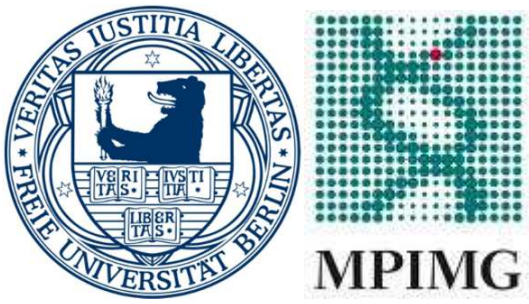
Integration and Visualization of Time Series Expression Data of Gene Regulatory Networks

Svetlana Mareva

October 13, 2010



Free University Berlin

Department of Computer Science

Diploma Thesis

Supervisors: Prof. Dr. Robert Tolksdorf, Prof. Dr. Elfriede Fehr

Advisors: Dr. Christoph Wierling, Dr. Hendrik Hache

Abstract

Time series expression experiments are used to measure the expression of thousands of genes at a time under certain conditions, such as disease or drug treatment. By evaluating the large amounts of data, scientists gather valuable knowledge on various biological questions. An important problem addressed by the study of time series experiments is the discovery of gene function, since it is still unknown for a large set of genes.

A web application- Expression Data Visualiser (EDVis), that enables the integration, visualization and evaluation of time series expression data, was developed and evaluated in the course of the thesis. EDVis provides several methods for comparison of time courses: Euclidean distance, Pearson and Spearman correlation and Dynamic Time Warping algorithm. Thus, one can identify highly correlated curves which in turn determine a possible similar function. Furthermore, the tool can be used to construct user-defined regulatory networks which are essential for the study of celullar processes.

Acknowledgements

I would like to express my gratitude to my supervisor, Prof. Dr.-Ing. Robert Tolksdorf, for his engagement in the process of the thesis construction.

I also thank Prof. Dr. Elfriede Fehr for her effort in the review of the thesis.

I would also like to thank my advisors Dr. Christoph Wierling and Dr. Hendrik Hache, for their invaluable and supportive advises throughout the project.

Not on last place, many thanks to my sister for being a great help for me even in stressful situations. As well as to my parents and my brother for supporting me even from distance. And finally, a big thank you to my boyfriend, for distracting me and cheering me up in difficult moments!

The project is supported by the German Federal Ministry of Education and Research within the MoGLI project of the MedSys program and by the Max Planck Society.

EDVis is being developed in the Systems Biology group of the Vertebrate Genomics Department at the Max-Planck-Institute for Molecular Genetics in Berlin, Germany by myself, Svetlana Mareva under the advisory of Christoph Wierling and Hendrik Hache.

Declaration

I declare that this thesis was written by myself and hereby certify that unless stated, all work contained within this paper is my own.

The thesis is being submitted for the degree of graduate computer scientist at the Free University of Berlin, Department of Computer Science.

Svetlana Mareva

Contents

1 Introduction

1.1 Biological Background

Biological systems consist of groups of components that work together in a complex way to perform a certain task. The harmonious cooperation of these components is the basis for the functionality of every organism on the planet. A disturbance of the system, such as external factors or gene mutations, can lead to diseases or even death of the organism.

1.1.1 Gene Expression Experiments

Nowadays, many experimental methods for observation of biological systems exist. Gene expression experiments are used to answer a large variety of biological questions. Gene expression is a highly complex process in which a gene is switched on at a certain time point and gets activated. Expressed genes code for proteins that are essential for development and maintanance of an organism. Figure 1.1 depicts the gene expression process (very simplified): the information encoded by the DNA is transcribed into mRNA. The information encoded by the RNA is subsequently translated into a defined sequence of amino acids forming a protein. Thus, gene expression can be measured on two levels: either by the amount of the gene-specific mRNA or by the abundance of its respective protein.

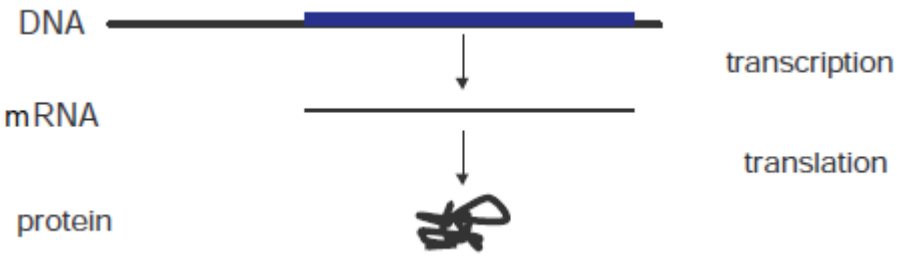


Figure 1.1: This figure shows the basic steps of gene expression: transcription of the information encoded by the DNA into a molecule of mRNA that is subsequently translated into a defined sequence of amino acids forming a protein.

Expression data is divided in two classes: static and time series data. A snapshot of gene expression levels is taken in static expression experiments, for example gene expression levels of tumor cells from different types of cancer. The other type of expression experiments, the time series experiments measure the expression levels of genes in a cell over time, meaning that a temporal process is measured [1].

Data measured by time series expression experiments is further used for the identification of a complete set of genes that plays a role in the biological system and to infer relationships and interactions among these genes [2]. A gene interaction is present when the expression of a gene is controlled by proteins produced by other genes. Time series expression experiments are used in many studies such as [3]:

- Cell cycle or cell-division cycle–A cycle consisting of series of events that lead to cell division and duplication. It is a process by which a single fertilized egg develops to a mature organism, also known as a process by which skin, blood cells, hair and some inner organs are renewed. It plays an important role in the study of cancer, development and many biological processes, thus, it's one of the most extensively studied systems.

- Genetic interactions–Genetic interactions reflect functional relationships between genes and the order in which they operate. Time series expression experiments are the basis for the identification of such interactions, which in turn can be used to build complex gene regulatory networks.

- Infectious and other diseases–Time series expression experiments are used to identify genes that show certain response to infectious and other diseases. Finding such genes is an important step in the development of drugs to fight these diseases.

- Development–Time series expression experiments can identify genes that play key roles in different stages of development. Finding such genes is a crucial factor for understanding many gene diseases.

