarray data M. There are meta data columns associated with both the samples D^S and the genes D^G. The color map below is used to color the heat map visualization accordingly to the expression levels of the genes.

vector machines or artificial neural networks. In order to incorporate a wide range of statistical tools into our framework, we integrated the *R* statistical computation package into our system [24].

The integrated statistical analysis tools operate interactively on both the samples and the genes, i.e., it is possible to cluster the genes and the samples. The results of any operation, like clustering, PCA, or, LDA, are stored and available for visual analysis immediately. At each iteration of the analysis, the expert runs a statistical analysis tool on the genes and/or the samples. The D^G and D^S values are updated with the new results. For instance, when the user clusters the genes, the clustering results are appended as a new D^G and in the case of clusters of samples, they are included as D^S.

An important point to mention is that, when the user runs a tool, the results are computed only for the selected genes G_L and samples S_L. This enables the analyst to interactively select different regions of interest, apply algorithmic tools on the samples and compare the results.

Another mechanism to compare both the genes and the samples is to devise statistics in the analysis. There are a number of different statistics that can be employed to summarize and aggregate information on the data. Here, we denote these different statistics with λ. For the sake of simplicity in this paper, we limit λ to be either the mean or the standard deviation, i.e., $\lambda = \mu, \sigma$. In the analysis, we compute the statistics over both the genes and the samples. To exemplify, we can estimate the standard deviation σ of the gene expression values of a particular gene g_i over all the samples. Similarly, we can estimate the σ of the gene expression values for a particular sample s_i over all the genes.

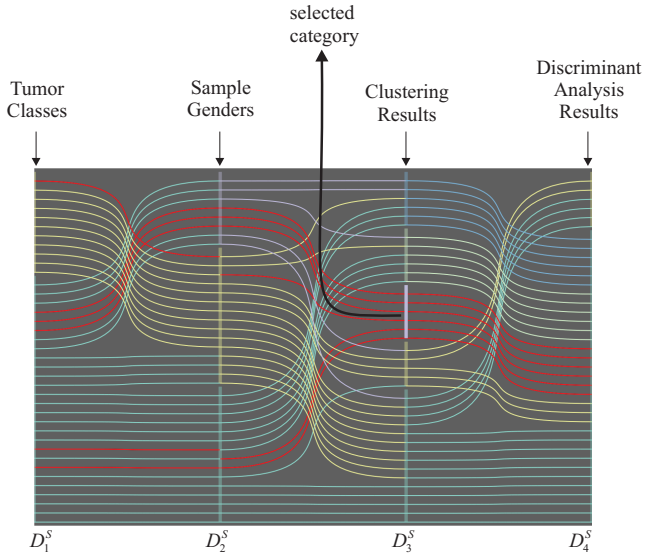

Figure 3: Categorical data view displaying different types of information. Each axis corresponds to a different source of information. Each curve that travels through the axes corresponds to a sample. The colored rectangles on each axis represent a category in the data associated with the axis. This view also enables selections of categories.

3.2 Categorical Data View

Our categorical data view enables the comparison of the meta-data related to the datasets with the results of the statistical analysis results. It is a modified version of the parallel cluster view [26] that we proposed in our earlier work. In this work, we extend the parallel cluster view in such a way that it displays both, the results of different statistical analysis tools, e.g., clustering, linear discriminant analysis, and, the categorical information about the samples/genes. In order to achieve this, we treat the results of each clustering and LDA result as categorical data. These results are visualized together with the meta-data. In other words, this view visualizes all the data columns which contain categorical data. With categorical data, we refer to data dimensions that consists of a limited, fixed number of values, where each value corresponds to a category value or a label.

We delimit the use of categorical data view only to the samples in this paper, since most of the meta-data is usually associated with the samples in microarray data analysis. In the categorical data view, each vertical axis corresponds to a column D_k^S, where k indicates the order of the column in the view, i.e., for the leftmost axis, $k = 1$ (Fig. 3). Each *rectangle* on an axis corresponds to a category (or a group) in D_k^S and each curve that travels through the axes represents a sample s_j.

