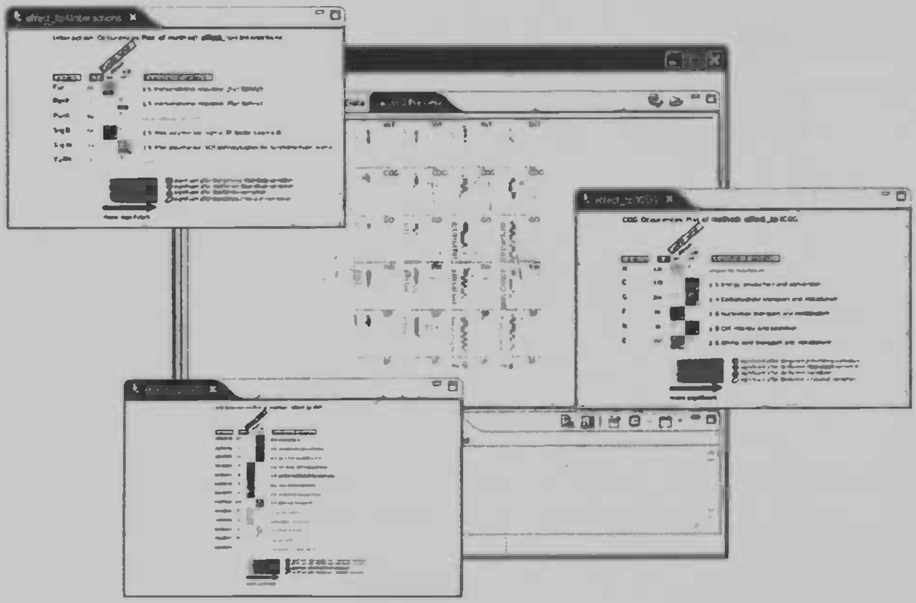
University of Groningen **Faculty of Mathematics and Natural Sciences**

**Department of
Mathematics and
Computing Science**

Design and evaluation of a visualization application for the analysis of transcriptome data

Lars Tijsma

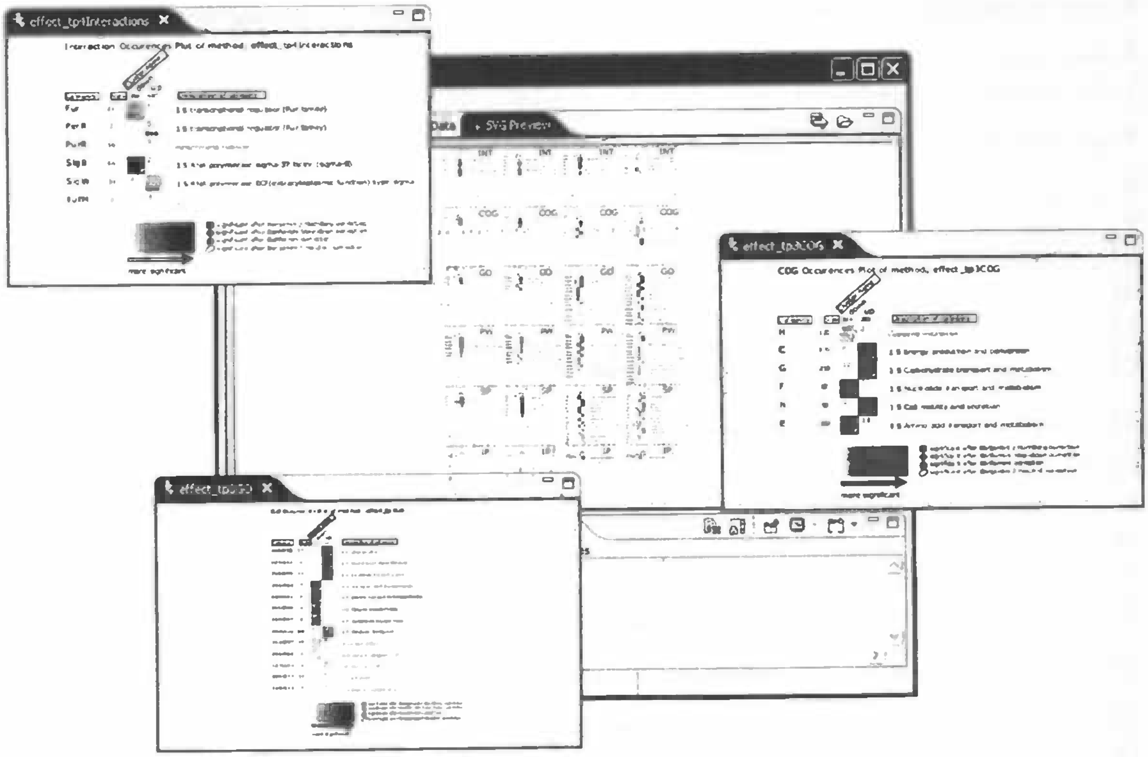
June 14, 2006



RuG

Master's Thesis

Design and evaluation of a visualization application for the analysis of transcriptome data



Lars Tijsma

Institute for Mathematics and Computing Science in cooperation with the Department of Molecular Genetics

University of Groningen

The Netherlands

Supervisors: Evert-Jan Blom, Dinne Bosman, Jos Roerdink and Oscar Kuipers

Wednesday, June 14, 2006

Contents

List of Figures

Abstract

Data from DNA micro-arrays provide biologists with huge amounts of information on gene activity. Following statistical analysis of the raw gene expression values, genes can be grouped in clusters exhibiting the same expression patterns. Relating the cluster information to known biological processes by hand is tedious and error-prone. To assist biologists in analyzing these data, the computer application FIVA (Functional Information Viewer and Analyzer) was developed. The application combines cluster information with information on known biological processes and generates colormaps showing the obtained functional profile.

FIVA was developed from a biological point of view and was in need of a redesign. To guide us in creating a new design and implementation of FIVA, we used known concepts from the field of information visualization like the visualization mantra about interaction, eight guidelines on multiple views and color gradient interpretation information. The visualization in the new version of FIVA uses two different view types. One view displays an overview of all information, the other view shows the details a user has selected.

To test whether the new implementation of FIVA satisfies real-life users, it was subjected to a usability study with ten specialized biologists as participants. We used a think-aloud protocol with coaching to study the interaction of the participants with FIVA when solving a set of real-world biological problems. After completing the tasks, participants were subjected to an interview.

All participants stated that they were satisfied and were planning to use the application in the future. The main usability problems include the lack of an adequate filtering option, inconsistent button locations, the colors that are used in the visualization and navigating through the overview diagram. The analysis of these usability problems showed that some problems could be traced back to violation of two multiple view guidelines and providing too little support for a part of the visualization mantra. The solutions to usability problems that could quickly be solved were implemented after the usability study.

Chapter 1

Introduction

New technologies provide extraordinary – almost supernatural – powers to people who master them

- Ben Shneiderman, 1998

One could argue that scientists should be very grateful for what computers can do for them. Computers can generate amounts of data so high that it would take thousands of scientists many years, if ever, to get hold of the same amount. New scientific fields are emerging from existing fields thanks to this capability of the computer. On the other hand these amounts of data may be so large that analysis of it takes many tedious hours of bookkeeping. To overcome this problem we need not only have the computer generate data, we also want a computer to visualize the data in such a way that it can assist scientists in analyzing and understanding the data.

One such area which is clearly in need of user-friendly data visualizations is the bioinformatics research field of DNA micro-array data analysis. DNA micro-array experiments provide information on the activity of thousands of genes in an organism. Relating these activities to known biological processes is a task which can highly benefit from computer applications that provide supporting visualizations.

In the Molecular Genetics department of the R*u*G biologists developed an application that can assist in relating genomic activity to known biological process by providing a visualization which is called FIVA (Functional Information Viewer and Analyzer). FIVA is designed to allow biologists to get an overview of micro-array data and give insight in details much faster than analyzing the data by hand. Since FIVA was developed from the biology point of view only, it could benefit from improvements inspired by concepts from the field of information visualization.

This thesis presents a report of the FIVA redesign project. In this project we have gathered general concepts from the field of information visualization and based a new design of FIVA upon these concepts. Besides creating a new implementation of FIVA, the project also encompassed a usability study to test whether the new FIVA was usable for the specialized biologists it was designed for.

1.1 Project objectives

The goals of this project were governed by two different interests: a practical and a scientific one. First we wanted to improve an application so biologists had an easier time using it. Second we wanted to find some guidelines that govern effective visualizations. To satisfy all interests the project included the following work:

- Understand which kind of biological data the existing FIVA visualizes.
- Study general concepts from the field of information visualization and usability engineering.

- Translate the general concepts in practical improvements for FIVA and implement these improvements.
- Perform an evaluation study on the improved FIVA using biologists that have been using or are going to use FIVA.

1.2 Related work

Currently there are many related programs developed that assist in relating transcriptomic data to biological processes. These include GoSurfer, FatiGO, GoFish, GOMiner, GenMap/MAPPFinder, David, GeneMerge and FuncAssociate [8]. Each program uses known information on biological processes in different type of organisms to assist the analysis of gene activity in that organism. The used biological information and the visualizations differ from program to program.

General information on the field of information visualization can be found in [19]. This book is a perfect starting point for those who want to delve in this new field of science. A complete overview of user interfaces and related concepts can be found in [16]. An overview of usability engineering methods can be found in [11]. Concrete questionnaires are listed in [16]. For a completely developed evaluation study refer to [14].

In this thesis the multiple view concepts are extensively used. A complete overview of this visualization technique can be found in [6]. An example study on colormaps and color gradients and relating concepts to applications is given in [5].

1.3 Thesis structure

In the next chapter we will start by providing a background on DNA micro-array experiments. Chapter 3 will give an explanation on the visualization FIVA uses. In chapter 4 we will discuss some information visualization concepts which are related to visualization aspects of FIVA. Chapter 5 presents the design of the new FIVA application. We will explain in this chapter how the concepts from chapter 4 are translated to concrete design decisions. In chapter 6 we will give some implementation details and explain which tools and techniques we used to realize the design. Chapter 7 deals with the evaluation study performed on the new FIVA application. We will present the experimental setup and the used usability methods. The results of the experiment and a discussion of these results is also present in this chapter. Finally in the last chapter, number 8, we draw some conclusions and provide remarks and suggestions for further research and development on FIVA are given. The appendices provide some additional information on the implementation of FIVA. Also all forms that were used during the usability tests are presented.

1.4 FIVA versions

Because both the old version and the new version of the application are called FIVA readers might get confused which version is meant. On the other hand, it would be a bit tedious to explicitly state which version we mean. Therefore, we have decided that in chapter 2 to 4 when we mention FIVA we mean the old version unless stated otherwise. From chapter 5 to chapter 7 we mean the new FIVA version.

1.5 Post study development

After the usability study was performed, some additional development work was done on FIVA. This development resulted from the usability study and remarks given by external scientists. Because of the additional work, some parts of the description of the visualization of the new FIVA version have become obsolete. We have decided not to update the description because the usability study is based on the version that is presented in this thesis. The latest developments are mentioned in the conclusion.

1.6 Website

The application's website is located at http://bioinformatics.biol.rug.nl/websoftware/fiva/. Here you will find the latest version[1] of the application available as a self-installing package for Windows and as archive for Windows, Mac, different Linux and UNIX versions. Earlier versions are not available. A link to required software can also be found on the website. The site also contains an installation guide, different tutorials and a FAQ section. Finally, some databases with genomic data for FIVA are available.

[1] By the time of writing the latest version of FIVA is 1.0.0

Chapter 2

DNA Micro-array Background

The late 20th century witnessed the rise of a new tool for measuring gene activity, called gene expression analysis, on a large scale. This tool is called DNA micro-arrays[2] (or just micro-arrays, or microchips). Picture a rectangle containing thousands of colored circular spots and you have an idea of what a DNA micro-array looks like. Each spot represents a unique nucleotide sequence of the gene of the organism that is studied. Micro-arrays can provide highly detailed information about the expression of every single gene in an organism ranging up to about 50.000 genes. Prior to the micro-array era, gene expression data were obtained on a single-gene-at-a-time basis, making it much more time consuming than micro-array analysis.

This chapter will describe micro-arrays experiments and the possible analysis of DNA micro-array data.

2.1 DNA Micro-arrays experiments

Micro-arrays can be used for a wide variety of experiments, which can be categorized in a few distinct groups [12]. In spite of this variety, the experimental flow and analysis of the results are very similar. Figure 2.1 depicts the basic stages of an experiment[3]. These stages will be used as a guide for the explanation below.

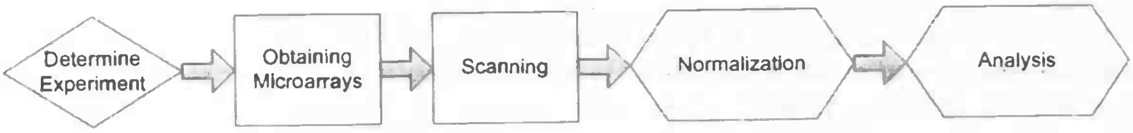


Figure 2.1: Basic stages of a DNA micro-array experiment

Determine experiment

Before starting a DNA micro-array experiment, decisions about the experiment must be made. An important decision is determining which populations of cells will be compared. For simplicity, we will use examples with only two populations. Thus, the goal of the experiment in this example is obtaining the expression ratio of the genes in the two populations.

Obtaining micro-array data

The first step in obtaining the data is isolating the mRNA molecules from the cell populations[4]. Next, the mRNA sequences are converted into cDNA sequences and

[2] Micro-arrays can also be used for the analysis of proteins, in which case they are called protein micro-arrays. Protein micro-arrays are a bit different from their DNA counterparts. Because the project described in this thesis is based on DNA micro-arrays we will not discuss the protein ones any further. See chapter 9 from [18] for more information.

[3] The stages are a bit simplified. More details can be found in [12].

[4] Actually this part of the process is very different for the different type of experiments

labeled with two different fluorescent dyes. The cDNA sequences from one population are labeled with a green colored dye, the sequences from the other population with a red colored dye. The cDNA sequences are then combined in equal amounts and the mixed cDNA is placed on the DNA micro-array. The cDNA that was present in the mix will hybridize to the spots on the micro-array containing the cDNAs complementary DNA. Finally, the unbound cDNA is washed off and the micro-array chip is ready to be scanned.

Scanning

To detect the bound cDNA the micro-array is scanned with a red and a green laser. The green laser excites the green-labeled cDNA and the red laser excites the red-labeled cDNA. The results of the two scans are stored in two different image files. An image file contains spots of different brightness, which are colored green or red. A brightly colored spot indicates that high levels cDNA of that particular type occur at that spot DNA, indicating that the gene responsible for creating this cDNA is highly active.