Microarray experiments are a prominent example for time series expression experiments in biology. In the following I will give an overview on how microarray experiments are organized and carried out, on the purpose of the method and on the type of data produced by them in order to enlighten biological concepts used in the course of my thesis.

The human genome consists of 25,000 to 30,000 genes, according to recent data from sequencing the human genome [5]. Instead of analyzing genes one by one, by the old-fashioned and slow way, scientists are inventing new technologies that allow the observation of thousands of genes at a time. Microarrays (also referred to as DNA chips) are a multiplex technology in bioinformatics and molecular biology, newly invented and at the same time one of the most powerful tools which has emerged from genome studies. A genetic microarray is made with thousands of features (spots) of DNA, containing picomoles of a specific DNA sequence at defined positions on the chip (probes), which will be used to determine the levels of mRNA expression in a collection of cells. Each probe represents a unique region of a gene in the genome. Generally speaking a microarray chip is a grid of DNA spots. The main goal of this method is to determine genes that are expressed when cells are exposed to experimental conditions, such as stress, drought or a toxic chemical.

To understand the essence of DNA chip technology, consider an experiment (example taken from [4]) in which genes that are expressed in cancerous and normal tissue are compared [1]. When a gene is expressed, it is transcribed to mRNA (Figure 1.1). In the example experiment the mRNAs from both tissues are isolated (Figure 1.2 (a)) and converted to complementary strands of DNA (cDNA) (Figure 1.2 (b)).

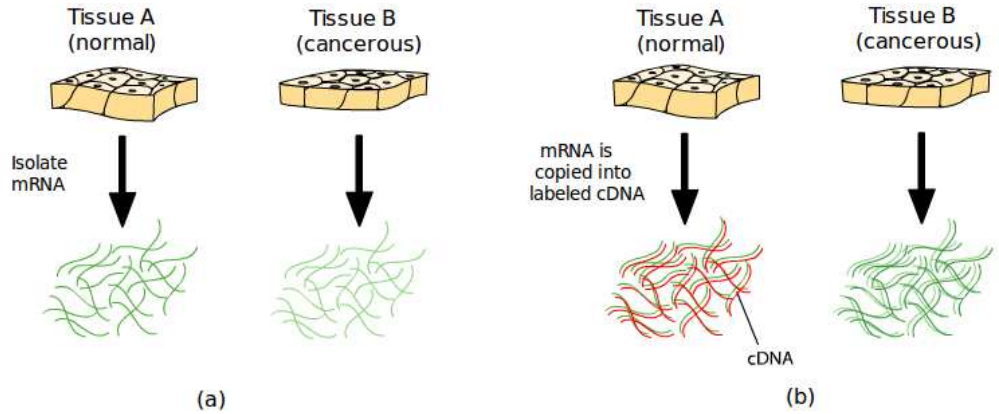


Figure 1.2: (a) Isolation of mRNA, (b) Convertion of mRNA to cDNA, cDNAs from different tissues are labeled with different fluorescent dyes (here red and green). Figure taken from [4].

[1]The course of events for the experiment is strongly simplified for the convenience of a non-biologist reader.

The resulting cDNA samples are mixed together and added to the DNA chip. cDNAs that are complementary to the probes on the chip will bind (hybridize) with the DNA and stick to that location on the chip (Figure 1.3(a)). Unbound cDNA is washed away. cDNA molecules are tagged with fluorescent dyes, so that the expression pattern can be visualized as an image (Figure 1.3(b)). The ready image is further scanned. The provided intensity data for each probe indicating a relative level of hybridization corresponds to expression values that are analyzed by scientists. The measurement of the intensity and the assignment of expression values are rather complicated processes and are therefore not further considered for the description of the experiment. Because each spot on the DNA array contains a known DNA sequence, corresponding to a known gene, it is possible for scientists to detect genes that are expressed differently in the cancerous tissue and to use this information to develop treatment strategies. This type of microarray chips is also known as two-color or two-channel microarrays.

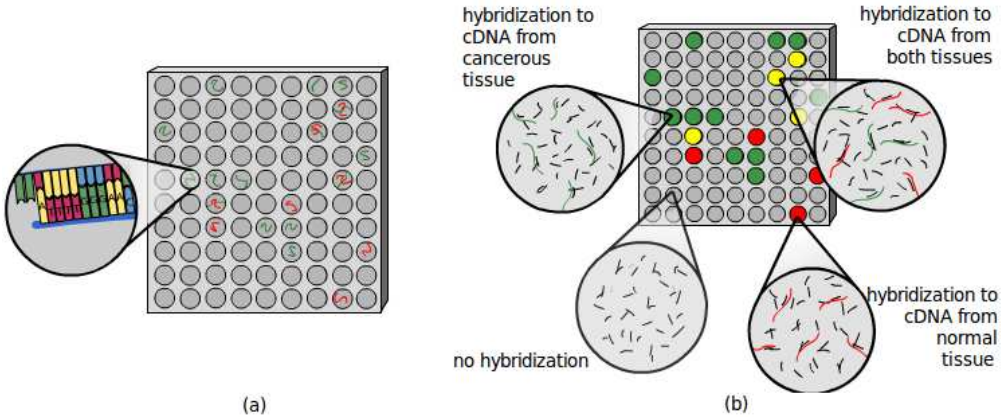


Figure 1.3: (a) Hybridization of cDNA with the probes, (b) Color assignment to tagged cDNA molecules according to fluorescent intensity. Figure taken from [4].

Another type of microarray experiments: single channel microarray experiments are designed to give estimations of the absolute levels of gene expression for each probe. Each array is exposed to only one sample (in constrast to two-color arrays where two samples are tested). Therefore, the derived data cannot be influenced by other factors like a second sample. Another benefit is that data can be easily compared between arrays from different experiments. The absolute values of gene expression can be compared between experiments conducted months or

years apart. The drawback of this type of microarray experiments is that twice as much arrays are needed to compare samples within an experiment.