All axes contain a set of categories (labels) where each category is represented by a different color. Curves between axis k and $k+1$ are colored with respect to the colors of the category they are members of in D_k^S. This coloring schema improves the comprehension of the membership relations between different columns. When one of the visualized columns is a clustering result, each rectangle on an axis represents a single cluster. In case of hierarchical clustering result, user sets a cut through the cluster hierarchy interactively.

Analysis view is integrated into the interactive analysis cycle by category level brushes and links to the selection mechanism in all other views. The user can select any number of categories, e.g., clusters, using a combination of Op operators. The selected items in the other views are also highlighted in this view. Further details on this view can be found in our previous paper [26].

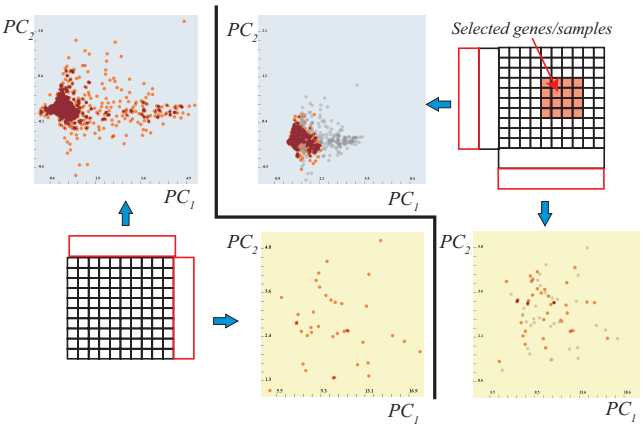

Figure 4: Illustration of dynamic PCA plot. It is possible to apply PCA interactively both on the samples and the genes. PCA results are computed using all the genes and all the samples. This operation can be seen as adding a new row or a new column to the dataset (left part). Dynamic PCA plots are updated automatically with the selections on the microarray data (right part). The updated results (computed using only the selected genes/samples) are overlayed on the previous PCA projections for comparison.

3.3 Dynamic PCA Plot

In our model, we introduce dynamic scatterplots to display PCA results. These plots are different than conventional scatterplots in terms of their focus+context (F+C) visualization. In conventional scatterplots, while the focus, i.e., the selected items, is highlighted in a more saturated colors, the context, i.e., the rest of the items, are visualized in a less saturated color. In order to enable the comparison of different PCA results, we modify this F+C visualization technique.

Dynamic PCA plot visualizes PCA results, where the genes or the samples are plotted using their projected values, e.g., projected onto the first two principal components. Contrary to the conventional scatterplots, it visualizes two different PCA results at a time. As its context, dynamic plots visualize the PCA results that are computed using all the genes and all the samples. And as its focus, it visualizes the PCA results that are computed using only the selected genes G_L and samples S_L. As the user updates a brush in any view, the PCA is computed interactively and the dynamic scatterplot is updated. In Figure 4 (the left part), we apply PCA on both the genes and the samples. The results are visualized as scatterplots that show the projection of the data to the first two principal components. When the user selects a new subset of genes and samples, the dynamic plot performs PCA analysis automatically in the background using only the selected genes and the samples. As a result, the focus of the plot highlights the new results and the context visualizes the results that are computed using the whole dataset as seen in Figure 4 (right part). The interactive computation of the PCA results enables the user to assess the impact of the current selection (on the PCA results) immediately. Since this plot depicts different projections of the data due to interaction, mechanisms to

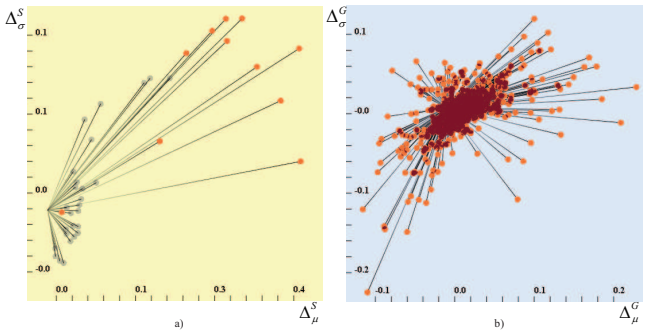

Figure 5: Deviation plots for the samples (a) and the genes (b). These views depict the changes in $\lambda = \mu, \sigma$ values w.r.t. the brushes.