Once the data have been scanned it is exported as a digital image file[5] and converted to an ASCII dataset file by dedicated software. All colored spots are now converted to numbers. Because we are interested in the expression ratio of the genes from the two populations and not the absolute expression values, the files are merged into one file. The merged file consists of numbers representing the expression ration of the two populations. Once all individual experiments are scanned separately the character dataset files are combined into one big table, which is used for analysis.

Normalization

The obtained micro-array data is by no means perfect. The data files do not only contain the biological variation one is interested in, but also variation from other sources (like dye and spotpin effects). Actually, these other sources of variation vastly outweigh the biological variation [12]. Therefore, the next step is the correction of the obtained data for all sources of variation. This process is called normalization[6].

Analysis

A common method is to enter all data in a text editor or spreadsheet program and check which genes are differently regulated. Using a spreadsheet can actually be quite insightful, because of its ability to sort the data making the most dramatic effects readily observable. Many techniques exist to assist the mining of these large amounts of data. One of these techniques is clustering of expressions[7]. This technique involves grouping of different expression values, reducing the data dimensions significantly. Different expression values, for example, can be reduced to up-regulated, down-regulated and not regulated. Genes from one population are up-regulated if their expression values are

[5] A commonly used file format is the Tagged Image File Format, known as TIFF.

[6] Because there are so many sources of variation the term normalization refers to many different correction methods. An explanation of all these methods is beyond the scope of this thesis. Refer to [12] for a complete description.

[7] There are many more techniques in existence. Refer to [12] for a complete description.

higher than the expression values of the other population. The genes are down-regulated if their expression value is lower.

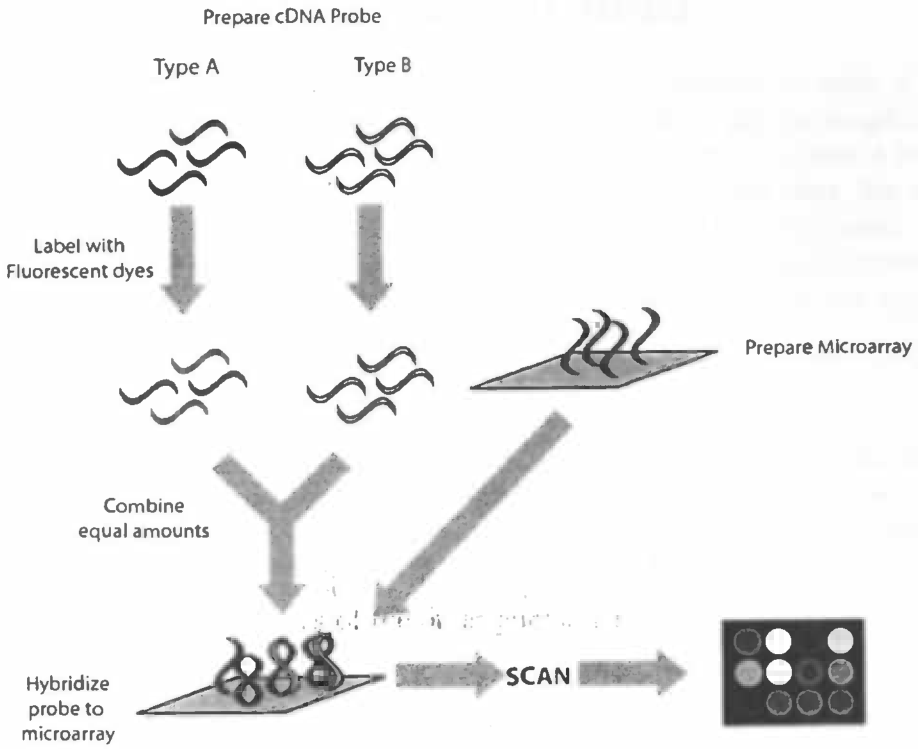


Figure 2.2: Overview of a common dual-channel micro-array experiment comparing two populations of cells

2.2 Functional information

A lot of information on function and behavior of genes already exists from before the DNA micro-array age. This functional information, as it is often called, is categorized in different functional modules. Each functional module is composed of different functional groups, also known as categories. In a functional group, clusters of genes exhibiting similar expression patterns are grouped together and are labeled with the name of the biological function they perform. Some functional modules consist of a few functional categories representing high-level biological functions. Other modules, on the other hand, consist of many low level functions.

The DNA micro-array data is often analyzed with this known functional information in mind. This analysis involves checking how many genes showing a particular expression pattern are also present in a functional group, thus reducing the dimensionality of the dataset considerably. In short, the obtained micro-array data is combined with known information from the literature[8] to determine which biological functions are affected during the experiment. A general purpose software tool, like a spreadsheet

[8] Unfortunately, a lot of functional information can still only be found in scientific publications. Specialized databases do exist but they often differ in vocabulary.

program, is often used for this analysis. The drawback of using non-specialized software, is that interesting data may be overlooked. This is where FIVA comes in.

Chapter 3

FIVA background

Inspecting huge amounts of numbers in a spreadsheet program is today a common form of DNA micro-array data analysis. Although this method can be insightful, in an age of advanced interactive visualization application development, it seems a bit tedious and error prone[9]. To aid the biologist in analyzing the micro-array data, the computer application FIVA (Functional Information Viewer and Analyzer) was developed.

In this chapter, we will give a conceptual overview of FIVA. We will explain what it can do, show a program flow and describe the inputs and output of the application. However, we will not delve into implementation details.

3.1 Overview

FIVA greatly enhances the analysis mentioned in the last chapter by automating the classification of genes from an experiment in functional groups. FIVA takes genome information from an organism with functional modules and data from a DNA micro-array experiment as input. With this information, FIVA creates a functional profile. This profile consists of multiple colormaps, showing which genes are overrepresented[10] in which particular functional group.

3.2 FIVA procedure

The micro-array data analysis in FIVA can be divided in four stages, which are shown in Figure 3.1. The stages will be described below.

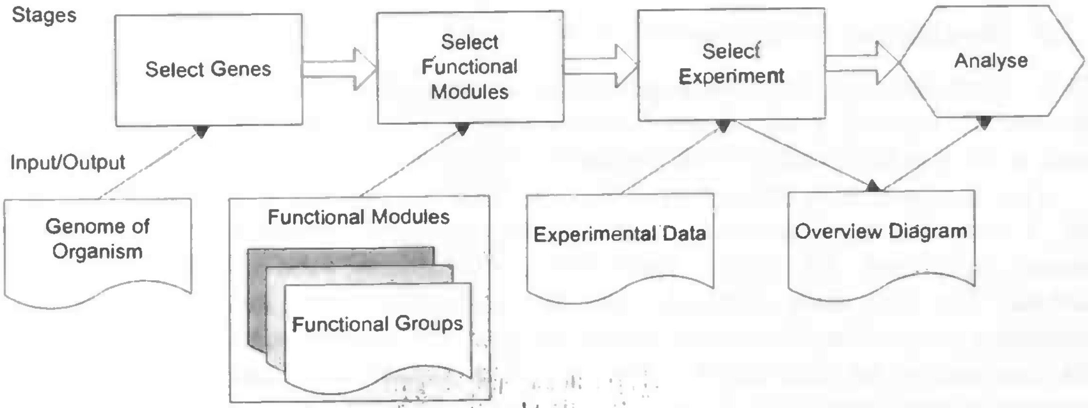


Figure 3.1: Stages in a FIVA analysis. The upper row shows the actual stages. The lower row shows the input and output accompanying each stage

[9] Although most information specialists agree that this is the case, a lot of biologists are quite happy with their methods in analyzing the data.

[10] Overrepresented indicates that a significantly number of genes that exhibit a particular expression pattern are also present in a single functional category

Select Genes

The first step is loading the genes of organism that need to be analyzed. This is done by loading a file containing a list with the names of all genes of the organism.

Select Functional Modules

The next step is loading the functional modules that are going to be used in the analysis. A functional module file contains a list with the names of functional groups in the module combined with the names of the genes that form that functional group.

Select Experiment

After the database of functional groups and genes is assembled, the data of a DNA microarray experiment must be loaded. The experimental data contain the following information: the name of the effect that was studied in the experiment, all genes from the studied organism, and the observed behavior of every gene. The experimental data is not limited to one effect; data from different experiments on the same organism can be combined in a single experimental data file.

Analysis

Taking the loaded information from the previous stages FIVA produces different colormaps. All colormaps show which genes in which functional categories are overrepresented in which clusters. A colormap is created for every functional module in every experiment. The analysis consists of looking at the colors in the different colormaps and drawing conclusions.

3.3 FIVA Visualization

3.3.1 Significant occurrences

FIVA identifies and displays significantly overrepresented functional categories in clusters. To identify a significant functional category, the distribution of a cluster of genes in the functional category is compared with a reference distribution.

This comparison is realized by a Fischer-exact test [4]. The drawback of this test is that it generates false positives. This means that some occurrences are erroneously marked significant. To counter these errors, corrections were calculated using three methods. The first, most stringent, correction is the Bonferroni method. If a significant occurrence passes the Bonferroni correction than the Fischer-exact test rightly marked this occurrence as significant. However, this conservative test may reject some occurrences that were rightfully marked significant. Therefore, all significant occurrences are also corrected by the Bonferroni step-down and the False Discovery Rate corrections. Occurrences that pass these corrections but fail the Bonferroni test are marked as possibly significant. Only the occurrences that fail all three corrections are marked as false positives [8].

3.3.2 Colormap description

Next, we will explain how the information about significantly overrepresented categories is displayed in a FIVA colormap. As seen from the last section we want the colormap to show multiple information about the significance of overrepresented categories simultaneously. The colormap should show which categories are significant as determined by the Fisher-exact test. Moreover, we also want the colormap to show which of these categories are surely significant, possibly significant and not significant, as determined by the three corrections.

An example colormap is shown in Figure 3.2. The colormap should be read as a matrix with rows and columns. The squares in the same columns contain information on genes with similar expression. The squares in the same rows contain information on genes in the same functional category.

The fill color of a square represents the significance of occurrence as determined by the Fisher-exact test. A white fill represent a low level of significance whereas a black fill represents a high level of significance. Some squares have a colored stroke. The color of this stroke represents the result of the corrections. A purple stroke represents occurrences that are certainly significant. A green stroke (not shown) represent occurrences that are possibly significant. A square with a colored fill indicating a significant occurrence but no stroke represents a false positive.

Next, we will examine the other attributes of the colormap, starting with the top of the image.

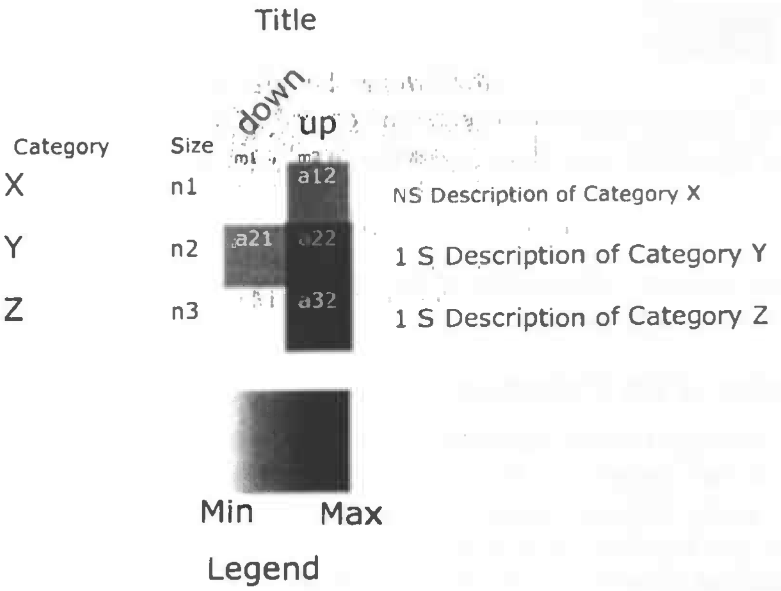

Title
down
up
Category
Size
m1
m2
X
Y
Z
n1
n2
n3
a12
a21
a22
a32
NS Description of Category X
1 S Description of Category Y
1 S Description of Category Z
Min
Max
Legend

Figure 3.2: Example FIVA colormap showing three functional categories (x, y and z) and two expression values (down and up).

Title

Located at the top of the colormap, the title contains the name of studied effect in the experiment and the name of the functional module.

Expression values

Located directly below the title, the name of the expression value of the genes in the column below is given. In this section, expression values are also referenced as clusters and in this example, the expression value of a cluster is either up-regulated or down-regulated.

Category

Found at the left of the plot. Category stands for functional category. The names located in the category column are the names of the different functional categories in the functional module. In this example, functional categories X, Y and Z are present.

Size

All blue numbers close to the colored squares represent sizes of different groups. The numbers to the right of the functional categories (n1, n2, n3) represent the number of genes in that category. The numbers below the expression values (m1, m2) represent the number of genes with that expression value.

Description

Left of the colored squares the description of the row's functional category is located. This gives a short description of the functional category. The number and capitals in front of the description denote how many squares are significant (S), possibly significant (PS), and non-significant (NS).