Next to absolute expression values, microarray experiments produce detection p-values. A detection p-value measures the reliability of concentration measurements. If a p-value lies beyond a predefined value, the probe concentration is not distinguishable from the background noise and is flagged as "unreliable" or "absent". On the other hand, if a p-value is lower than that value, the concentration is "reliable" or "present". The detection p-value helps to detect non-significant components. Another important type of data calculated for microarray experiments and time series expression experiments as a whole is a ratio. A ratio value reflects the ratio of probe concentration measured at a time point of an experiment versus another experiment. Generally speaking, in the context of biology a ratio value tells us whether the concentration of a gene/protein probe in a given experiment is higher or lower in comparison to another experiment.

1.1.2 Gene Regulatory Networks

By now, major effort has been put in molecular biology research in order to study relevant components (proteins, metabolites) of cellular networks in isolation. Thousands of genes have successfully been characterized and functionally annotated by this approach. For biological systems being highly complex and their components being in constant interaction, it is not enough to study only their physiological functions, but also their interactions [6]. Thus, a major goal is to characterize interactions, in particular–gene regulation and its effects on cellular systems, leading to investigation and understanding of multigenic and complex diseases and the development of systems-based medical solutions. A new discipline in biology–called systems biology concentrates on understanding how parts of the organism interact in complex networks. This involves a close study of a large set of genes and proteins aiming to comprehend systems instead of isolated components [7]. The visualization and analysis of time series expression experiments is one of the ways for scientists to identify genetic interactions and build gene regulatory networks.

Gene regulatory networks consist of sets of genes, proteins, small molecules and their mutual regulatory interactions. Development and functioning of an organism's cell result from interactions in gene regulatory networks. Figure 1.4 shows an example of a gene regulatory network inferred from HeLa cell cycle gene ex-

pression data [8]. The visualization of gene regulatory networks plays an important role in undestanding complex gene interactions within such networks.

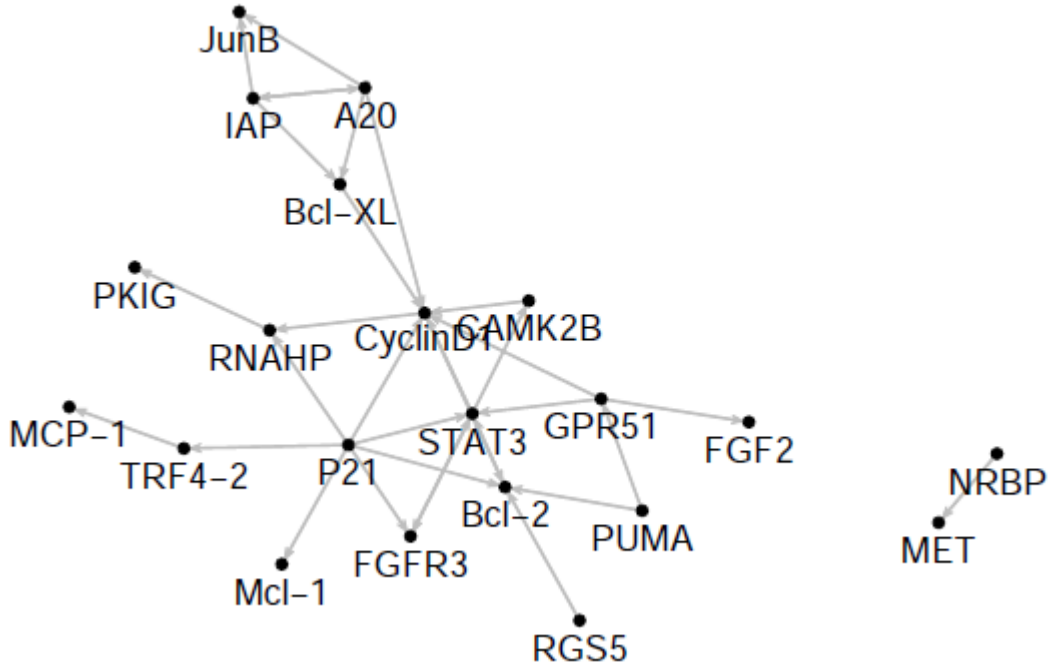


Figure 1.4: An example of a gene regulatory network. Graph nodes are genes, proteins or small molecules, a directed edge connecting one node with another stands for gene regulation.

1.2 Objectives and Outline

As the volume and variety of experimental data grows, the exploration of expression data becomes a challenge. Data visualization plays next to statistical methods a central role in the analysis of experimental data. The field of large scale gene expression analysis is relatively new to scientists since it originates from new technology that uses microarray chips to measure gene expression for genomes. This new method can be used to study the effect of certain treatments

(drugs) or diseases by, e.g., comparing the gene expression of infected versus uninfected tissue. This topic occupies scientists of several disciplines: biologists, computer scientists and bioinformaticians. One of the ways to statistically analyse expression data is through conventional Analysis of Variance (ANOVA) approaches. ANOVA tests are used to find factors in a model that influence the model at most, that are genes with differential expression in control and treatment experiments. It is important for the group of interesting genes to be small enough, so that further investigation is straightforward. However, these tests fail to detect significant genes, that would be recognized as interesting by plotting data [9].

Plots of expression data can be used to focus on differentially expressed genes, to find similar profiles of genes, etc. The most commonly used types of plots are heatmaps, scatter plots and parallel coordinate plots.

Generally speaking, heatmaps are colored matrix plots, with number of rows being equal to number of genes and number of columns - to number of experiments. Each cell in the matrix corresponds to an expression value of a gene for an experiment. The color of each box can vary from red to green, with red reflecting high expression and green reflecting low expression. In order to gain new knowledge from such plots, the cells of the colored matrix have to be reordered, so that genes with similar colors appear in the same region. That way, clusters of genes which behave similarly across a set of experiments can be distinguished.

Most available tools focus on the reorganization and interpretation of such plots. Figure 1.5 shows an example of a heatmap with 77 genes and 4 experiments. The first heatmap is not interpretable, because of the random placement of its cells. The clustered heatmap is a permutated version of the first one. One can easily identify two clusters of genes: one with low expression for the first and second experiment and high expression for the third and forth experiment, and a second cluster with high expression for the first and second experiment and low expression for the third and forth experiment. The major disadvantage of this type of plots is, that it can not detect outliers, genes with significant difference in the concentration for a control and treatment experiment because of the difficulty to map a numerical value to a color. Such outliers are easy to identify in scatter plots.