assess the projections could be included as well. In order to derive more quantitative conclusions on the resulting projections, one can make use of quality measures. One such relevant measure is related to class consistency suggested by Sips et al. [21]. The authors propose measures to evaluate the data projections where class labels are present for the projected items. In such a way, the dynamic PCA projections could be evaluated quantitatively.

3.4 Deviation Plot

Another powerful tool to analyze microarray data is to visualize statistics that describes the genes and/or the samples. As noted earlier, we refer these statistics as λ in our framework. In order to achieve interactivity, we link the computation of the λ values with the selections on the items and the dimensions. To illustrate, assume that we estimate the σ value for a g_i using all the values for all the samples. Than, a subset of the samples is selected. When we estimate the new σ value for g_i, we perform this computation using only the selected samples. Such an operation yields a different σ value. The difference between the first σ value (computed over all samples) and the second σ (computed over only the selected samples) carries relevant information for the analysis.

Firstly, we consider the difference computation over the genes. We estimate the statistic λ for two arbitrary gene subset selections B^G_1, B^G_2 and compute the difference Δ^G_λ. This can be denoted as:

$$\Delta^G_\lambda = \lambda(B^G_2) - \lambda(B^G_1)$$

Here, we refer to $\lambda(B^G_1)$ as the *reference selection value* and $\lambda(B^G_2)$ as the *current selection value*. Similarly for the samples, Δ^S_λ denotes the difference value between two λ values (computed for the two sample subset selections). In order to facilitate these difference values in our analysis, we introduce the *deviation plot*. This view visualizes the Δ values associated with two different statistics computed between the two identical brushes. Figure 5 illustrates such views for both the samples (a) and the genes (b). In both of the visualizations, the x-axis represents the change in the μ values ($\lambda_1 = \mu$) and y-axis represents the change in the σ values ($\lambda_2 = \sigma$). We denote such scatterplots as $[\Delta^S_\mu, \Delta^S_\lambda]$ and $[\Delta^G_\mu, \Delta^G_\lambda]$ with respect to the displayed statistics.

Assume we have a deviation plot of the samples similar to Figure 5-a, where each point represents a sample. When there is no change in either the μ or the σ value for a particular sample, the resulting point is at the origin (i.e., no change between the two selections , $\Delta^S_\mu = 0$, $\Delta^S_\sigma = 0$). On the other hand, assume that the difference values are, for example, $\Delta^S_\mu = 3$ and $\Delta^S_\sigma = 5$. Then, we draw a point on $(x, y) = (3, 5)$ and connect this point with a line to the origin (0, 0).

One important aspect of the deviation view is that it always depicts the difference between the aforementioned reference value and the current value. The reference value can be set interactively at any stage in the analysis. In order to do this, the user determines a selection B_1 which can be used as a basis to compare to further selections. Then, the statistics are computed using B_1 and the result is set as *the reference value*. From this point on, when the user updates the current brush in the system, the Δ_λ value is computed according to this saved reference value. It is possible to have multiple and linked instances of these deviation views and each view can have a separate reference value. This mechanism enables us to save certain brushes for future comparisons. When multiple instances with different reference values (each corresponding to a different previous selection) are used, these views provide a history of previous steps in the analysis.

4. ANALYSIS PROCEDURE

In microarray data analysis, *marker* genes are the genes that characterize certain diseases or properties in the given samples.