Gradient

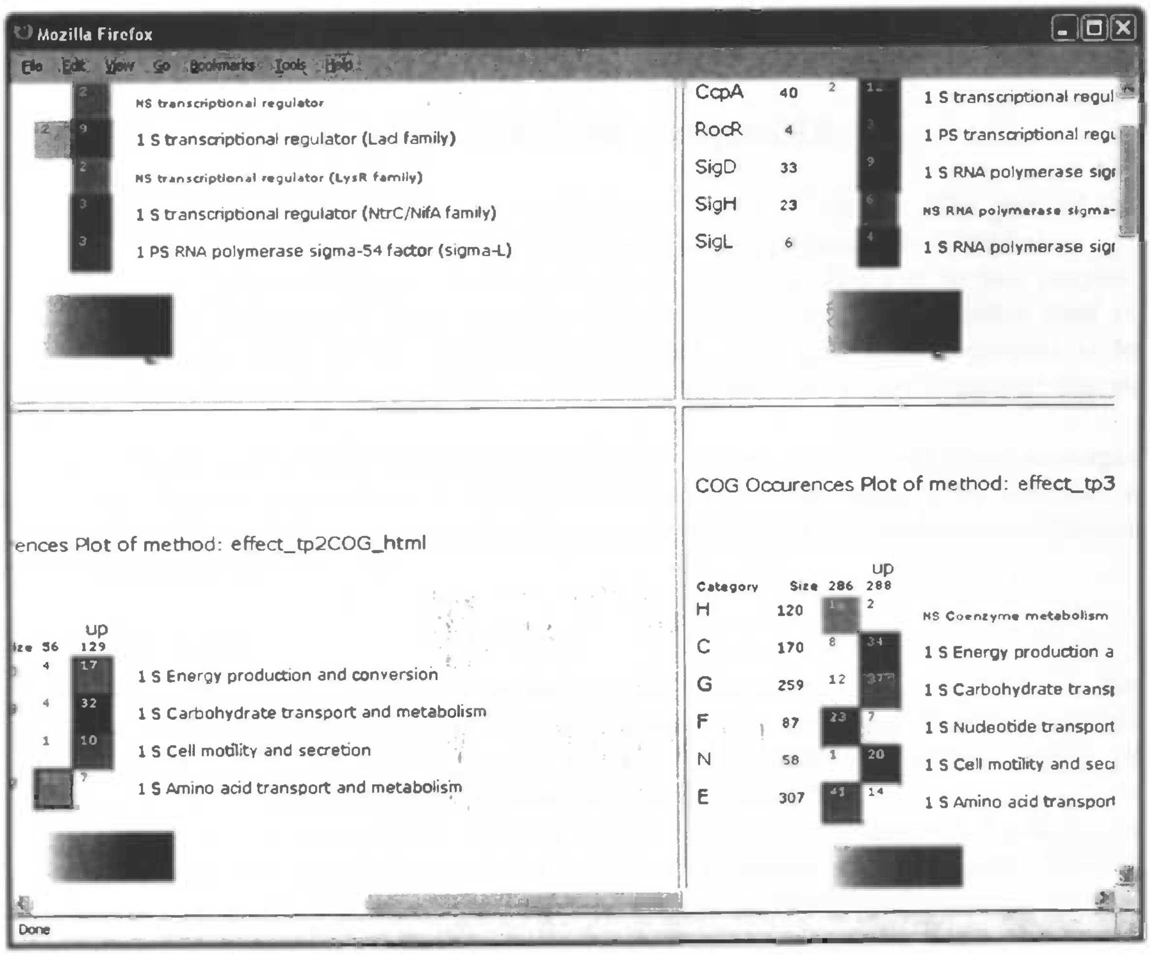
The gradient is located below the colored squares. This gradient shows which color stands for most significant and which color for the least significant.

Legend

Most colormaps contain a description of the experiment they represent. This description is located at the bottom of the map.

3.3.3 Overview of the Colormaps

The colormap presented above represents the regulation data of one experiment combined with the functional categories from one functional module. Since FIVA creates a colormap for every selected functional module and every selected experiment. All colormaps are put together in a large overview diagram. The rows of the overview diagram contain the colormaps with the same functional modules. The columns contain the colormaps from the same experiment. All information is presented in full detail and no resizing is performed. In all but a few cases will the diagram be completely visible on a computer monitor. Figure 3.3 below shows a screenshot of the overview diagram presented in a browser application.



NS transcriptional regulator
1 S transcriptional regulator (LacI family)
NS transcriptional regulator (LysR family)
1 S transcriptional regulator (NtrC/NifA family)
1 PS RNA polymerase sigma-54 factor (sigma-L)

CcpA	40	2	1_	1 S transcriptional regul
RocR	4		3	1 PS transcriptional regu
SigD	33		9	1 S RNA polymerase sigr
SigH	23		6	NS RNA polymerase sigma-
SigL	6		4	1 S RNA polymerase sigr

ences Plot of method: effect_tp2COG_html

ize	56	up 129	
	4	17	1 S Energy production and conversion
	4	32	1 S Carbohydrate transport and metabolism
	1	10	1 S Cell motility and secretion
		7	1 S Amino acid transport and metabolism

COG Occurences Plot of method: effect_tp3

Category	Size	286	up 288	
H	120	1	2	NS Coenzyme metabolism
C	170	8	34	1 S Energy production a
G	259	12	37	1 S Carbohydrate trans
F	87	23	7	1 S Nucleotide transport
N	58	1	20	1 S Cell motility and sec
E	307	41	14	1 S Amino acid transport

Figure 3.3: Screenshot of the overview diagram viewed in a web browser

Chapter 4

Visualization Aspects

Information visualization research is a fairly new area of science. The goal of the research is to discover visualizations of abstract data that are perceptually effective.

In the previous two chapters, the background of FIVA was laid out. In this chapter, we will take a closer look at some concepts from the information visualization field of science and relate them to the visualization of FIVA. The goal of this chapter is to provide information visualization concepts on which we can base a new improved design of FIVA.

The chapter is divided into two parts. In the first part, we present some concepts regarding the interaction with the FIVA visualization. In the second part, we will take a closer look at the presentation of the visualization used in FIVA and analyze the different information elements.

4.1 Interaction

In [16] all visual design guidelines regarding user interaction are summarized in the following-visual-information-seeking mantra: *Overview first, zoom and filter, then details on demand.* A refinement to this mantra is that the all of these activities should be supported by a good visualization. Not necessarily in that order [19].

In the current version of FIVA, nothing of this mantra is implemented. As shown in chapter 3, the FIVA visualization does not contain an overview or zooming and filtering options. The FIVA diagram only provides a large diagram containing all details at once.

Different techniques exist for implementing this mantra in visualizations. The first is presenting a complete overview scaled down to fit on a screen and provide options for zooming. Different zooming techniques exist. One can use distortion techniques like the table lens, or the fish-eye view [19].

Another technique for implementing this mantra is the use of multiple views. Ware [19] argues that when an overview is too complex for a user to hold in their visual memory, multiple views are more effective. Next, we will explore the multiple view concepts a little further.

4.1.1 Multiple views

A multiple views system is a system that uses two or more distinct views to display a single conceptual entity. Nowadays one can find countless examples of systems using multiple views one way or the other. A system using multiple views can offer benefits like improved user performance, discovery of unforeseen relations and unification of views on the desktop [6]. On the other hand, these systems are difficult to design and design mistakes are often made. To help the design of multiple views systems Baldonado et al. have identified a set of guidelines. These guidelines address issues concerning when to use multiple views and how to use them.

4.1.2 When to use

In the study of finding defining guidelines concerning multiple view systems [6] four rules are identified concerning when one should use multiple views in a system.

- Diversity
- Complementarity
- Decomposition
- Parsimony

Diversity

Use multiple views when there is a diversity of attributes, models, user profiles, levels of abstraction, or genres.
This rule is applies to the FIVA overview diagram because it contains different levels of abstraction. The overview contains information about the functional profile. Zoomed in it provides information about the significance of each functional category and the activity of individual genes in the categories.

Complementarity

Use multiple views when different views bring out correlations and/or disparities.
The FIVA overview diagram is just a collection of colormaps. By comparing the colormaps in the overview, one can extract pieces of information one cannot extract by inspecting a single colormap. Still a FIVA overview diagram offers no support for detecting correlations and disparities among the individual colormaps. A different view can assist in extracting such information.

Decomposition

Partition complex data into multiple views to create manageable chunks and to provide insight into the interaction among different dimensions.
Since the overview diagram presents a large amount of information at the same time, especially when a large number of time-points is present, it might become overwhelming. Extra views can be used to isolate different colormaps, thereby reducing the amount of data a user needs to analyze at the same time.

Parsimony

Use multiple views minimally.
Systems using multiple views are more complex than systems using a single view. Using to many views can cancel out the benefits of using more views.

4.1.3 How to use

Apart from defining four rules guiding the decision to implement multiple views, [6] also identifies four rules stating how multiple views should be implemented in a system. They are:

- Space/time resource optimization

- Self-evidence
- Consistency
- Attention management

Space/time resource optimization

Balance the spatial and temporal costs of presenting multiple views with the spatial and temporal benefits of using the views.
This rule states that different views should be presented to the user in such a way that switching to different views and reorganizing views should take as little time as possible.

Self-evidence

Use perceptual cues to make relationships among multiple views more apparent to the user.
This rule states that the contents of the views and the types of the views in the system should be easily be recognizable so that they are easily identified

Consistency

Make the interfaces for multiple views consistent, and make the states of multiple views consistent.

Attention management

Use perceptual techniques to focus the user's attention on the right view at the right time.
Because in a multiple view system different events can occur in different views, the user should be guided by the system to the most important view.

4.2 Presentation

A FIVA colormap contains many different visual cues. As mentioned in the previous chapter each cue gives a different kind of information about the significance of occurrences. The different visual cues can be globally divided in three categories.

- Colored fill
- Colored labels
- Spatial Layout

Colored fill

FIVA maps the significance of occurrence to a color gradient consisting of five base colors. These colors together with their RGB (Red Green and Blue) values are listed in Table 4.1.

Color	Color name	Red (0 – 255)	Green (0 – 255)	Blue (0 – 255)
⬜	White	245	250	255
🟪	Pink	245	120	255
🟧	Orange	245	120	1
🟩	Green	1	120	1
⬛	Black	1	1	1

Table 4.1: Colors from which the color gradient is built

Using these colors as a reference the gradient presented in Figure 4.1 can be built.

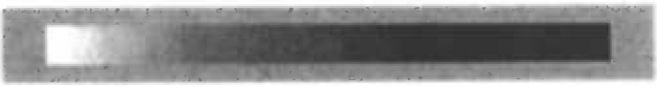


Figure 4.1: Color gradient used in FIVA

The task of these colors is steering the user to interesting categories and giving a measure of significance in relation to the other categories. Whether these colors are best suited for this task is unknown. Some alternative gradients are given in Figure 4.2 below. The rainbow color gradient depicted in (a) is the one that is most commonly used. This color gradient has the disadvantage that it is conceived as being organized in discrete regions, each represented by one of the rainbow colors [7]. When choosing a color gradient one must also consider the background and expectations of the intended user [5]. Since the users will most probably all be DNA micro-arrays experts they will be most familiar with the gradients (b) and (c) from Figure 4.2.

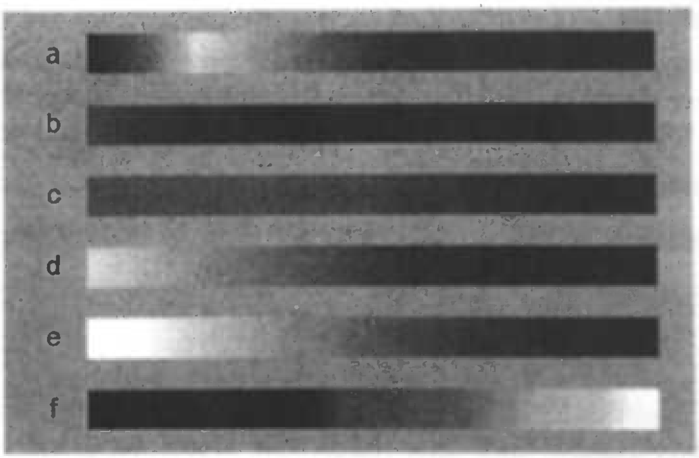


Figure 4.2: Alternative gradients.(a) Spectrum approximation [19]. (b) and (c) sequences commonly used in micro-array experiments [12]. (d) Gradient perceivable by people suffering from the most common form of color blindness [19]. (e) Saturation gradient [19]. (f) A gradient in which each color is lighter than the previous one [19].

Colored labels

The colored label encompasses all visual cues providing information about the statistical corrections. These cues contain the stroke color of the squares, the color of the description and the letters and numbers in that description. These visual cues induce the following uncertainty. It is unknown to which visual cues the attention of the users is mostly drawn. It could be the fill color of the squares, the squares border, or the descriptions to the right of the squares.

The second uncertainty results from the phenomenon that the border of a square may contradict the color of the square's area. Squares having a black fill for example but no border should be interpreted as not being significant. The question remains whether these squares are interpreted as insignificant.

Spatial ordering

The last visual cue is the spatial ordering of the categories of a functional module. In most colormaps, the location of the square gives information about the significance of the category it represents. The most significant categories are situated at the top of the diagram and the least significant categories on the bottom. In the colormaps of two functional modules, the spatial ordering is alphabetic because most users are familiar with their names. It is at this point not clear whether this conflicting information over different functional modules poses any problems.

Chapter 5

Design

See things as you would have them be instead of as they are.

-Robert Collier

The previous chapter took us through some information visualization concepts related to the visualization of FIVA. In this chapter, we will explain how these general concepts are used in the design of the new FIVA version. We will also present the design of the new FIVA in this chapter. In the first section, we will explain how the visualization techniques from the previous chapter are used and present a conceptual design. In the second section, we will present the main components of the new FIVA and describe how they shall interact.

5.1 Conceptual design

5.1.1 Interaction

The first version of FIVA generates diagrams for every functional module and experiment and puts them in one large overview. In all but a few cases, the overview diagram is too large to fit on a regular computer monitor. A maximum of four adjacent diagrams are visible at the same time. To view other diagrams users need to scroll through the overview. Users generally want to compare significant categories from all different diagrams in the overview and not only the adjacent ones.

When we look at the visualization mantra from the previous chapter, we see that the only thing the previous version FIVA does is providing details. These details are however not provided on demand as the mantra states. Users must create the overview in their minds by inspecting the details. Although FIVA does not support filtering by allowing users to hide uninteresting data, filtering is partially supported by FIVA through the use of colors. These colors help users filter uninteresting parts mentally.

In terms of data abstraction, the visualization generated by FIVA shows two levels of abstraction. The first level represents the relations of the functional groups in each individual colormap and the second level the relations of function groups in all colormaps.

According to the multiple view rules of diversity and decomposition mentioned in the previous chapter, we should split these two levels of abstraction into multiple views. One view will show only the overview and one view will show the details of only one colormap. To accomplish this we have split the visualization in two different view types:

- One overview
- A detailed view for every colormap

The overview should be presented to the user first. This overview contains all colormaps generated by FIVA. The diagram is scaled down so that it is entirely visible on the screen and no scrolling is necessary. The colormaps still show the color sequence, but no colored borders are shown. The names and descriptions of the different categories are still present, but the text is unreadable because of its size.

After inspecting the overview diagram the user should be able to select those diagrams, he is interested in and FIVA should present each diagram in a separate view of its own. All selected detailed views should, however, be organized in a single window to minimize the temporal cost of switching to the detailed views and reorganizing them, as stated by the multiple view rule of space/time resource optimization. Adhering to the rule of attention management the window containing the detailed views should automatically be given focus.

A detailed view contains a complete colormap showing all details, as presented in Figure 3.2. Each view will contain only one colormap. Figure 5.1 shows a sketch of the overview and four detailed views. Since we do not expect users to compare the detailed views with the overview, we should present the detailed views in a separate window. This satisfies the multiple view rule of space/time resource optimization.