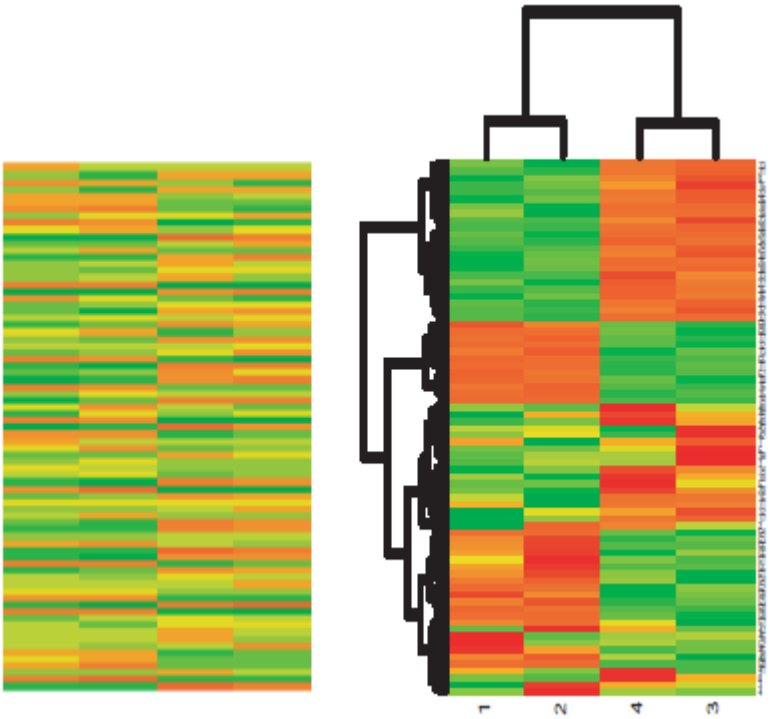


Figure 1.5: A heatmap before and after reorganization [9]. Rows of the first heatmap are of random order, thus, being uninterpretable. The second heatmap is reorganized by a computer program, with clearly detecting two clusters of genes.

Scatter plots are plots that visualize pairwise correlated variables, e.g., control versus treatment experiment. A scatter plot matrix consists of more than two such variables. This type of visualization manages to identify outliers easily. Figure 1.6 shows an example of a scatter plot matrix using the same data as Figure 1.5, but depicting the presence of one outlier. For experiment 1 being control and experiment 2 being treatment, the outlier has lower expression in experiment 1 and higher expression in experiment 2, therefore it responds to the treatment.

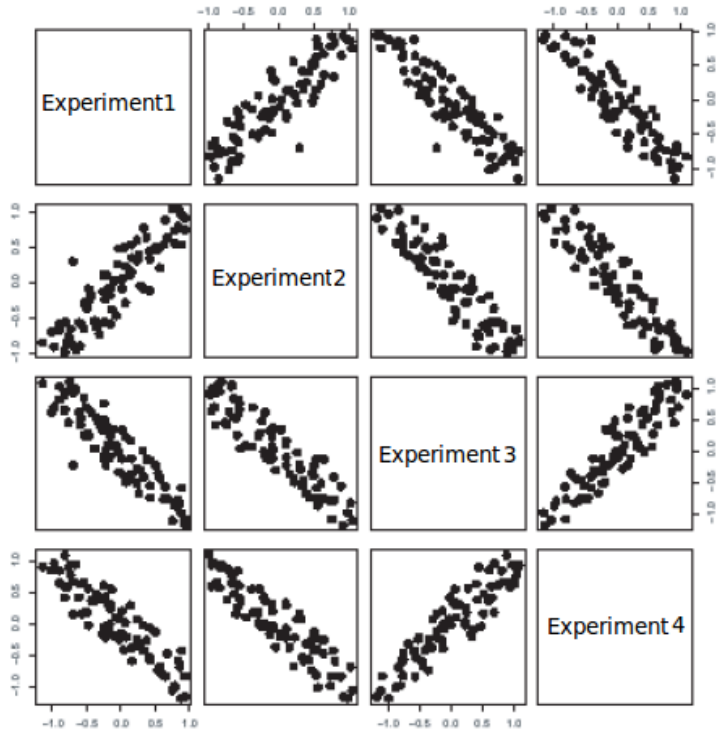


Figure 1.6: A scatter plot matrix with a distinguishable outlier in Experiment 1 versus Experiment 2.Figure adopted from [9].

In contrast to scatter plots whose axes are orthogonal, parallel coordinate plots lay out the axes in parallel. This data analysis tool is used to determine relative and common patterns in the expression profiles of components. Since scatter plots and heatmaps do not hold any information on time, parallel profile plots are most suitable for the visualization of time course expression data. Furthermore, such plots can be used to determine genes with similar profiles (co-expressed genes), which are potentially co-regulated. Finding such groups of genes is often an important step in examining new gene functions and in developing gene regulatory networks [10]. An example plot [11] is shown in Figure 1.7.

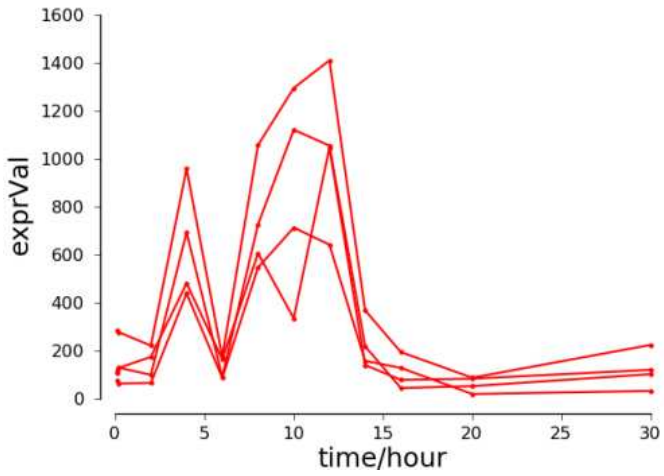


Figure 1.7: Parallel coordinate plot with a cluster of similar gene expression profiles.