One of the most common approaches used in the identification of such genes is to search for genes that have expression levels below or above a certain threshold. The genes are marked as *down-regulated* or *up-regulated* if they are below or above the predefined threshold. Such genes are then investigated for the production of specific proteins and therefore critical in understanding the development of specific behavior in different organisms. Such an identification phase is usually followed by a classification or grouping phase, where the analyst tries to find the gene or the sample groups which share common characteristics. Such groups then form the basis of diagnosis and the classification of certain diseases or properties. A very common and well-studied example for such analyses is the classification of tumors using gene expression data [2].

In this paper, we focus on a visual analysis procedure where the main goal is to find out sets of genes/samples that are important in determining the classification and the groupings of the samples/genes. Additionally, we utilize the available metadata, accompanying the expression values themselves, that classify the samples into different classes $C = \{C_1, \ldots, C_k\}$, where $\sum_i |C_i| = |S|$. These classes can stand for distinct parameters, spanning from different tumor types that the sample was obtained from, to the gender of the sample source (male or female). Such classifications provide useful selections to compute the *reference* or the *current* values in the deviation plot.

In this paper, we work on a gene expression dataset provided by Golub et al. [7]. Here, the samples are known to come from two types of acute leukemia, namely acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). The dataset consists of 7129 genes taken from 38 different tissue samples where 27 is known to be ALL and the rest AML, i.e., we consider two groups $C_1 \equiv ALL$ and $C_2 \equiv AML$. More formally, the corresponding data matrix M comes with $|G| = 7129$ and $|S| = 27$. Additionally, we load the known class labels (C_1, C_2) of samples into D^S.

As the first step, we apply unit scaling to the data (i.e., scaling to $[0, 1]$), so that the columns of M are comparable. There are extensive studies on the effects of normalization on gene expression analysis [27] and we prefer unit scaling as it is one of the preferred normalization methods in tumor classification studies [2].

As the second step, we estimate the two following statistics, $\lambda = \{\mu, \sigma\}$, over three different sample selections determined by the classification; i.e; $B(C_1) = C_1$, $B(C_2) = C_2$ and $B(\forall) = C_1 \cup C_2$. Essentially, we add six rows to the data table $\mu_S(C_1), \mu_S(C_2), \mu_S(\forall)$