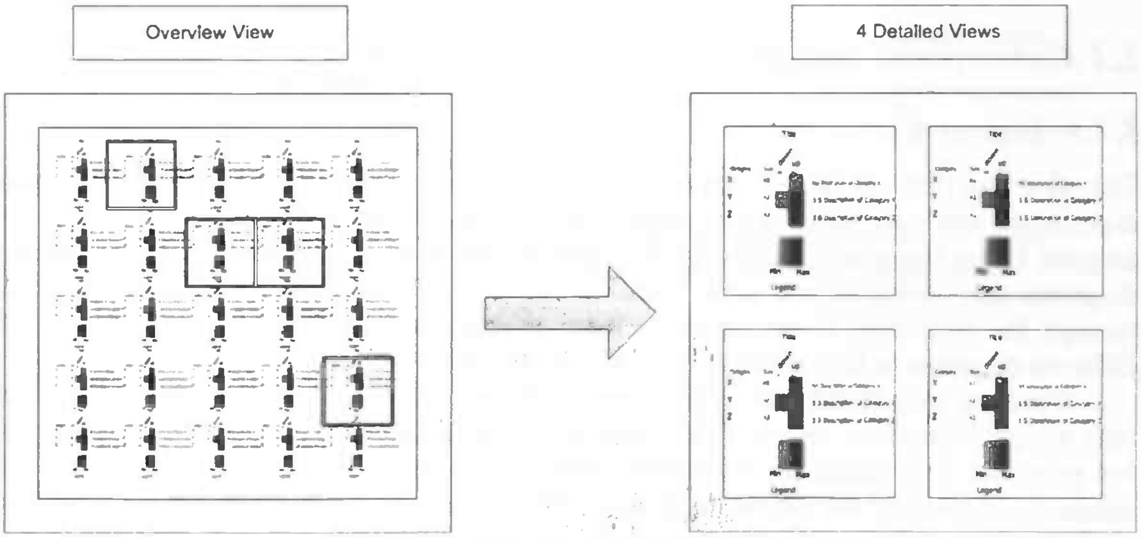


Figure 5.1: Overview and four detailed views. The detailed views show the diagrams selected in the overview

5.1.2 Visualization

In chapter 4, we provided some alternative color gradients. The arguments presented there suggest that an alternative scheme will be better suited for the application. Since we are currently not sure which of the alternatives are better, if any, than the one currently used we will not design an alternative gradient at this stage of development.

5.2 Technical design

5.2.1 Main components

To provide full support for the two view types described above we designed the new FIVA to be composed of two main components: an input component and a miner component.

FIVA Input component

This component houses the overview part. It also provides functionality for loading all data needed for the visualization and provides progress and data feedback. In order to provide this functionality in an orderly fashion we decomposed the input component in the following five parts:

- Import Part
  This part provides functionality for loading genome data and functional modules.
- Search Part
  The functionality for loading experimental data and selecting visualization settings is provided by this part.
- Overview Part
  This part displays the overview picture and provides functionality for selecting individual colormaps. It will also be possible to zoom in on the overview and scroll through it.
- Information Part
  This part presents the number of genes in the database and number of elements in the functional groups that are loaded.
- Feedback Part
  Because biological data is far from perfect, various errors can occur. These errors general consist of missing genes in a dataset, or inconsistent gene identifiers. These errors are shown in this part. It also displays progress information.

FIVA Miner component

This component is the home to the detailed views. It consists of two parts:

- Detailed views part
  This part shows the detailed views that were selected in the overview part and all these views are displayed simultaneously. The part provides zooming and scrolling functionality for each detailed view individually. The text of the colormaps in the detailed views links to files containing more information about the genes in the selected groups. It is also possible to select different squares in the colormap. This selection can be used to gain more information about the distribution of genes over the different functional categories.
- Venn master part
  This part creates a Venn diagram showing the functional groups selected in the detailed views part. Since the functional groups consist of genes, the diagram shows the dependency of different functional groups on each other.

5.2.2 Component interaction

Figure 3.1 shows the stages of the old version of FIVA. The first three stages, concerning loading all data, still satisfy our needs and will be maintained in the new FIVA. The fourth stage, being the analysis, will be divided in two new stages: diagram selection and analysis. Figure 5.2 shows this changed program flow of the new FIVA relative to the flow depicted in Figure 3.1. In the first three stages, the user selects the input data he wants to analyze. In the fourth stage, FIVA presents the user with an overview of all data as explained in the previous section. In this stage, the user selects diagrams he wants to see in mores detail. In the fifth stage, these diagrams are shown and the user can make a detailed biological analysis. If the user wants to see other diagrams, he can switch back to the overview and make a new selection. This new program flow adheres to the information visualization mantra presented in chapter 4, because the new FIVA presents an overview first and detailed diagrams can be shown on demand.

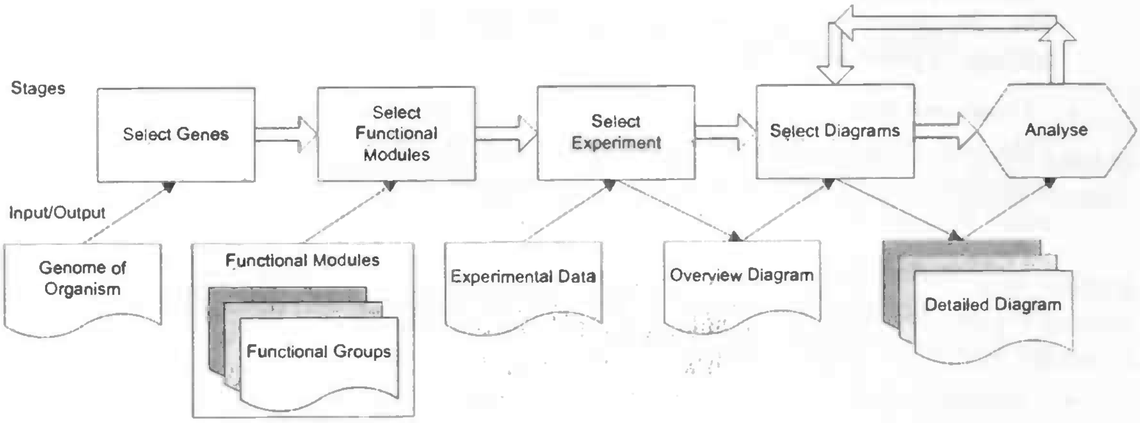


Figure 5.2: Program flow of the new FIVA. The first four stages are part of the FIVA Input component; the last stage is part of the FIVA Miner component.

Chapter 6

Implementation

In the previous chapter, we presented the design of the new FIVA. This chapter explains the implementation of the design. We will limit the description to the information visualization and the GUI parts. A description of other parts of FIVA, responsible for processing biological information files, is beyond the scope of this thesis.

6.1 Development platform

The entire application is built on Eclipse 3.1[11], using Java as programming language. Eclipse provides GUI components which can be reused and adapted to the needs of FIVA. The most notable Eclipse reused component is the *view*. All GUI parts mentioned in the previous chapter are implemented as Eclipse views[12].

All diagrams in the overview and detailed views are generated using the SVG 1.1 (Scalable Vector Graphics) format specified by W3C (World Wide Web Consortium). SVG is a modularized language for describing two-dimensional vector and mixed vector/raster graphics in XML [17]. For a complete specification refer to [17].

We used components from the Batik toolkit to display the SVG images in FIVA. More information on Batik can be found in [2].

6.2 FIVA Input

6.2.1 Main functionality

As mentioned in the previous chapter the FIVA application consists of two main components: FIVA Input and FIVA Miner. FIVA Input is used for loading all data and showing the overview. FIVA Miner is used to display all individual colormaps selected in the overview. Each component is implemented as a different Eclipse program and is contained in its own application window. All parts of each component are implemented as a separate Eclipse view. Thus, FIVA Input consists of the following five views.

- Import Information view
  This view implements the Import Part. The view consists of a panel with different widgets. These widgets are used for loading different data and specify certain settings.

- Search data view
  The Search Part is implements in this view. It is a panel containing widgets for loading experimental data and setting processing and visualization options.

- SVG Preview

[11] For more information on Eclipse refer to [9].

[12] This terminology can be a little confusing since we already used the term view extensively in previous chapters. In this chapter when we use the word view we mean an Eclipse view, which is a well defined reusable Eclipse component.

The SVG Preview implements the Overview Part. It contains a canvas for displaying the overview diagram. This canvas also provides functionality for zooming and scrolling and for selecting different diagrams in the overviews. The view also contains some buttons that provide some additional functionality.

- Information view
  This view shows a read-only list of two columns. The table shows the number of elements in the gene database and the functional groups. The table is implemented as a JFACE table.
- Console view
  This view shows status information and errors when processing input.

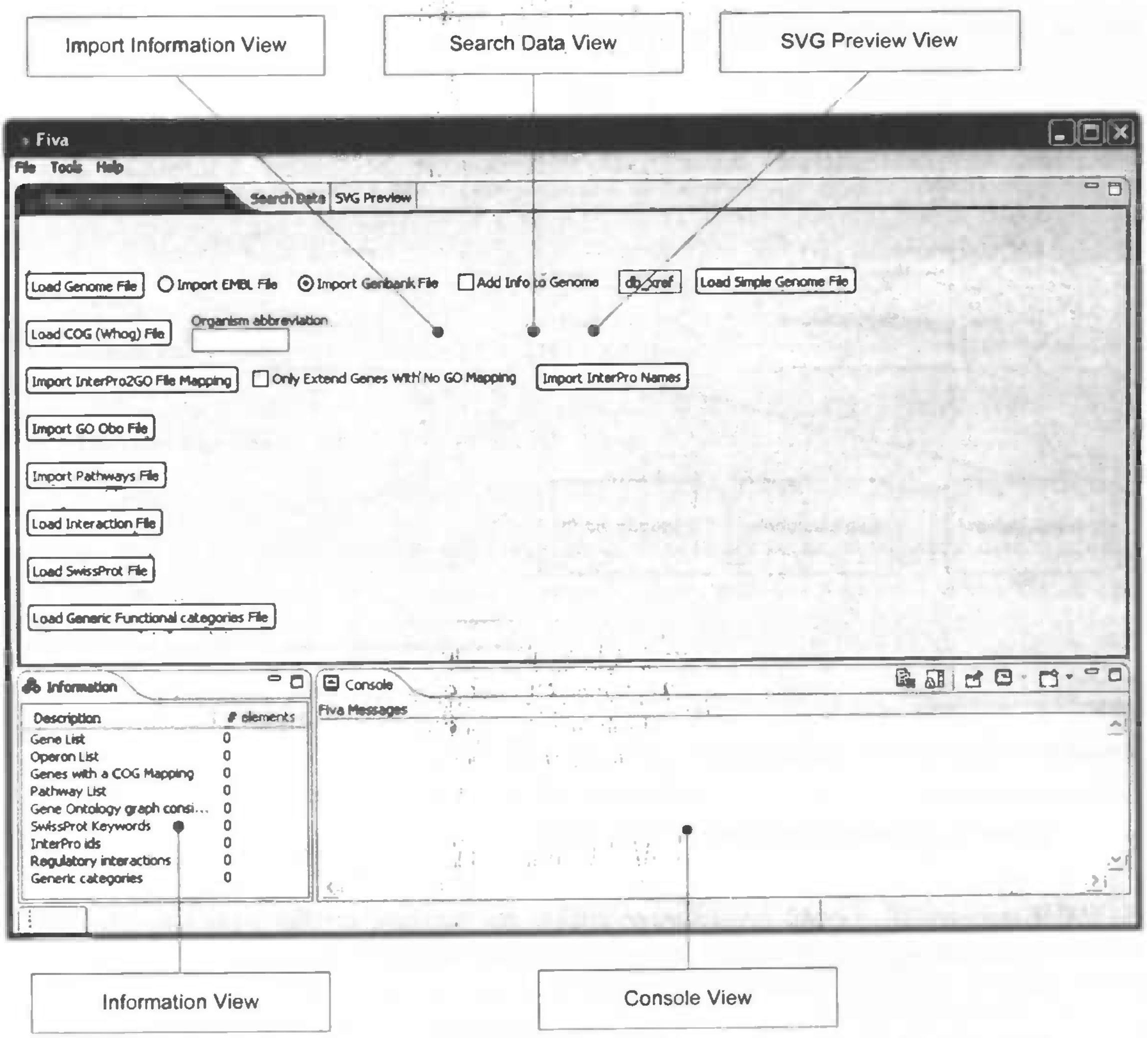
6.2.2 *Additional functionality*

Besides improving the visualization of FIVA, some additional functionality was added to make FIVA more convenient to use. This functionality is implemented in separate menus.

- Save Database
  Users generally want to use the same input data more than once. Because loading the same genome file and relevant functional groups quickly becomes a tedious business, users can save the input once it is loaded in a file. This option is implemented as a file menu entry.
- Load Database
  Saving a database is only useful if it can be loaded at a later time. This option can also be found in the file menu.
- Specify memory settings
  When processing large files, the standard amount of memory that is allocated to the application might not be sufficient. This option allows the user to change the allocated memory. It is a reused Eclipse preference dialog. It can be found in the options menu.
- Import pathways
  The user must be able to import pathways from the KEGG (Kyoto Encyclopedia of Genes and Genomes) database. It can be found in the options menu.
- Create Input Files
  The user must be able to store the input settings in an input File. This option can be found in the options menu.

6.2.3 *Layout*

The Import information, Search data view and SVG Preview are put in the same container and are initially not simultaneously visible. The Information view and Console view are located at the bottom of the application window and remain visible.



○ Import EMBL File ● Import Genbank File □ Add Info to Genome

□ Only Extend Genes With No GO Mapping

Description	# elements
Gene List	0
Operon List	0
Genes with a COG Mapping	0
Pathway List	0
Gene Ontology graph consi...	0
SwissProt Keywords	0
InterPro ids	0
Regulatory interactions	0
Generic categories	0

Figure 6.1: Layout of FIVA Input. The Import Functional Information view is visible in this picture

6.2.4 Package structure[13]

Now we know which parts are responsible for which functionality, let us delve a little deeper in the implementation details. All components are implemented as one or more JAVA classes. In this section, we will describe the organization of these classes in packages.

[13] The FIVA packages all have the character "e" trailing their names. This peculiar character is a legacy phenomenon. The character was initially used to differentiate between the old FIVA and the new FIVA under development. "e" stands for Eclipse or Extended.