Biological networks visualize different types of component relationships, such as, gene interactions, protein/protein interactions and metabolic pathways. They can gather user-defined information as well as literature data in order to help observers to understand the complexity of biological systems and to gain new knowledge in this domain.

However, most common visualization tools concentrate on one of these aspects and do not combine both methods- plot creation and network visualization. A variety of computer applications exists for network visualization as well as for plot creation, but none of these programs is in a position to combine both techniques. Hence, the need for a tool that is able to support following functionalities arises:

- Visualization of time-course expression data
- Visualization of gene regulatory networks
- Search of similar expression profiles
- Integration of user-defined data
- Integration of biological networks from external databases

The new tool has to be user-friendly and expandable for new ideas and functionalities. This diploma thesis focuses on concept design and implementation of Expression Data Visualizer (EDVis), a tool responding to these requirements. EDVis can be reached under http://pybios.molgen.mpg.de/EDVis.

The thesis is structured as follows: Section 1 gives a short introduction to the biological background of the domain and the objectives of the thesis. Section 2 briefly presents some applications closely related to the topic of the thesis. The concept of EDVis including requirements, functionalities and conceptual approaches is described in Section 3. Section 4 represents the implementation of the tool and Section 5 the evaluation of usability and running time, as well as a comparison of the implemented methods for discovery of curve similarity. Some related problem domains and the portability of the tool to those domains are described in the discussion. Finally, the thesis is summarized in Section 6.

2 Related Work

This chapter briefly presents some of the current applications closely related to EDVis and compares them on several criteria in order to show the need to implement a new web application that responds to the requirements.

2.1 Related Applications

Cytoscape

Cytoscape[2] [12] is an open source bioinformatics software platform for visualizing molecular interaction networks and biological pathways and integrating these networks with annotations, gene expression profiles and other state data. It is distributed under the Library GNU Public License (LGPL) and implemented in Java. Many plugins for Cytoscape in form of separate works exist, some of them are for free use and can be easily added to cytoscape.

Cytoscape focuses on gene/protein network visualization and analysis. Network data can be integrated via files (standard formats, such as SIF , GML [13] , XGMML [14] , BioPAX[3], SBML[4] , etc as well as delimited text formats are supported) and viewed by the VizMapper. It represents a bright pallet of features including a variety of layout algorithms for the network visualization, bird's eye view for navigation in large networks and a network manager for the organization of multiple networks. Depending on the level of gene expression, nodes are colored on the scale from one color to another (e.g green to red for negative to positive expression values). Furthermore, web service clients are available. That means that the application can directly connect to external databases and import network interactions. Currently, following databases are supported: Pathway Commons[5], IntAct [27], BioMart[6], NCBI Entrez Gene[7], and PICR[8] . The tool allows the analysis of networks by many criteria:

[2] http://www.cytoscape.org/
[3] http://www.biopax.org/
[4] http://sbml.org/
[5] http://www.pathwaycommons.org/pc/
[6] http://www.biomart.org/
[7] http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene
[8] http://www.ebi.ac.uk/Tools/picr/

- Filtering–select a subset of nodes that share a common property, e.g., genes involved in a number of given interactions .
- Finding Clusters–the network is studied against gene expression data to find related subsets of nodes (clusters, highly interconnected regions).

GenMapp

The Gene Microarray Pathway Profiler (GenMapp)[9] [15] is a stand-alone computer application that views and analyses microarray gene expression data in the context of pathways. Similar to Cytoscape GenMapp displays gene expression data on interaction networks by color-coding the nodes based on criteria defined by the user. Additionally the tool provides links to external databases with further information about genes and interactions. A number of tools concentrate on the analysis of microarray gene expression data by disregarding known gene functions, thus avoiding the bias of the previous knowledge and providing possible gene interactions. GenMapp on the other hand uses this knowledge for further investigation in order to provide new hypotheses about possible new interactions and relations. GenMapp loads pathway information from several databases: the Alliance for Cellular Signaling[10], BioCarta[11], EcoCyc4 [16] and MetaCyc5[12], the Kyoto Encyclopedia of Genes and Genomes (KEGG [17]) and PathDB [18]. Moreover, the user has the opportunity to change the networks for their own use, to create new networks and to apply complex criteria for viewing gene expression data. Users can export their defined networks in a special format defined by GenMapp– the MAPP format and exchange it among themselves. Additionally, existing MAPP's included with the software and created by the help of articles, textbooks and public databases can be downloaded and used.

[9] `http://www.genmapp.org/`
[10] `http://www.afcs.org/`
[11] `http://www.biocarta.com`
[12] `http://metacyc.org/`

PubGene

PubGene[13] [19] is a collective name for the PubGene web tool and an extracted database gathering data from the medical subject heading (MeSH[14]) and the Gene Ontology (GO[15]) database. Data have been automatically extracted from over 10 million MEDLINE (Medical Literature Analysis and Retrieval System Online[16]) records. PubGene is a 'literature network' that organizes data in a way easy to access and navigate. It helps scientists to retrieve comprehensive information about genes and proteins without having to search through numerous databases or papers. PubGene uses text-mining algorithms to retrieve information from the abstract texts of millions of articles and link protein or gene pairs. It generates 'literature networks' in which nodes are genes or proteins and the edges represent the number of articles where pairs of components are mentioned together.

VisAnt

Visual Analysis Tool (VisAnt)[17] [20] is an application for the integration of bio-molecular interactions into graphical networks implemented with Java Applets. The tool can be used as a web-application or as a stand-alone computer program. VisAnt integrates data from a large range of published information on bio-molecular interactions as well as such uploaded by the user. VisAnt uses the MVC (Model-View-Controller) design pattern to separate data from logic and representation, improving data integrity, testing and flexibility.

The main interface of the application, the network visualization panel, supports viewing of large biological interaction data sets (successful test cases with 15,447 nodes and 1,722,708 edges have been carried out). Furthermore, methods for editing, prediction and construction of interactions are implemented. Expression data is visualized in the context of pathways. Networks created by the user can not only be exported in several formats (SVG [21], PNG, JPEG...), but also put online by constructing hyperlinks that open the networks in VisAnt. This feature is convenient for example for paper references or links in other web pages.