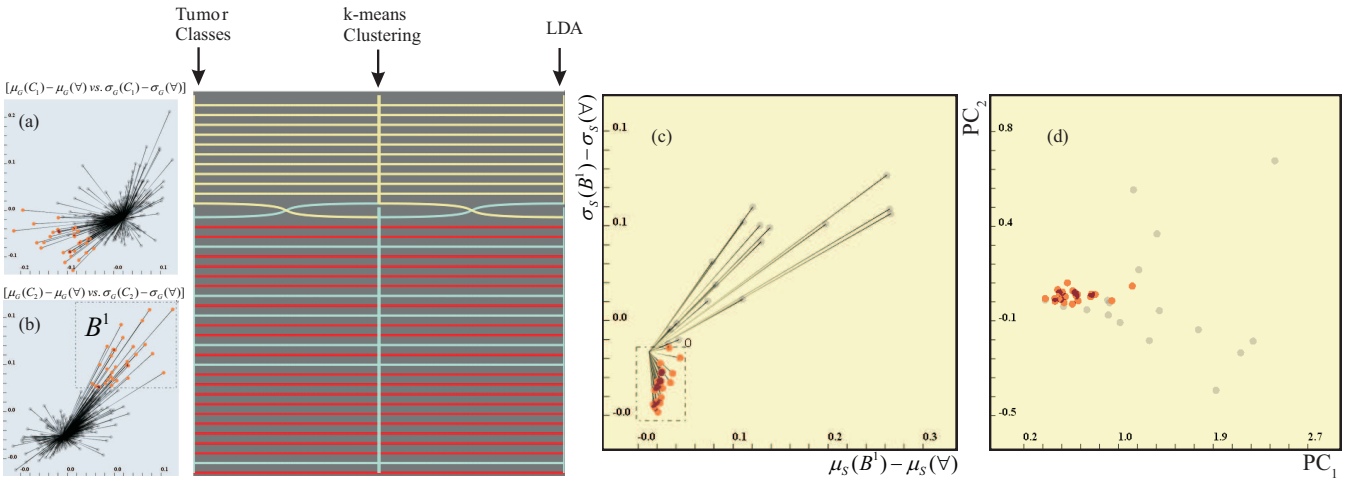

Figure 6: Through deviation plots (a) $[\mu_G(C_1)-\mu_G(\forall),\sigma_G(C_1)-\sigma_G(\forall)]$ and (b) $[\mu_G(C_2)-\mu_G(\forall),\sigma_G(C_2)-\sigma_G(\forall)]$, a user can spot genes that are significantly expressed for classes C_1 and C_2 respectively. After performing a brush selection in (b), having significant values with respect to class C_2, we perform k-means clustering and *LDA* and produce them into the Categorical View. The deviation plot on samples (c) shows statistics on the samples $[\mu_S(B^1)-\mu_S(\forall),\sigma_S(B^1)-\sigma_S(\forall)]$ while the Dynamic PCA plot displays projection of samples onto two principal components (PC_1, PC_2) computed only by means of the genes in brush (b). We brush a small group of tightly placed in samples in (c) and through the linked views, we observe the highlighted samples in the categorical view. The categorical view depicts the resulting classes in the clustering and the LDA results, together with the tumor classes in the leftmost axis.

and $\sigma_S(C_1),\sigma_S(C_2),\sigma_S(\forall)$. It is important to mention that any further statistics can be computed and subsequently utilized in such analyses.

Afterwards, we produce two Deviation plots on genes, where one depicting $[\mu_G(C_1)-\mu_G(\forall),\sigma_G(C_1)-\sigma_G(\forall)]$ (Plot A, Fig 6-a) and the other one showing $[\mu_G(C_2)-\mu_G(\forall),\sigma_G(C_2)-\sigma_G(\forall)]$ (Plot B, Fig 6-b). Here, we can easily observe that the genes that are significantly expressed with respect to both classes C_1 and C_2 respectively. Thus, we brush a set of genes in plot B that have significant values with respect to class C_2; i.e., producing the selection B^1. The selected genes are also highlighted in Figure 6-a, by the linking mechanism.

Subsequently we perform k-means clustering (Fig 6,middle axis) and *LDA* (Fig 6,right axis) on the genes in B^1, while showing them with tumor types separation in Categorical View (Fig 6, left axis). These two classification are used to analyze the appropriateness of the gene selection B^1. The corresponding heat map for genes in B^1 is depicted in Figure 7-a.

Then we produce two more views, one Deviation plot on samples showing statistics on the samples $[\mu_S(B^1)-\mu_S(\forall),\sigma_S(B^1)-\sigma_S(\forall)]$ (Plot C, Fig 6-c), and one Dynamic PCA plot to display the projection of the samples onto the two principal components (PC_1, PC_2) computed only by means of the genes in B^1 (Plot D, Fig 6-d). Importantly, by changing the gene brush B^1, we can interactively see the change in all the accompanying sample views that depict the statistics and/or PCA results. In Plot C, we observe a set of samples grouping together, which we then select by a rectangular brush to select a set of samples B_S. Here we can interactively observe how the current selection (B_S) relates to the real categories. In the categorical data view, we observe that the selected samples (B_S) are from the ALL type. However, there are still a number of samples from the ALL group that are not included in B_S. This result can then be improved further with different selections and using other computational tools. Accordingly, we perform another brush (in Plot A, Fig 8) to select also the significant genes for the class C^1. Here, we keep the previous brush B^1 and the selection is updated with the new selection B^2through the $\cup$ operator, i.e, $B^2 = B^2 \cup B^1$. As was mentioned above, the linked views (Plot C and Plot D, Fig 8) are automatically updated. Additionally, we update k-means clustering (Fig 8,middle axis) and *LDA* (Fig 8,right axis) for a set of genes B^2. The corresponding heat map showing just the brush in Plot A is displayed in Figure 7-b. Now, by selecting the class C^2 (Fig 8, left axis), we can notice that the clear separation of samples in the statistics based view (Plot C) is lost, since B^2 now contains mu-