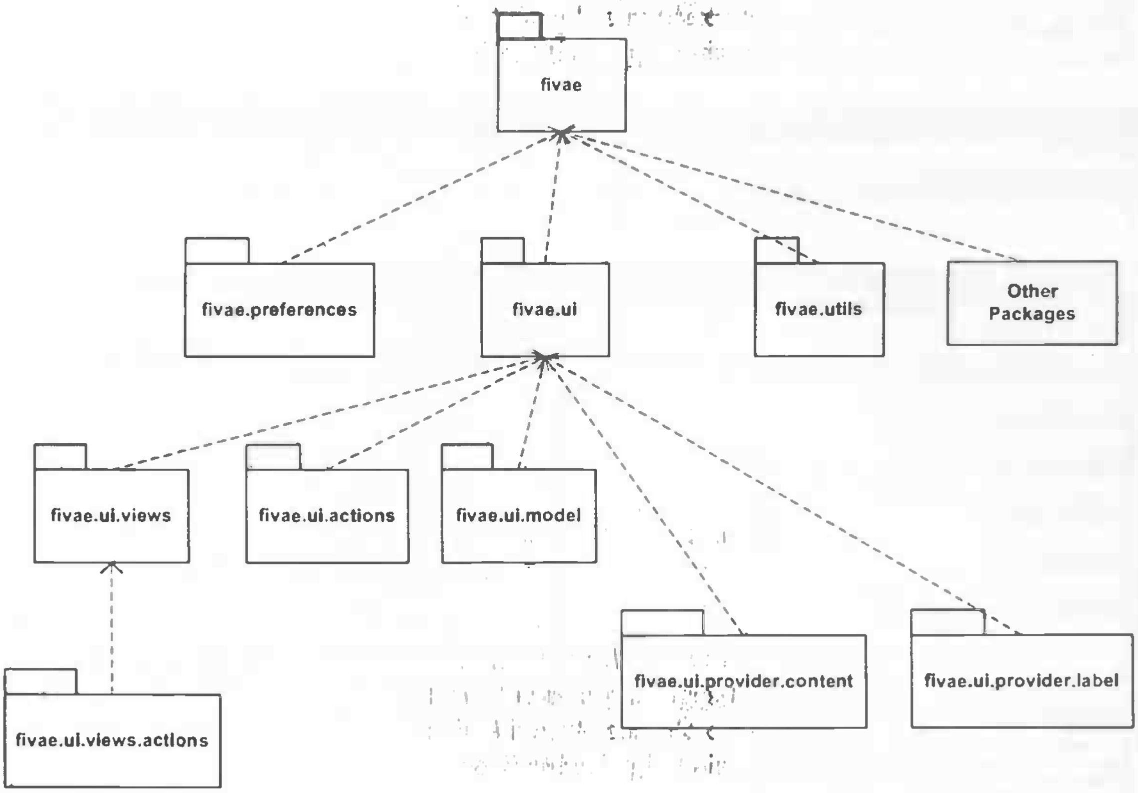


Figure 6.2:Package structure of FIVA Input

- fivae
Contains all global classes responsible for starting up the application, loading saved settings and cleaning up the host system when FIVA exits.

- fivae.preferences
Contains all that is necessary for a preference dialog. The classes in this package implement the specified memory settings functionality.

- fivae.ui
Contains all global UI classes.

- fivae.ui.actions
All classes that manage the actions which are available in all views of Fiva. Each action is managed by one class.

- fivae.ui.model
Contains classes that serve as input for JFace viewers. The classes in this package together with the classes from the two packages below form the implementation of the JFace viewers used in Fiva. A complete explanation of the model view control principle used by JFace viewers is given in chapter 15 from [9].

- fivae.ui.provider.content
Contains classes that implement the JFace content provider structure. See chapter 15 from [9].

- fivae.ui.provider.label

Contains classes that implement the JFace label provider structure. More on this in chapter 15 from [9].

- fivae.ui.utils
  Contains a variety of classes used to generate the visualizations from the biological data.
- Other Packages
  This element contains all packages and classes needed to for FIVA to perform its task, but have nothing to do with the GUI or the visualization. A description of these components is beyond the scope of this thesis.

6.3 FIVA Miner

6.3.1 Main functionality

The Miner component is a much smaller one than the FIVA Input, since its only function is to display individual diagrams and allow user interaction with them. It consists of one type of view.

- Colormap view
  This view implements the Detailed View Part. The view displays a single colormap. It supports zooming and scrolling functionality. Users can also select and deselect individual squares in the colormap.

6.3.2 Additional functionality

Elements from most functional categories can also be found in other functional categories from the same or different functional modules. To get an idea how elements in a category are distributed over other categories or how different functional categories overlap, users can use the following two functionalities:

- Save Selection
  This option saves elements from the selected squares in the different colormap views in a user-defined cluster. Next FIVA Input automatically calculates how the genes in cluster are distributed over all functional categories that were initially loaded. The results are shown in an overview diagram that is added at the right side of the overview diagram. This functionality is implemented as a file menu option.
- Generate Venn Diagrams
  This option starts Venn Master using the selected functional categories as input. Venn Master generates Venn diagrams of the selected functional categories. This functionality is also implemented as an option in the file menu of FIVA Miner.

6.3.3 Layout

The layout of the Colormap Views in FIVA Miner depends on the number of selected colormaps in the SVG Preview of FIVA Input. All selected Colormaps Views are shown simultaneously and are distributed equally over the available space. To adhere to the multiple view rule of self-evidence the headers of each view contain the name of the functional module of the colormap and the time-point of the map. Figure 6.3 shows an example layout consisting of four Colormap Views.

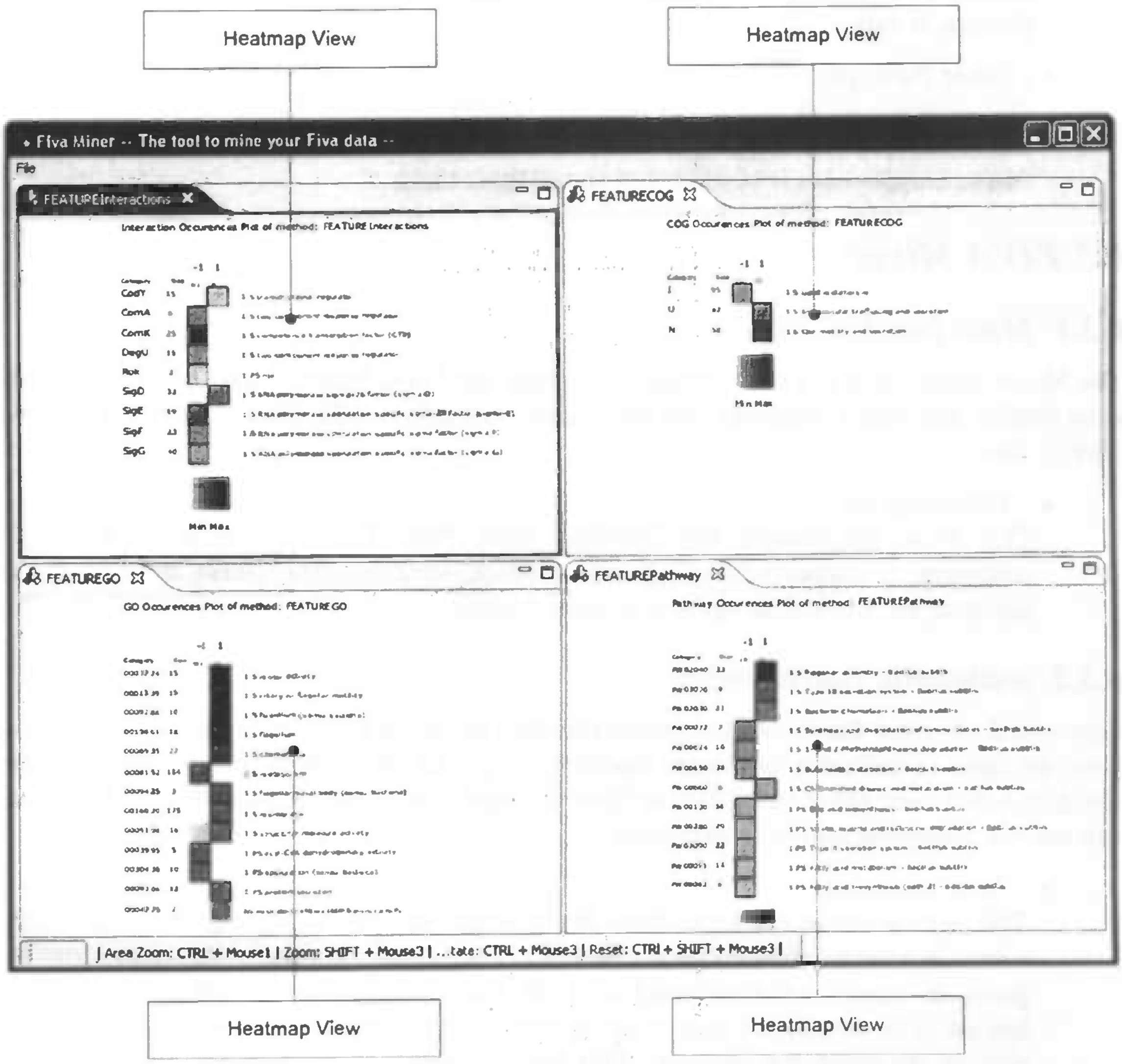

Heatmap View
Heatmap View
Fiva Miner -- The tool to mine your Fiva data --
File
FEATUREInteractions
Interaction Occurences Plot of method: FEATURE Interactions
FEATURECOG
COG Occurences Plot of method: FEATURECOG
FEATUREGO
GO Occurences Plot of method: FEATUREGO
FEATUREPathway
Pathway Occurences Plot of method: FEATUREPathway
Min Max
| Area Zoom: CTRL + Mouse1 | Zoom: SHIFT + Mouse3 | ...tate: CTRL + Mouse3 | Reset: CTRl + SHIFT + Mouse3 |
Heatmap View
Heatmap View

Figure 6.3: Layout of four detailed views in FIVA Miner

6.3.4 Package structure

Figure 6.4 below shows the package structure of the FIVA Miner component. FIVA Miner only has to provide functionality for visualization and interaction. All handling of biological data is done in FIVA Input, thus FIVA Miner is composed of much less packages than FIVA Input.

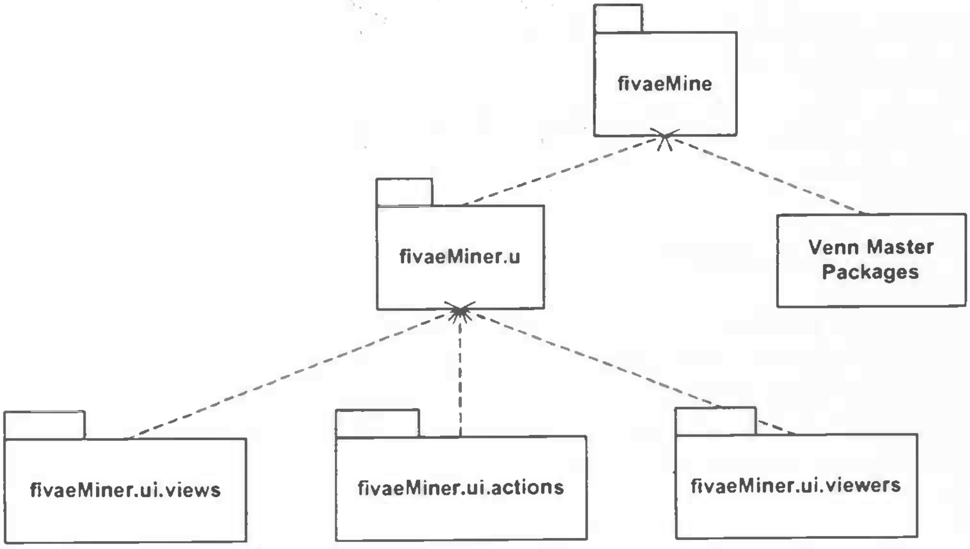


Figure 6.4: Package structure of FIVA Miner

Below is a description of these packages.

- fivaeMiner
  This package contains all classes responsible for initializing the application and cleaning up when it has finished.
- fivaeMiner.ui
  Contains the class responsible for implementing the layout and placing the Colormap Views.
- fivaeMiner.ui.views
  Contains the class that implements all Colormap View functionality.
- fivaeMiner.ui.actions
  The classes in this package are responsible for implementing the Save selection and Generate Venn diagrams functionality.
- fivaeMiner.ui.viewers
  This package contains the class responsible for displaying the actual colormap in the Colormap View.
- Venn Master Packages
  This element contains all packages needed for the Venn Master to work. The Venn Master is a third party application and more information on it can be found in [10].

Chapter 7

Usability Evaluation

The proof of the pudding is in the eating

-Ancient English proverb

We have used theories from information visualization, usability engineering and some intuition to guide us in designing and implementing a new version of FIVA. To test the new FIVA we designed and conducted a usability study. This study is used to check which aspects of the application are usable and which aspects are not. This chapter will describe that study. We will define the goal of the study, provide the used usability methods and present the results of the study.

7.1 Goal

The aim of this study was gathering information on how the users (e.g. specialized biologists in the field of molecular genetics) interact with FIVA. Since there were no previous usability studies performed regarding FIVA, we saw no point in measuring performance and decided to do a qualitative study only. By performing this study, we wanted to verify that the theories from information visualization were adequately used in the design and implementation of FIVA. Moreover, we wanted to check which functionalities of FIVA are used the way they are designed to be and which functionalities are not used. Finally, we want to find any usability flaws of the application that we did not think of at all.

This study should also provide a starting point for further more detailed studies on this tool or more general concepts related to information visualization.

7.2 Experimental design

7.2.1 Usability methods

A wide variety of usability methods used for evaluation purposes exist. We used three techniques: Thinking aloud, observation and conducting an interview [11].

Because we want to obtain qualitative data, the most suitable method for this study is a thinking aloud method. In this method, test subjects continuously verbalize their thoughts when interacting with the application. The thinking aloud method can provide a lot of qualitative data using a fairly small number of subjects. A variant of the Thinking aloud method is the coaching method. In this method, subjects are steered in the right direction by a systems expert. Subjects are also allowed to ask system related questions. This method can be useful because it quickly familiarized novices with a new system, allowing research to focus on the expert parts of the program. Because our main goal is studying the interaction of subjects with the visualization, we adopted this method.

To gain additional data we also observed the actions of the participants while performing the tasks. Observing the users is a method to check whether the new features are used as they should be. Screen recorder software was used to record the user interaction and notes were taken regarding the use of mouse and keyboard. Participants were also allowed to take notes to help completing the tasks. These notes were used in the analysis.

To gain additional information we conducted an interview with each participant. This interview is used to determine whether users were satisfied with FIVA and which aspects they liked, which they disliked and if they would like to see any new features. In addition, we used the interview to clarify thoughts and actions of the participants during the test.

7.2.2 Participants

Ten biologists from the R*u*G (Rijksuniversiteit Groningen) were asked to participate, since FIVA was designed to be mainly used by biologists specialized in the field of molecular biology. All participants are currently working at the group of molecular genetics. All of the participants knew of the existence of FIVA, but only one did use it on a regular basis. The others had very little or no experience. Half of the users were DNA micro-array experts. The other half was a little less experienced with the micro-arrays.