[13] `http://www.pubgene.org/`
[14] `http://www.nlm.nih.gov/mesh/`
[15] `http://www.geneontology.org/`
[16] `http://www.nlm.nih.gov/pubs/factsheets/jsel.html`
[17] `http://visant.bu.edu/`

Ingenuity IPA®

Ingenuity IPA® is a web-based application that supports integration, visualization and analysis of gene expression data, microRNA and SNP microarrays and different experiments generating gene lists. IPA® gathers information on genes, drugs, biomarkers, etc. , extracted by experts from a large range of scientific literature and eases immensely the search by merging this data in one tool. Similar to all mentioned tools, it supports network graph visualization, editing and analysis. IPA® allows users to share their research with colleagues through interactive e-mails, lists, analysis summaries and pathways. IPA® is not available for free use, a free 2-week trial can be downloaded.

EGAN

Exploratory Gene Association Networks (EGAN)[18] [22] is a Java desktop application that integrates and visualizes results from exploratory experiments in interaction graphs, providing meta-data for gene lists and direct links to literature and databases (NCBI Entrez Gene, PubMed, KEGG, Gene Ontology, iHOP[19] , Google, etc.). EGAN uses Cytoscape libraries for graph viewing. The tool is similar to the programs mentioned above in its functionality. There are certain functionalities that make EGAN stand apart from the rest though. The program is free of charge in academic research. EGAN performs network module discovery, which means that it can discover genes that weren't present in the experimental set of genes. In other words, it provides possible significant genes that can be added to the experimental network. EGAN is entirely separate from the data model, almost all data can be changed or added by the user, this data is never transmitted to another server. Furthermore, EGAN can be used as a module in a pipeline analysis program.

GeneSpring

GeneSpring[20]is a powerful bioinformatics software, providing methods for statistical analysis and visualization of gene expression data. The tool is created for

[18] http://akt.ucsf.edu/EGAN/

[19] http://www.ihop-net.org/UniPub/iHOP/

[20] http://www.chem.agilent.com/en-US/products/software/lifesciencesinformatics/genespringgx/pages/default.aspx

the needs of biologists in order to favor the investigation and understanding of biological data.

GeneSpring provides statistical tools for testing differential expression (measurements before and after treatment). This feature is of great importance for the investigation of diseases and treatment testing. Differentially expressed genes give important information on how a treatment effects a disease. The biological meaning of differential expression differs immensely from the mathematical definition of the term. In mathematics a difference in the concentration is any non-zero difference before and after treatment. The testing for relevant differential expression has to be adapted in a way that is biologically meaningful [23]. Furthermore, GeneSpring offers methods for pattern discovery in expression data. Clustering algorithms group together similar expression profiles, thus, distinguishing genes with similar biological function. The application displays data in form of various types of plots and diagrams, allowing the simultaneous view and comparison of results from different experiments. Additionally, GeneSpring offers a GeneSpring Workgroup, an environment where scientists can import, exchange, visualize and search analysis results. GeneSpring is not available for free academic use.

Comparison

As the volume of biological data increases, the role of visualization tools becomes of great importance for scientists since it is one of the key methods to gather knowledge from the data. There is a large number of visualization tools available nowadays. In section 2.1 I describe some of the most common tools related to the topic of my thesis. This chapter compares these tools and argues for the need to develop EDVis for none of the applications responds to the requirements (see 3.1 for detailed description). Table 2.1 shows a short overview of the mentioned tools with requirements being satisfied by each application.

Table 2.1: Tool comparison on the basis of supported functionalities.

	Network Visualization	External Database Data Import	External Database Links	Plot Creation	Data Upload	Free of Charge
Cytoscape	✓	✓	✓	✗	✓	✓
GenMapp	✓	✓	✓	✗	✓	✓
PubGene	✓	✗	✓	✗	✗	✓
VisAnt	✓	✓	✓	✗	✓	✓
Ingenuity IPA®	✓	✓	✓	✗	✓	✗
EGAN	✓	✓	✓	✗	✓	✓
Gene Spring	✗	✓	✓	✓	✓	✗

One of the approaches in systems biology for gaining new insights about a molecular network is to integrate preexisting knowledge with large scale experimentally-derived datasets. This idea is used in all mentioned tools except for PubGene. All of them are able to extract knowledge from several databases and combine it with user-uploaded data. Most applications support network visualization and linking to external databases for detailed information about network components or relations. Cytoscape is probably one of the most powerful tools available in this domain. It is being continuously improved, many plug-ins exist and a large number of them can be freely used. It implements many layout algorithms for the visualization of networks and supports graphs with a large amount of nodes. Whereas Cytoscape is a general network visualization tool, GenMapp and VisAnt can view relationships between genes and proteins in networks. Although PubGene is a network visualization tool, it does not respond to the requirements. PubGene supports only 'literature networks', such networks reflect only literature knowledge and no expression data. Nodes in the network are connected if they are mentioned together in the literature. The rest of the tools is similar to GenMapp and VisAnt in their functionalities with some minor differences.

The major problem that occurs by all tools but GeneSpring is that none of them supports plot creation. Undoubtedly, the network visualization is of great importance, but it's not sufficient if a detailed exploration of component profiles is

demanded. Profile plots allow the comparison of many gene expression profiles and can be used to determine genes with similar profiles which are in most cases co-regulated. Finding such genes helps to deduce gene functions and create gene networks.

2.2 Related Databases

CPDB

The ConsensusPathDB (CPDB) [24] is a database that integrates human functional interactions. CPDB is being developed by the Bioinformatics group of the Vertebrate Genomics Department at the Max-Planck-Institute for Molecular Genetics in Berlin, Germany. As the amount of current knowledge about interactions grows, it becomes even more difficult for scientists to extract this information since data is dispersed in more than 200 databases each with a specific data format and focus. The need of a database that integrates comprehensive human interaction data arises considering that collecting such data is the key to gain new insights in cell biology. CPDB stores different types of functional interactions that interconnect different types of cellular entities. The focus of the database is held on the integration of existing database resources, a manual upload of interaction data is also supported. Currently, the database contains human functional interactions, such as gene regulations, physical interactions and biochemical interactions, integrated from 18 publicly available database sources including: Reactome [25], KEGG (metabolic reactions only), HumanCyc [26], PID[21], BioCarta, NetPath[22], IntAct (data from small-scale experiments only), DIP [28], MINT [29], HPRD [30], BioGRID [31], SPIKE [32] and others. As the integrated data overlaps at a certain extend, CPDB offers a method to merge identical physical entities and identify similar interactions.