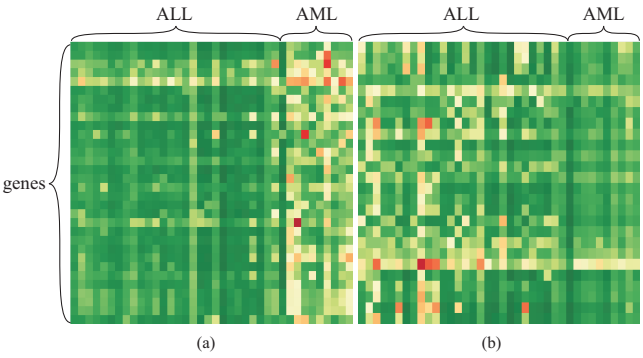

Figure 7: Gene expression levels through a standard heat map visualization, where genes being the rows and samples being the columns. The color map from Figure 2 is used to color-code expression values. The heat map for genes obtained by the brush B^1 (being significant for class C^2) (a) and the ones defined just by brush B^2 excluding B^1 (b) (being significant for class C^1). We can clearly see the separation of the expression values along all the samples (columns).

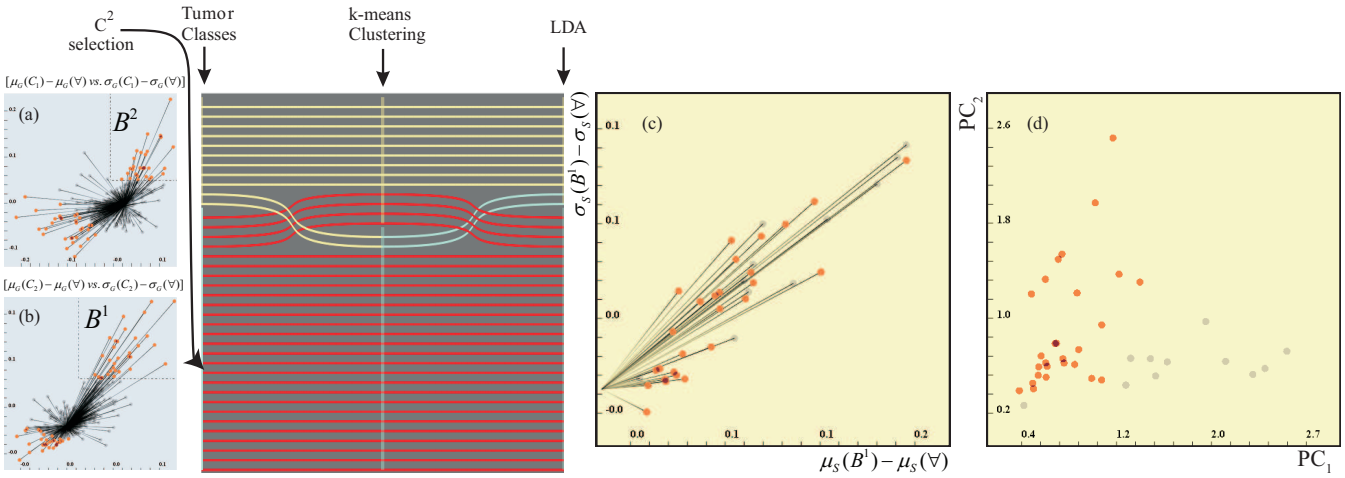

Figure 8: Continuation of the analysis from Figure 6. In Deviation Plot (a), we additionally brush (∪) the significant genes representing C^1; the plots C and D are automatically updated. We also recompute *k*-means clustering and *LDA*, and refresh the Categorical View. We notice that the clear separation of samples in plot C is lost, since the brush contains mutually significant genes representing both classes equally. However, Plot (d) reveals a grouping structure. We also brush the class C^2 in the Categorical View, where we observe a perfect separation of the samples into two groups as a result of *LDA* applied only on the brushed genes.

tually significant genes representing both groups equally. On the other side, Plot D reveals a good clustering of the samples through the updated PCA projection. As a major result, we found the selected genes B^2 are producing a perfect separation of samples into two groups by applying *LDA* (Fig 8,right axis), while utilizing of *k*-means is good but still not perfect (containing three more samples from C^1). This was achieved by selecting the class C^2 in the Categorical View (Fig 8).