7.2.3 Experiment

The participants were given three tasks and an interview was held. Below we will give a description of these different tasks.

Procedure

Firstly, the participants were first given three instruction videos showing the options of FIVA and how to use them. These videos were intended to make the participants somewhat comfortable using FIVA. We wanted the users to focus on the FIVA visualization during the tasks and not the FIVA interface.

Next, the participants were asked to complete three different tasks. In a task, participants are given a short background and a research question. The participants need to analyze the given data and find an answer to the posed question by using FIVA. This data consists of two files. The first file is a complete gene database including all relevant functional modules and the second file contains the experimental data. The same gene database was used in all three tasks. Participants were asked to load these two files themselves and select the settings (stage 3 from Figure 3.1) we gave them.

To make the participants more familiar with all features of FIVA, we explicitly told them about the interaction possibilities in each view. We also asked the participants to practice using these possibilities in the first task. Also, when the posted research questions were unclear to the participants, or when they had no clue what to do next they received helpful hints.

Participants were also given a piece of paper and a pen to make notes during the task. Participants were only allowed to use a web browser to view information and sites available by using FIVA. They were not allowed to use the browser for other purposes or use any other software application. The participant had to write down the answer to the posed question and when (s)he felt comfortable with the answer, the task was completed.

Following the completion of the third task, we conducted the interview with the participant. The interview was based on a couple of standard questions, but we allowed ourselves to ask improvised follow-up questions. The complete interview was recorded using audio recording software.

Tasks

As mentioned above, the participants were given three tasks. The first task consisted of finding the most striking biological effects in the data of a single array experiment. The second task involved the comparison of the behavior of functional groups to the behavior of a certain regulator over four time points. In the last task participants had to find a common biological effect by inspecting a large time series consisting of thirteen time points. The main difference between the three tasks is the number of time points in the dataset. We choose these different datasets because we expect participants to behave differently depending on the number of time points involved. These three datasets were typical examples of the three different cases. For a complete description of all tasks, see Appendix B.

Output

The output of the experiment consists of the following three elements. A recording of the interaction of all the participants with FIVA, together with the thoughts that were spoken aloud and the notes they took. These recordings include spoken observations made by the experimenter regarding the behavior of the participant. The second output listed the notes made by the participants and the answers the participant gave to the questions. The last output is a recording of the interview that followed the completion of the tasks.

7.2.4 Test facility

We conducted the usability test on a PC with the following specification. The processor is a AMD64 3200+ running on 2 GHz and the PC has 1 GB RAM (Random Access Memory). The PC was connected to a 19" CRT (Cathode Ray Tube) Monitor having a resolution 1280 x 960 showing 32 bit colors and a refresh-rate of 85 Hz.

To record the user interaction and spoken aloud thoughts we used Screen Recorder Pro from RiverPast [15]. The interview was recorded using the audio recorder and editor Audacity [1].

7.2.5 Pretest

Before performing the usability experiment, we tested the whole procedure using two participants. One person has a background in bioinformatics and worked on FIVA, the other had been using FIVA on a regular basis and is also familiar with the experimental data.

The results of the pretest showed that participants used the zoom and scroll functionality in the FIVA Input part more extensively than we had expected. Since these functionalities were not easily accessible, we decided to coach the participants in using all features. In this way we made sure the participants were familiar with all options in FIVA during the test.

7.3 Results

7.3.1 Results of the tasks

From the data, we classified user behavior that occurred frequently in the categories shown in Table 7.1. The "yes" column contains the number of participant that showed the behavior once or more. In the "no" columns are the number of participants that did not show the behavior in all three tasks.

User behavior	Yes	No
Uses zoom and scroll functionalities	9	1
Complains about the difficulty of the zoom	5	5
Uses FIVA Input FIVA Miner	10	0
Uses SHIFT/CTRL for multiple selection	5	5
Asks what abbreviation means	3	7
Asks for easier way to select and deselect diagrams	4	6
Not clear about time-points layout	5	4
Trouble finding the right button	6	4

Table 7.1: Results of the usability test

Uses zoom and scroll functionality

The overview diagram can be zoomed in and out. We did not expect users to make use of this functionality, but left it in place because removing it was more work. During the pretests the zoom and scroll functionalities were extensively used, so we decided to inform the test subjects of this functionality and let them practice with it. Most users started using the zoom functionality during task 2 or task 3 when the experimental data contains more time points.

Complained about the zoom and scroll functionality

Using the zoom and scroll functionality is by no means easy since it requires the user to hold the SHIFT button, an appropriate mouse button and move the mouse. This difficulty made people think twice before using it. Since it was hard to measure how much effort it took the participants to use this functionality we decided to measure which participants actively complained about the zoom functionality. Half of the participants made criticizing comments about this difficulty. From the other half it is not clear whether they did not mind, or did not feel the need to complain.

FIVA Input to FIVA Miner sequence

We designed FIVA to display an overview diagram containing all colormaps first. We expected users to select the interesting diagrams without using the zoom functionality. It turned out that some users preferred using zoom functionality over selecting diagrams and view them in the FIVA miner component. Therefore we decided to count how many of them were still interested in using the miner component. All participants still used this component. It was most frequently used in task 1 and least frequently in task 3.

Uses SHIFT/CTRL for multiple selection

Selecting FIVA diagrams in the overview is done by clicking on them using the mouse. The only way to deselect a selected diagram is to click on the diagram again. Half of the users pressed the SHIFT or CTRL button with their other hand when they wanted to select more diagrams.

Asks for select all and deselect all functionality

In the overview, the only way to select and deselect an individual diagram is by clicking on it. Four participants commented on the labor involved and suggested a select/deselect all or similar functionality during the task.

Not clear about time-points layout

The overview of the FIVA Input part provides no directly visible time point information. The only information directly visible is an abbreviation of the functional module that the diagram shows. Information regarding the time points is available by using the zoom functionality in the overview. The absence of visible time point information made four participants ask whether the columns in the overview diagram contains information on functional modules at the same time point.

Trouble finding the right button

Switching to different components in FIVA requires different actions depending on the component that is currently active. This occasionally causes the users to move the mouse to a wrong position in the component before realizing they are required to perform another action. In the worst case, users are so confused that they asked how a desired action is performed. The effects are listed below:

- Invoking the FIVA Miner component from the FIVA Input component. The invocation requires pushing a button in the preview. Some users moved the mouse to the File menu.
- Invoking the FIVA Input component from the FIVA Miner component for inspecting the distribution of genes of the selected functional categories. This task requires a file menu action and switching to the FIVA Input component manually using an operating system specific command. Some users expected to find a similar button as the one used for invoking the FIVA Miner using FIVA Input. Others were confused by the need to switch manually from Miner to Input using the windows taskbar.
- To inspect the genes that are contained in a functional category an internet browser is used. Some users erroneously used the browsers Back button when they wanted to switch from the browser to one of the FIVA windows requires.

Six participants suffered from one of the above effects.

7.3.2 Other behavior

Apart from categories of user behavior listed above, we observed other behavior that occurred less frequently. Since these observations were made sporadically we only give a qualitative description.

Multiple window confusion

Every time Miner is called from the FIVA Input, it is started in a new application window. Users that did not close the Miner windows experienced problems finding the right application window.

Six diagrams maximum

The FIVA Miner component is able to display as many individual diagrams simultaneously as the user wants. It became clear during the experiments that in case of more than six visible diagrams in the Miner, the diagrams became too small to be readable without zooming.

Cognitive overload

Because the FIVA overview presents much information at the same time we expected users to be overwhelmed by the sheer quantity of information. Especially during task three in which users were presented with 78 diagrams in one overview. Many participants, especially first time users, expressed this overwhelming effect by commenting on it. Still all users were able to devise an exploration strategy within 45 seconds.

Trouble switching to Search Data View

After the database has been loaded in FIVA, users need to switch to the Search Data View. In the Search Data View, users have to press the button "Process File". Some users did not know what to do next after the database had been loaded. Sometimes users had no problems following this procedure during task 1 and task 2, but still got confused during task 3. One user commented that for him the procedure was illogical. This observation is hard to quantify in this experiment because the supervisor assisted first time users to follow the right procedure.

7.3.3 Results Interview

In this section, we will present the results from the interview following the final task.

User satisfaction

The participants were asked if they liked using FIVA and were planning to use it in the future. All users liked using FIVA and all of them confirmed that they are planning to use the application in the future.

All users stated that liked the ability of FIVA to provide them with a quick overview of the all micro-array data from an experiment, thus allowing for quick preliminary conclusions. Some users stated that they liked FIVA because, in addition to providing an overview, it also allowed them to scan all data on specific biological details quickly.

User dissatisfaction

All users commented on the terrible keyboard-mouse combination needed for zooming and scrolling the overview diagram. Some users were hesitant using the potentially useful zoom functionality because of this combination. Other users were confident that they would not have any problems using the button combination after a few days of practice. A couple of participants also stated that they lost the spatial context of the individual diagrams when zoomed in. On top of that, it was not clear to most users to which time point a diagram corresponds.

Additional functionality

We asked the participants what additional functionality FIVA should have and participants gave the following three types of answers.

Some users would like to see a more extensive filter option in the overview. The filter should be able to turn off a wide variety of information elements. Some would like to hide everything except the most significant categories; others would like to hide one or more functional modules.

Other users would like to see an option to visualize the regulation behavior of one functional category over time. Functional categories are more or less significant in different time points which users need to remember. Some users drew plots of the behavior of functional categories on a piece of paper. These users liked to see a plot drawing function in FIVA. In essence, some participants liked to see an implementation of the multiple view rule of complementarity presented in chapter 4.

Some users commented on the lack of tool tips in the FIVA diagrams themselves. By clicking on different visual elements in a FIVA diagram, users are redirected to different sources of information (like a gene list, or a complete database). Users had a hard time remembering to which sources different visual elements point.

Educational value

As a side question, we also asked the participants if they thought FIVA has any educational value. Eight users thought that FIVA could be of help to students starting to learn DNA micro-array analysis. One participant suggested it could be used during lectures, as a global illustration. Two participants stated that students should first familiarize themselves with DNA micro-arrays before starting with FIVA.

7.4 Discussion

From the results of the usability study, we concluded that the FIVA succeeds in being usable. All users stated that with FIVA, one can view a large amount of data and draw some conclusions about the data very quickly. Also, the miner component proved to be invaluable.

That being said, FIVA is by no means perfect and there are still usability concerns in the application. We have categorized these usability issues in the following three categories:

- Interaction issues
- Visualization issues

- GUI issues

Interaction issues

The first category concerns all issues regarding the interaction with the micro-array data that is visualized. Most usability problems resulted from the fact that FIVA does not adhere closely enough to the visualization mantra presented in [16]. We initially assumed that users would view the overview, select a few diagrams that they would like to see in detail and if they needed to see different details move back and to the overview and select new diagrams.

From the usability study is became clear that the overview lacked adequate filtering and zooming functionality as the visualization mantra prescribed. Some inadequacies, like more user-friendly zoom and scroll buttons, are easily implemented.

Other inadequacies, including a useful filter function or navigation functionality that prevents user from getting lost in the zoomed image are more difficult to implement. The details on demand part of the visualization mantra is supported adequately. Still the FIVA Miner windows could greatly be enhanced by providing them with more identifying perceptual cues. This is exactly what is stated by the multiple views implementation guideline number three.

Another usability problem is finding the right miner window at the right moment. All detailed views adhere to the multiple view rule of self-evidence; the window containing the views, however, does not. Not applying the rule of self-evidence to all application windows leads to confusion.

Finally, another observation is that half of the participants tend to press the SHIFT or CTRL button when they want to select multiple FIVA diagrams in the overview. In most application (like the windows explorer) selecting multiple data items can be accomplished by pressing the SHIFT or CTRL button. Clicking on an empty space, without using either button results in the deselection of the selected data items. Apparently, the overview presents diagrams in a way that triggers this kind of interaction behavior.

Visualization issues

The second category, visualization issues, concerns the usability issues regarding how the data is visualized by the FIVA colormaps. The usability remains unclear in two aspects of the visualization.

The first aspect is the colors used by the color scheme of the colormaps. Most users did not find the scheme intuitive. On top of that, most of the users were only attracted to black and the green colors, dismissing the other colors as insignificant. It seems that this colormap suffers from the same effect as the rainbow colormap described in chapter 4.

The other issue is the amount of information presented in the visualization. There are many different visual cues in one colormap. Some cues contradict each other in a single colormap, like the colored fill of a square and the color of its stroke. As stated in [11], this can be confusing.

GUI issues

The last category consists of GUI usability problems. There is the issue of having a different button on different locations for actions that are similar from a user point of few. Most of these buttons issues are easily solved, by relocating different buttons in FIVA. Some button problems are a bit harder since they originate from FIVA using external programs to display some types of data.

The other main GUI usability issue is that most users had a hard time moving on to the search data view when they were finished loading a database. The answer to this issue can be found in Figure 5.2. This figure indicates that the only thing a user should do when he is finished loading functional modules is move on to a next part of FIVA. Still FIVA does not provide any clue whatsoever to point a user in this direction, making it difficult for users to move on. Another point regarding the Import Information part and the Search Data part is that in most cases they will be used only once. The Overview part on the other hand will be used much more extensively. Still all three parts are implemented in FIVA as equal views.

Chapter 8

Conclusion

In this thesis we have presented the redesign of FIVA, a computer application used by biologists for DNA micro-array data analysis. We have introduced the general concepts of performing micro-array experiments and explained which information is visualized by the first version of FIVA. We also presented concepts from the scientific field of information visualization that were used for enhancing FIVA. These concepts include the visualization mantra that reads: *Overview first, zoom and filter, then details on demand.* We also presented eight guidelines concerning effective use of multiple views and discussed different visual information aspects of the FIVA colormaps.

We presented the design and implementation of the new FIVA in which we tried to use as the guidelines as effectively as possible. The main design decision was to invent two distinct components. One component handles the overview of all experimental data; the other component is responsible for the detailed views of selected diagrams.

Finally we showed the design and results of the usability tests of the new FIVA in which ten biologists participated that were already using FIVA, or were potential users. The aim of the study was to find out if FIVA was usable enough for the participants and to study common usability problems.

From the usability study we can derive the conclusion that the use of multiple views in the implementation of the new FIVA proved to be effective. All participants stated that they could easily get an overview and all participants used the FIVA Miner component to view more details. Most usability problems regarding the interaction are a result of not providing full support to the visualization mantra. The new version of FIVA provides too little support for the filter part of the mantra.

After closer examination, the usability problems regarding finding the right application window and finding the right button were traced back to a violation of the rules of self-evidence and consistency respectively.