Furthermore, CPDB offers a web interface [23]. The user is able to search for interactions of specific physical entities or pathways by name or database identifiers through the search function of the CPDB. Found interactions are displayed in a form of a network graph in the visualization environment of CPDB. Network graphs are composed of nodes and edges. Nodes can either be physical entity

[21] `http://pid.nci.nih.gov`
[22] `http://www.netpath.org`
[23] `http://cpdb.molgen.mpg.de`

nodes or interaction event nodes. Different node colors are used to reflect the particular node role. Edges are used to connect interactions with physical entities. Apart from the search of interactions, the user is able to search for shortest path of interactions between each two distinct physical entities. Furthermore, CPDB supports import, export and expansion of networks. Networks can be imported or expanded via files in one of the supported common formats: BioPAX, PSI-MI or SBML.

CPDB is closely related to the thesis application since it holds valuable interaction data integrated from many publicly available databases. A connection to the CPDB would enable scientists to combine user-defined networks with already known functional interactions. Thus, expanding defined regulatory networks for further investigation.

3 Concept

As the load and variety of biological data produced by diverse experiments grows, the need of analytical tools supporting experiment specific functionalities arises. Many existing tools are adapted for a certain type of experiment data and therefore implement analytical and visualization methods adjusted for particular experiment types. Thus, the number of visualization tools has grown, as has the diversity of analytical methods. Section 2 has already shown commonly used visualization tools in the context of expression data. However, none of the applications could offer an appropriate interface for time series expression data visualization and analysis in context of the requirements to the tool.

3.1 Requirements

The central goal of my diploma thesis is to develop a web application, that supports time series expression data integration, graph and plot visualization, as well as methods to identify possible closely related components. Its purpose is to serve as a visualizing tool that can be used to discover new unknown relations or to build assumptions which can be further investigated in laboratories.

Given a large set of time series expression data in a predefined table format, the needed application has to be able to integrate, visualize and analyse the data in a user-friendly way. Furthermore, quick response times for most frequent user requests (e.g. search, creation of plots and graphs) are expected, as well as support of different data resources (lab data and user defined data).

At the same time the web application has to be simple and plain, sophisticated graphical interface is not required or wished. The main focus is kept on the functionalities. A short documentation in a form of help menu is to be delivered. As a part of user friendliness the response times of the application have to be kept as quick as possible, most frequent requests have to be processed in acceptable time[24]. Furthermore the web program has to support different types of data. Users should be able to compare data coming from laboratory experiments with user-defined, e.g., simulated, experimental data in order to be able to verify biological models of diseases.

[24] Chapter 5.3 gives a more detailed description of tolerable response times for web applications and evaluates them in the context of the application.

3.2 Conceptual approaches

As already mentioned in Section 1.1 a main challenge of many biological studies is to discover co-expressed genes. Finding such groups of genes is often an important step in examining new gene functions and in understanding interactions between them. Many of the component interactions are already known to scientists, however there is a large set that has not been discovered yet. One of the main goals of the application is to enable users to find possible unknown dependencies between components. One method is to compare the time courses of all components for an experiment and choose related curves. Genes with similar expression profiles are expected to be functionally related or co-regulated [33]. Grouping genes into clusters on the basis of similarity between their expression profiles has been the main approach to predict functional modules, from which important inference or further investigation decision could be made [34].

One of the key issues in finding similar time series profiles is to define the similarity between two time series. Distance measurements and correlations are commonly used as similarity definitions. One of the most used strategies is to measure the Euclidean distance or make use of a correlation, e.g., Pearson correlation, Spearman correlation. Nevertheless, these methods depend on the measured data. Therefore each of these approaches can be more appropriate for one type of data, but not suitable for another. All three approaches are to be implemented and their results to be compared and observed in the course of my thesis. In addition to these three methods, a forth algorithm - the Dynamic Time Warping algorithm is proposed and implemented [38].

In the following a description of each method is introduced with corresponding application fields for each of them.

3.2.1 Euclidean Distance

The Euclidean Distance determines the actual distance between each two points that can be measured with a ruler. It is the most common use of a distance metric. Applied to vectors, it measures the summed distances between each two points, given that the length of the vectors is equal. Given two vector variables $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$, then the Euclidean Distance d(X,Y) is defined by:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

The use of the Euclidean distance is accurate if one is interested in finding curves with minimal distance between each value pair. The lower the distance the more are two curves related to each other.

However, this classical distance metric fails to capture temporal variations since it is very sensitive to small distortions in the time axes and consequently produces in some cases poor similarity measures between time series. Further disadvantage of this metric is that it cannot detect profiles with exactly the same shape but a relatively large difference in the amplitude (Figure 3.1(b)). Components with different profiles, but with lower expression levels can end up close together (Figure 3.1 (a)).

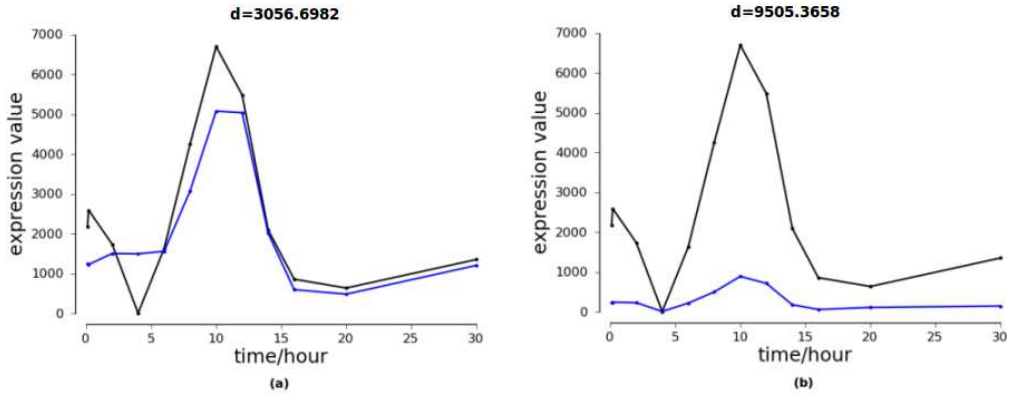


Figure 3.1: According to the Euclidean distance metric the profiles in (a) are closer correlated than the profiles in (b), although the time series in (b) have almost the same shape.