In order to evaluate if the selected genes are biologically relevant, we evaluated the above selected gene set by comparing it to the results found in the reference study by Golub et al. [7]. The authors present an advanced, neighborhood and correlation based class predictor to distinguish different cancer types. In their paper, they determined 50 genes (out of 7129 genes) that are the most distinctive with respect to their algorithm. We compared the genes that are found in our analysis with their outcome. With the above selections in the two deviation plots (Figure 8-a,b), we select 89 genes in total. When we compared these selected genes with those in the paper by Golub et al., we observe that our selection contains 25 of these distinctive genes. Since Golub et al. are using more advanced computational methods to determine the distinctive genes, our selection contains only half of the listed genes. However, the overall result of our evaluation clearly shows that even with the use of basic descriptive statistics, our methodology results in a highly relevant subset of genes. Such a subset could easily be subject to more advanced and deeper analysis for a more exact result.

One important point to mention here is that our aim with the above evaluation is not to compare the predictive capability of our method with the other prediction methods. In fact, our goal is to improve such methods rather than to compete with them. We aim to aid and guide the analysts with the use of computational tools, such as the one suggested by Golub et al., and help them to interpret the outcomes much more easier.

The above process is the basic steps of the analysis that is possible with the proposed dual analysis scheme and the special views. All the above steps of the analysis can be performed iteratively in any order until sufficient insight is achieved.

5. CONCLUSION & DISCUSSIONS

In this paper, we introduced a dual analysis framework that enables the joint and the linked analysis of the genes and the samples through the use of integrated statistical analysis modules. We firstly described the integration of statistical tools into the IVA cycle. These tools always operate only on the selected samples and the selected genes, and their results are available for interactive analysis immediately.

We described a set of specialized views to use in the visual analysis of microarray data. These views are: i) categorical data view that compares the results of different statistical tools such as, clustering and LDA, together with the meta-data on the samples, such as the gender of the samples ii) dynamic PCA plot that performs PCA on the data interactively as the user modifies the selections iii) the deviation view that depicts the changes in the computed statistics due to interactive selections.

We provided the steps of analysis procedures that are possible with the introduced framework. The proposed analysis scheme is based on the iterative analysis of the data, where the user explores and refines the statistical analysis results at each iteration. Such an iterative process not only leads to more insights but also more reliable statistical results.

We observed that our framework brings new opportunities in generating and evaluating hypotheses in the analysis of gene expression data. With the tight integration of statistical tools in interactive visual analysis cycle, it is possible to overcome the black-box characteristics of these tools. This brings the possibility to produce reliable results that are easier to communicate.

The proposed framework is currently being developed and there are a number of possible future directions. We plan to integrate a larger variety of statistical analysis methods into the view and develop specialized views for the results of these methods. Possible methods can be listed as support vector machines, decision trees and factor analysis.

In this paper, we worked only on non-temporal datasets. However, microarray datasets with a temporal nature, i.e., acquired over a period of time, are also common. In order to extend our approach

to such datasets, we need to include views that are capable of handling temporal data, such as function plots.

Another possible future direction is the integration of information from different sources. Examples of such information are metabolic pathways, gene ontologies, and, more comprehensive clinical data. Additionally, we plan to extend our methods to longitudinal studies where the temporal domain of the datasets brings new challenges in analysis.

6. ACKNOWLEDGMENTS

The authors would like to thank Animesh Sharma from Department of Informatics, University of Bergen for providing the dataset and related information. The authors also thank Kjell Petersen from Computational Biology Unit, Bergen for the valuable discussions and ideas.