The effectiveness of the FIVA diagrams remains unclear. Different users used different visual cues, but the attention of most of them was mainly drawn to the black colors

Finally, we conclude that the Import information and Search Data parts of the GUI were confusing to some users and should be redesigned.

8.1 Further work

The usability study made clear that FIVA in its current state is usable and will be accepted by biologists for analyzing DNA micro-array data. On the other hand the study also pointed out that there is much room for further development.

8.1.1 Study

As said earlier, it remains unclear whether the current color gradient is the most effective gradient for use in FIVA. Users commented that they had to get used to the color scheme and proposed other schemes using green, red and yellow colors. The usability study also showed that users were most attracted to the blacks and the greens and that other colors were dismissed as insignificant. This led us to the preliminary conclusion that this gradient is conceived as being divided in discrete regions, just like a rainbow gradient. A specialized test to study the effectiveness of the color gradient may shed some light on this issue. The PRAVDAColor module [7] of IBM's Open Visualization Data Explorer [3] might provide some assistance in the study.

As said earlier the FIVA diagrams contain many visual cues, where each cue contains different amounts of information. Some cues complement each other, but some cues contradict which might lead to a wrong interpretation. A careful analysis of which information is presented exactly by which visual cue and an analysis of the relation between different visual cues might solve some problems.

8.1.2 Design & Implementation

Zooming and Scrolling

Interacting with the FIVA overview can be greatly improved by simplifying the key combination needed for zooming and scrolling. Instead of using the mouse buttons and keyboard keys, only mouse buttons should be used for zooming and scrolling.

Another enhancement could be incorporating scrollbars in the overview. The scrollbars would be very helpful for navigation the zoomed overview diagram.

Apart from the scrollbars, more navigation cues could be helpful. A grid showing the time points on the horizontal axis and the functional modules on the vertical axis, much like a grid used in regular maps will probably be a great help for navigation.

During scrolling, the image is currently rendered after the user releases the mouse button. Although no remarks were made, we suspect that the application will benefit if the image is rendered more quickly.

Filter options

The overview provides much information and users have to filter out unimportant pieces of information themselves. This view can greatly benefit from having options for turning on and of different information elements.

GUI improvements

Most users experienced troubles moving to the search data view after loading a database. Clearly, the functionality of the Import Functional Information view and the Search Data view should be implemented in a different way. A docentric [16] redesign using the overview image, containing all FIVA diagrams as central document might result in a more intuitive user interface.

Another, somewhat simpler, GUI improvement is the relocation of certain buttons. Users got confused because, in their eyes, similar actions required different GUI actions. The buttons used by FIVA Input and FIVA Miner can easily be changed. Since FIVA also uses third party applications, some buttons are somewhat more difficult to change.

Functional profile over time

The behavior of the significance of functional categories over time cannot be visualized. Users have to remember this information or write it down. An option that visualizes this information like a graph displayed in a separate window will certainly of help to some users.

8.2 Development

To make the application as appealing as possible some of the above recommendations that could quickly be implemented have been realized. This implementation consists of the features listed below. Figure 8.1 shows a screenshot of the latest FIVA Input version and Figure 8.2 shows the latest version of a FIVA colormap.

Select/deselect all

The overview is extended with a select all and a deselect all button. The buttons are located directly above the overview diagram.

Relocation of two buttons

The two buttons in the overview used for opening the FIVA miner and opening a previously generated overview diagram respectively have been relocated. They are now situated above the overview diagram next to the select all and deselect all buttons.

User-friendlier scroll and zoom functionality

Zooming and scrolling can now be performed using only the right and left mouse buttons respectively. Also resetting the overview diagram to its original size can be performed by clicking the middle mouse button.

Addition of Scroll bars

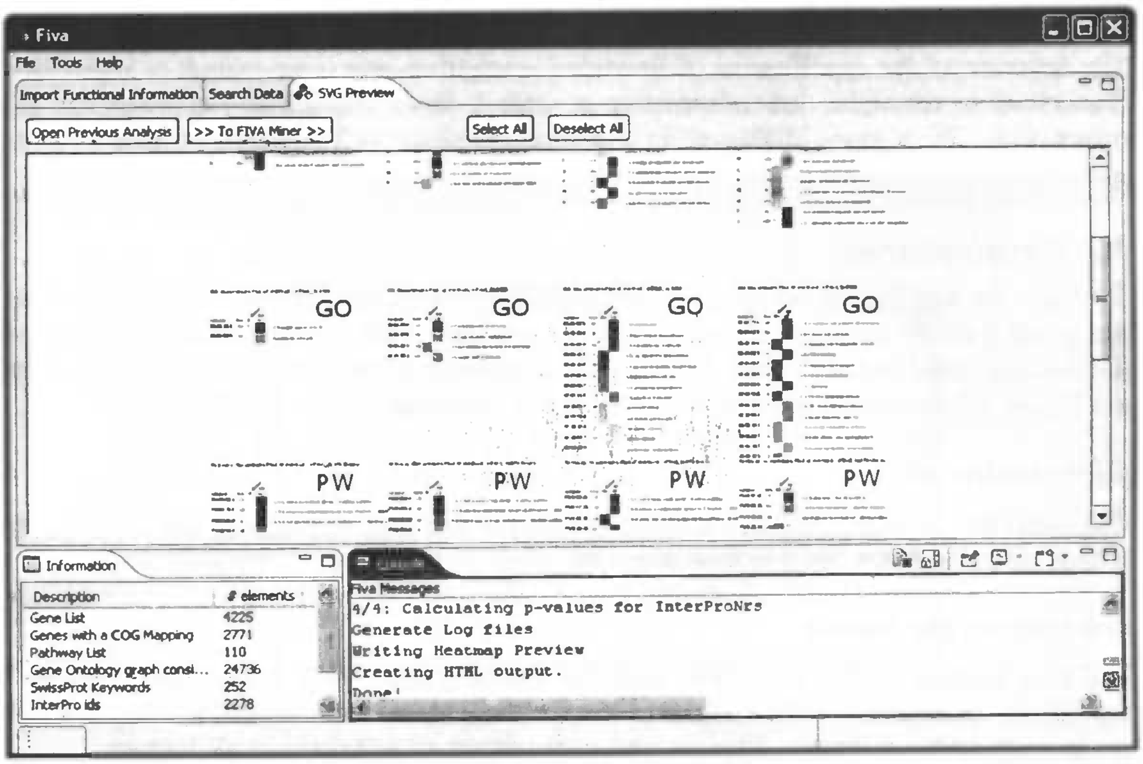
A horizontal and vertical scroll bar has been added to the overview.

Implementation of an additional color gradient

We have added an additional color gradient. The gradient is based on the one depicted in Figure 4.2 (d) which should make the diagrams interpretable for people suffering from the most common form of color blindness. The new gradient can be selected in the Search Data view.

Removal of the colored strokes

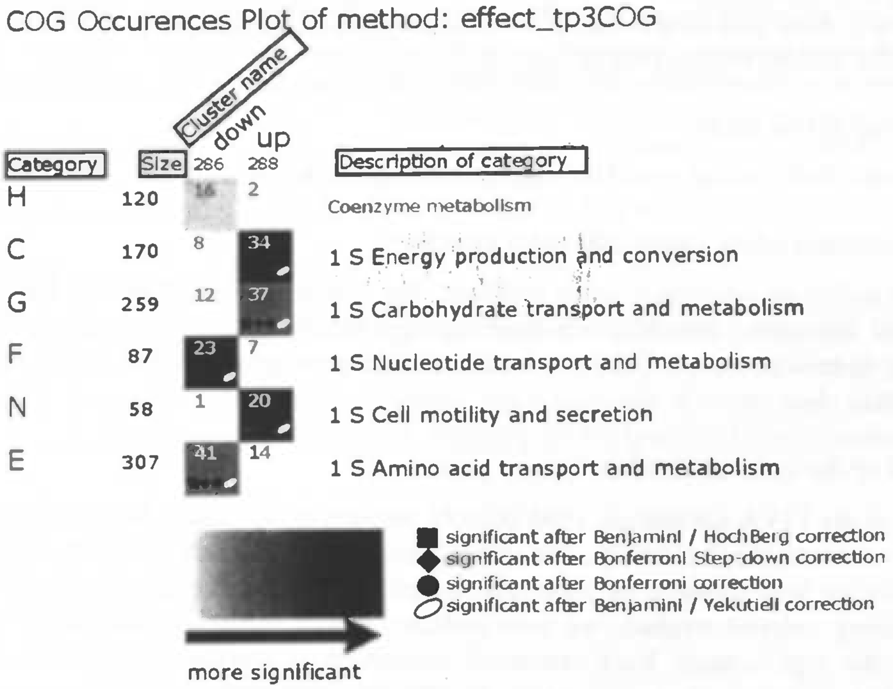
Squares in the FIVA colormaps used colored strokes to indicate whether the significance was real or a statistical anomaly. The classification in significant, possibly significant and not significant was found to be incorrect. Instead of classifying the squares in these three groups using colored strokes, we now indicate which statistical correction specifically supports the significance. Each statistical correction is represented by a distinct shape. When a specific correction supports the significance, the shape of that correction is added to the square.



Description	# elements
Gene List	4225
Genes with a COG Mapping	2771
Pathway List	110
Gene Ontology graph consi...	24736
SwissProt Keywords	252
InterPro ids	2278

```
4/4: Calculating p-values for InterProNrs
Generate Log files
Writing Heatmap Preview
Creating HTML output.
Done!
```

Figure 8.1: Screenshot of the latest FIVA Input component showing the overview diagram



Category	Size	286	288	Description of category
H	120	16	2	Coenzyme metabolism
C	170	8	34	1 S Energy production and conversion
G	259	12	37	1 S Carbohydrate transport and metabolism
F	87	23	7	1 S Nucleotide transport and metabolism
N	58	1	20	1 S Cell motility and secretion
E	307	41	14	1 S Amino acid transport and metabolism

Figure 8.2: FIVA colormap taken from a detailed view as shown in the latest version of FIVA

Appendix A

Eclipse

Eclipse is an IDE for anything and nothing in particular

- [13]

As any introductory document about the Eclipse Platform will tell you, it has many aspects. This section explains which of these aspects are used in FIVA. First a (very) short overview will be given. This section is intended as a background for understanding the design of FIVA. This description is based upon Eclipse version 3.1.

A.1 What is Eclipse

The Eclipse Platform, or just Eclipse, provides in the first place a generic environment for any IDE application. This ranges from different programming languages to graphic design applications[14]. The IDE is highly customizable and extensible by changing its settings or adding home-baked functionality. Apart from being the IDE described above it also provides a platform for building any Java application with GUI. This platform is called the Rich Client Platform (RCP) and is basically composed of the Eclipse UI modules and a runtime engine. All RCP modules can directly be reused, or customized to the applications needs. FIVA was built on this platform using some RCP modules.

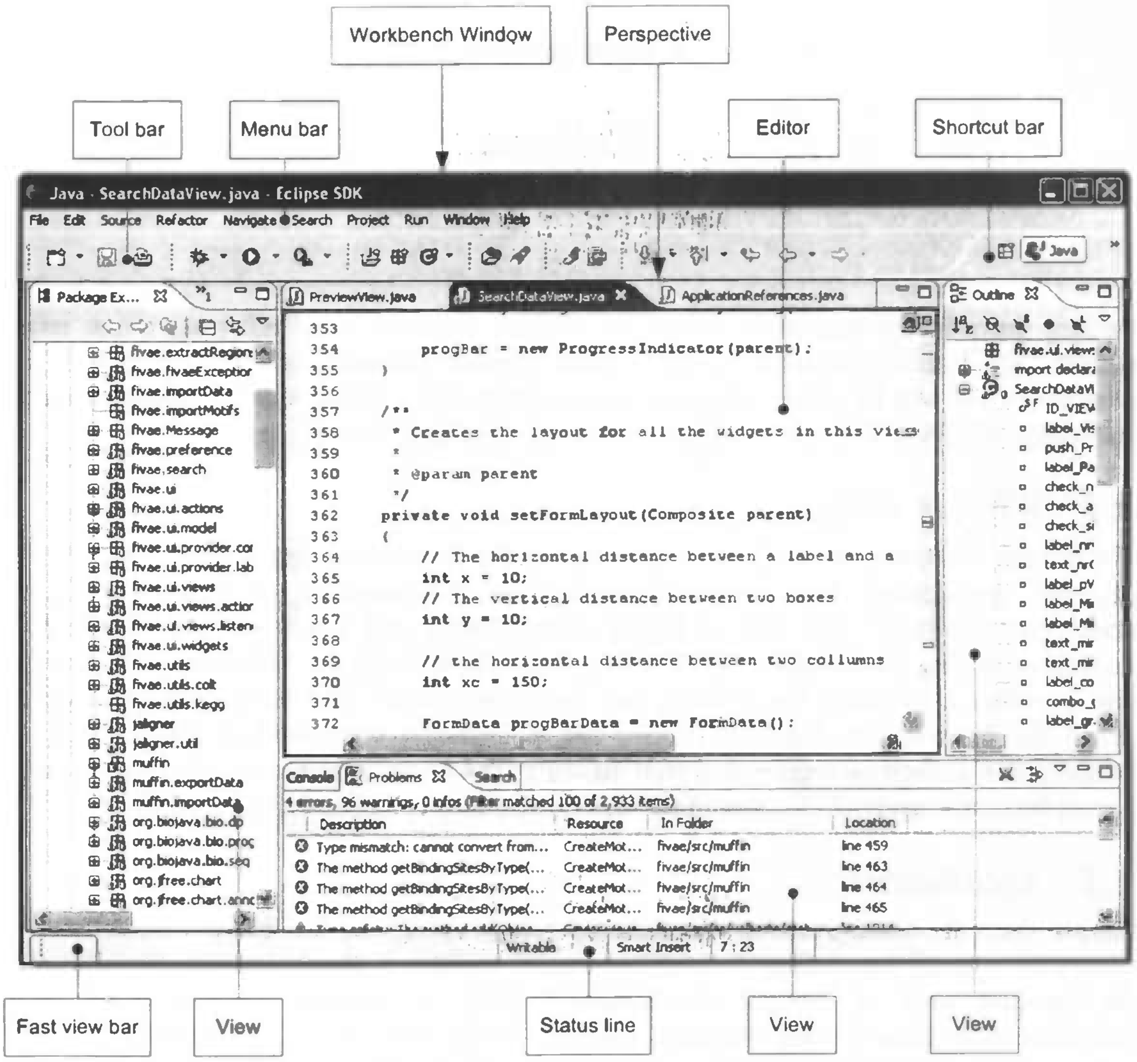
A.2 Architecture

Eclipse gets its customizability through the use of plug-ins. All Eclipse functionality, except for a small part called the runtime engine, resides in these plug-ins. "A plug-in is the smallest unit of Eclipse functionality" [13]. A plug-in contains some base functionality making it work correctly, a set of settings and a set of extension points. The settings are used to control the general behavior of the plug-in in the application. The extension point provides a clean way for adding application specific functionality to the plug-in. The extension point usually provides an API. All plug-ins are interconnected through these points making the design of the application very clean.