7. REFERENCES

[1] J. Dietzsch, J. Heinrich, K. Nieselt, and D. Bartz. Spray: A visual analytics approach for gene expression data. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 179–186. IEEE, 2009.

[2] S. Dudoit, J. Fridlyand, and T. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, 97(457):77–87, 2002.

[3] B. Dysvik and I. Jonassen. J-Express: exploring gene expression data using Java. *Bioinformatics*, 17(4):369, 2001.

[4] M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95(25):14863, 1998.

[5] N. Gehlenborg, S. O'Donoghue, N. Baliga, A. Goesmann, M. Hibbs, H. Kitano, O. Kohlbacher, H. Neuweger, R. Schneider, D. Tenenbaum, et al. Visualization of omics data for systems biology. *Nature methods*, 7:S56–S68, 2010.

[6] GENESPRING. Cutting-edge tools for expression analysis, 2011.

[7] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531, 1999.

[8] D. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang. ipca: An interactive system for pca-based visual analytics. In *Computer Graphics Forum*, volume 28, pages 767–774. Wiley Online Library, 2009.

[9] R. Johnson and D. Wichern. *Applied multivariate statistical analysis*, volume 6. Prentice Hall Upper Saddle River, NJ:, 2007.

[10] I. T. Jolliffe. *Principal Component Analysis*. Springer, second edition, Oct. 2002.

[11] D. Keim, G. Andrienko, J. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Visual analytics: Definition, process, and challenges. *Information Visualization*, pages 154–175, 2008.

[12] M. Kosorok and S. Ma. Marginal asymptotics for the "large p, small n" paradigm: with applications to microarray data. *The Annals of Statistics*, 35(4):1456–1486, 2007.

[13] A. Lex, M. Streit, E. Kruijff, and D. Schmalstieg. Caleydo: Design and evaluation of a visual analysis framework for gene expression data in its biological context. In *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, pages 57–64. IEEE, 2010.

[14] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg. Comparative analysis of multidimensional, quantitative data. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1027 –1035, 2010.

[15] A. R. Martin and M. O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *VIS '95: Proceedings of the 6th conference on Visualization '95*, page 271, Washington, DC, USA, 1995. IEEE Computer Society.

[16] J. Quackenbush. Computational analysis of microarray data. *Nature Reviews Genetics*, 2(6):418–427, 2001.

[17] O. Rubel, G. Weber, M.-Y. Huang, E. Bethel, M. Biggin, C. Fowlkes, C. Luengo Hendriks, S. Keranen, M. Eisen, D. Knowles, J. Malik, H. Hagen, and B. Hamann. Integrating data clustering and visualization for the analysis of 3d gene expression data. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 7(1):64 –79, 2010.

[18] A. Saldanha. Java Treeview – extensible visualization of microarray data. *Bioinformatics*, 20(17):3246, 2004.

[19] P. Saraiya, C. North, and K. Duca. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11:443–456, 2005.

[20] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results. *IEEE Computer*, 35(7):80–86, 2002.

[21] M. Sips, B. Neubert, J. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. In *Computer Graphics Forum*, volume 28, pages 831–838. Wiley Online Library, 2009.

[22] SPOTFIRE. Decision site for functional genomics, 2011.

[23] P. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining*. Pearson Addison Wesley Boston, 2006.

[24] R. D. C. Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. ISBN 3-900051-07-0.

[25] C. Turkay, P. Filzmoser, and H. Hauser. Brushing dimensions – a dual visual analysis model for high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2591 –2599, dec. 2011.

[26] C. Turkay, J. Parulek, N. Reuter, and H. Hauser. Interactive visual analysis of temporal cluster structures. *Computer Graphics Forum*, 30(3):711–720, 2011.

[27] Y. Yang, S. Dudoit, P. Luu, D. Lin, V. Peng, J. Ngai, and T. Speed. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic acids research*, 30(4):e15, 2002.

[28] K. Yeung and W. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763, 2001.