A.3 GUI components

Eclipse has a well-established terminology with respect to the components of the GUI. Figure A.1 shows a screenshot of the Eclipse Java IDE with the names of the different components. A description of all components in FIVA is given in the next section

[14] The default appearance of Eclipse is an IDE for the Java programming language for which it was originally designed.



```
    progBar = new ProgressIndicator(parent);
)

/**
 * Creates the layout for all the widgets in this view
 *
 * @param parent
 */
private void setFormLayout(Composite parent)
(
    // The horizontal distance between a label and a
    int x = 10;
    // The vertical distance between two boxes
    int y = 10;

    // the horizontal distance between two collumns
    int xc = 150;

    FormData progBarData = new FormData();
```

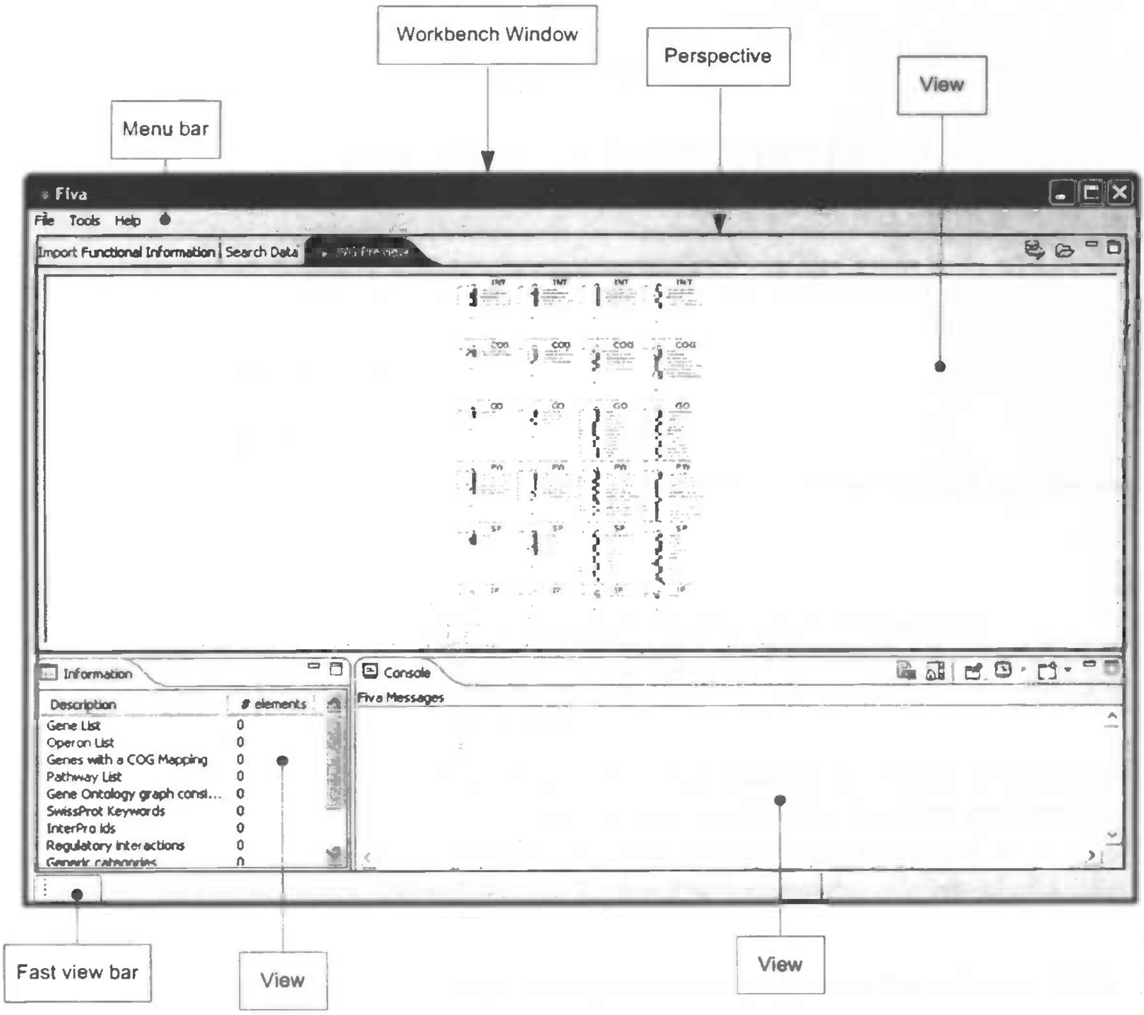
4 errors, 96 warnings, 0 infos (Filter matched 100 of 2,933 items)

Description	Resource	In Folder	Location
Type mismatch: cannot convert from...	CreateMot...	fivae/src/muffin	line 459
The method getBindingSitesByType(...	CreateMot...	fivae/src/muffin	line 463
The method getBindingSitesByType(...	CreateMot...	fivae/src/muffin	line 464
The method getBindingSitesByType(...	CreateMot...	fivae/src/muffin	line 465

Figure A.1: Screenshot of the Java IDE provided by Eclipse

A.4 FIVA as plug-in

All RCP applications are Eclipse plug-ins, because they are designed using the architecture described above. So FIVA can be seen as a plug-in, which includes Eclipse plug-ins. FIVA is not designed to be extended, so it contains no extension points of its own. In Figure A.2 is a screenshot of FIVA showing all Eclipse GUI components that are reused.



Description	# elements
Gene List	0
Operon List	0
Genes with a COG Mapping	0
Pathway List	0
Gene Ontology graph const...	0
SwissProt Keywords	0
InterPro ids	0
Regulatory interactions	0
Generic categories	0

Fiva Messages

Figure A.2: Screenshot of FIVA Input showing the reused Eclipse GUI Components

Appendix B

Evaluation Assignments

In this appendix, a complete description of the tasks mentioned in chapter 7 can be found. First a description of the tasks, including the answers to the questions, is given. This is followed by the questions that formed the basis of the interview.

B.1 Tasks and Answers

Assignment 1: Single Array

The function of gene *yFIVA* is still unknown. A micro array study on Δ*yFIVA* may reveal more on the function of this gene.

- Questions:

To validate the results of FIVA, which further experiments would you suggest.
Answer: Competence & motility assays

Assignment 2: Small time course experiment

Studies have shown that the CcpA protein of *Bacillus subtilis* is a major transcription factor mediating catabolite repression. A time course experiment has been performed in which the expression leveles of wildtype *Bacillus subtilis* was compared to a strain lacking the gene coding for CcpA

-Questions:

Which functional groups exhibit a comparable profile as compared to the behavior of the targets of the regulator CcpA?
Answer: First three pathways, some sugar transport systems from SP

Assignment 3: Large time course experiment

Bacillus subtilis exhibits a complex adaptive response to low levels of peroxides. Global transcriptional profiling was used to monitor the magnitude and kinetics of changes in the mRNA population after exposure to hydrogen peroxide (H2O2).

- Questions:

Which general biological effect is present in all time points?
Answer: Growth effects
Which (in this lab well-studied) phenomenon is noticeable in the late time points?
Answer: Competence

Appendix C

Interview Questions

This appendix shows the questions that were asked to the participant during the interview part of the evaluation study. These questions served as a starting point for the interview. Depending on the answers participants gave, we asked different follow-up questions.

- What do you like most about FIVA?
- What do you like least about FIVA?
- What would you like it to do also
- Do you think you are going to use FIVA
    - if so could you state for what purpose.
- What do you think of the used colors in the colormap
- Do you see any educational value in FIVA
- If so for what purpose would you use it
- For every function: Preview diagram, FIVA Miner and FIVA Miner output as new input
    - Rate the usefulness 1-5
    - Good things
    - Bad things
    - Improvements?

Appendix D

Forms

This appendix presents the form handed to the participants of the evaluation of FIVA prior to the start of the evaluation. All forms are based on the forms in [14].

D.1 Participant Permission Form

Dear Participant,

I hereby invite you to participate in the evaluation of FIVA (Functional Information Viewer and Analyzer) in the department of Molecular Genetics. This department is part of the Biological Center of the University of Groningen. The experiments will take place in room 22 situated on ground floor of the Hartman Wing (building 6516).

FIVA was developed by researchers from the Biology and Computer Science departments of the University of Groningen. FIVA is designed to aid users in quickly identifying relevant biological processes following transcriptome analysis. FIVA is capable of processing information contained in clusters of genes exhibiting similar gene expression patterns thereby constructing functional profiles. These profiles are visualized as colormap diagrams.

The main goal of this evaluation study is to evaluate the usefulness of FIVA to expert users. We will study if certain features are as useful and monitor any problems users experience with the visualizations of FIVA.

On the day of evaluation, you will be asked to complete three different tasks using FIVA. Prior to the tasks we will provide a short instruction regarding the use of FIVA. Some background information concerning the tasks will also be given. After you have completed the tasks you will be asked to fill in a questionnaire. The complete session shall take about 60 minutes.

All your interactions with FIVA together will be recorded using a screen recording software tool. The session will also be audio recorded. Portions of the recordings may be used for retrospective testing and/or presentations. No names or identifying information will be released without your consent. You may withdraw from this project at any time or refuse to participate without any consequences.

If you have any questions, please feel free to send an e-mail at l.tijsma@student.rug.nl

I look forward to hearing from you. Thank you for your cooperation.

Sincerely,

Lars Tijsma,

Scientific Visualization & Computer Graphics Research Group Dept. of Mathematics & Computer Science, University of Groningen

I agree to participate in the evaluation of FIVA project described above. I understand that there will be observations of my actions during the evaluation session. I agree that my interactions with FIVA are recorded with screen recording software and agree to be audio recorded.

Subjects Signature Name Date

___________________ ___________________ ___________

D.2 Usability Test Consent Form

Please read and sign this form.

In this usability test:

- You will be asked to perform certain tasks with FIVA
- We will conduct an interview with you
- Audio and video recordings of the session may be used

Participation in this usability study is voluntary. All information will remain strictly confidential. The description and findings may be used to help FIVA or any other application developed by members of the Molecular Genetics group at the R*u*G (Rijksuniversiteit Groningen). However, at no time will your name or any other identification be used. The audio and video recordings will be used only for purpose of evaluating the FIVA application and will not be distributed or viewed by anyone not associated with this evaluation process. You can withdraw your consent to the evaluation and stop participation at any time.

If you have any questions after today, please contact Lars Tijsma, send an email to l.tijsma@student.rug.nl , or call 050-3638087

I have read and understood the information on this form and had all of my questions answered

Subjects Signature Name Date

___________________ ___________________ ___________

Usability Specialist Name Date

___________________ ___________________ ___________

Glossary

Quotes are nothing but inspiration for the uninspired

- Richard Kemph

Computer science

Actions
Actions the program performs when the user selects an item from a menu – or tool bar.

API
Application Programmer Interface.

FIVA
Functional Information Viewer and Analyzer.

GUI
Graphical User Interface.

IDE
Integrated Development Environment.

RCP
Rich Client Platform.

SWT
Standard Widget Toolkit.

UI
User Interface.

Biology

cDNA
complementary DNA that is synthesized from mRNA. cDNA sequences are only those DNA sequences that code for genes.

COG
cluster of orthologous groups of proteins.

DNA
Deoxyribonucleic acid. The molecule that encodes genetic information.

Functional Category
A collection of genes representing a specific biological function in an organism.

Gene
The fundamental physical and functional unit of heredity.

Gene expression
the process in which a gene produces its gene product.

Gene product

Either RNA or a protein.

Gene regulation

The expression value of a gene in an organism subjected to certain conditions compared to the expression of the gene in a reference organism.

GO

Gene ontologies.

mRNA

messenger RNA. Molecule that is used as a template for protein synthesis.

RNA

Ribonucleic acid. Molecule that plays an important role in protein synthesis and other cell activities.

References

[1] Audacity: http://audacity.sourceforge.net/

[2] Batik: http://xml.apache.org/batik/index

[3] Open Visualization Data Explorer: http://www.opendx.org

[4] Agresti, A., "A Survey of Exact Inference for Contingency Tables", Statistical Science, 1992, 7: 131-153

[5] M. P. Baker and C. Bushell, "After the Storm: Considerations for Information Visualization", Visualization and Interactive Spaces Lab, 1994. http://vis.iu.edu

[6] M. Q. W. Baldonado, A. Woodruff, and A. Kuchinsky, "Guidelines for Using Multiple Views in Information Visualization." *Proc. ACM Advanced Visual Interfaces*, Palermo, 2000, pp. 110-119.

[7] L. D. Bergman, B. E. Rogowitz, and L. A. Treinish, "A Rule-based Tool for Assisting Colormap Selection." *Proc. IEEE Visualization* 1995, pp. 118-125.

[8] E. J. Blom, D. W. J. Bosman, S. A. F. T. van Hijum, L. Tijsma, J. B. T. M. Roerdink, and O. P. Kuipers, "*FIVA*: Functional Information Viewer and Analyzer - extracting biological knowledge from transcriptome data", Groningen: 2006, *In preparation.*

[9] J. D'Anjou, S. Fairbrother, D. Kehn, J. Kellerman, and P. McCarthy, *The Java Developers Guide to Eclipse*, 2 ed. Boston: Addison-Wesley, 2005

[10] Venn Master: http://www.informatik.uni-ulm.de/ni/mitarbeiter/HKestler/vennm/index.html

[11] J. Nielsen, *Usability Engineering*. Boston [etc.]: Academic Press, 1993

[12] M. A. Noback, "DNA Microarrays", Hanzehogeschool, Groningen 2005, *Not published.*

[13] Object Technology International Inc., "Eclipse Platform Technical Overview", 2003.

[14] P. J. Ogao, "Usability Evaluation of SARAgene-MINT Virtual Reality Application," R*u*G, Department of Mathematics and Computer Science, Groningen, 2004.

[15] Screen Recorder Pro: http://www.riverpast.com

[16] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 3 ed Addison-Wesley, 1998

[17] The SVG Working Group, "Scalable Vector Graphics (SVG) 1.1 Specification", 2003.
http://www.w3.org/TR/SVG/

[18] A. Tözeren and S. W. Byers, *New biology for engineers and computer scientists.* New Jersey [etc.]: Pearson Education, 2004

[19] C. Ware, *Information Visualization, Perception for Design*, 2 ed Morgan Kaufmann, 2004