On the other hand, the Pearson and the Spearman correlation are more robust distance metrics in this respect since they measure the course similarity of two curves.

3.2.2 Pearson Correlation

Pearson correlation indicates the degree of linear relationship between two variables. It was developed by Karl Pearson and is therefore named Pearson correlation coefficient. Next to the degree of correlation, this method gives information about the direction of the correlation. The value of the coefficient ranges between +1 and -1. A correlation between 0.75 and 1 means that the variables are in positive linear relationship (as the value of one variable increases, the value of the other variable decreases) and a correlation between -0.75 and -1 stands for a negative linear relationship (as one variable increases, the other decreases). If the value of Pearson's correlation coefficient lies between ±0.25 and ±0.75, then it is said to be a moderate degree of correlation. A Pearson correlation between ±0.25 to zero means that a low/no tendency exists. Scatterplots are useful for checking whether a relationship is linear.

Given two vector variables $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$, where n is the length of each vector, then the Pearson correlation $r(X,Y)$ is defined as the covariance of the two variables divided by the product of their standard deviations [35]:

$$r(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}.$$

An alternative formula for the Pearson correlation is also available:

$$r(X,Y) = \frac{\sum_{i=1}^{n} x_i y_i - \frac{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n}}{\sqrt{\left(\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}\right)\left(\sum_{i=1}^{n} y_i^2 - \frac{\left(\sum_{i=1}^{n} y_i\right)^2}{n}\right)}} \quad x_i \in X, y_i \in Y.$$

Figure 3.2 shows the Pearson correlation applied to two highly correlated ($r = 0.987$) time series variables X and Y, plotted as a scatter plot (a) and as a parallel profile plot (b). The scatter plot contains sets of (x,y) pairs, with $x \in X$ and $y \in Y$. It clearly represents the positive linear relationship between the two variables. The

parallel plot graphic displays both variables as time curves, showing the similarity of their shapes.

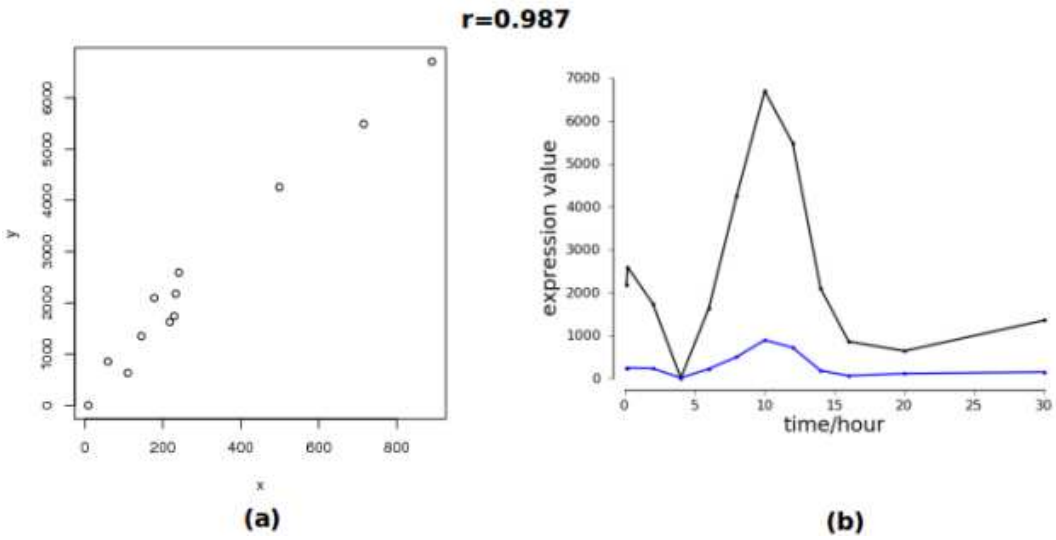


Figure 3.2: Pearson correlation shown for two time series variables: with scatter plot (a) and with a parallel profile plot (b).

An important property of the Pearson correlation is that it is invariant to changes of location and scale. In other words, if X is transformed to $a + bX$ and Y to $c + dY$, with a, b, c and d being constants, then the correlation coefficient remains unchanged. That is, the same correlation coefficient is detected although the curve has been shifted or scaled. The Pearson coefficient is also symmetric, i.e. $r(X,Y)=r(Y,X)$. This means that if we calculate the Pearson correlation between X and Y, or between Y and X, the value of the correlation coefficient will remain the same. Another property of the correlation is its independence of the unit of measurement. For example, if one variable's unit of measurement is meter and the second variable is inches, even then Pearson correlation coefficient would not change.

The Pearson correlation is a common method for measuring curve similarity in time series data. Nevertheless, this method is not suitable for distributions different than normal ones and for variables that have outliers. In these cases the

coefficient is not stable and might show a correlation although none is present. Therefore a more robust correlation coefficient such as the Spearman correlation has to be used.

3.2.3 Spearman Correlation

Spearman correlation (also known as Spearman rank correlation), named after its developer Charles Spearman, is independent of the variable distribution, because it is calculated via Pearson correlation over variable ranks instead of variable values. It is mostly used when the Pearson correlation gives misleading results. The Spearman correlation coefficient ranges also between -1 and 1. If Y tends to increase when X increases, then the correlation is positive. If Y tends to decrease when X increases then the correlation is negative. Zero reflects no correlation. If $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$ are two vector variables with x_i and y_i converted to ranks, n is the length of each vector and d_i is the difference in statistical rank of corresponding variables x_i and y_i, then the Spearman correlation ρ if no tied ranks exist, is given by [36]:

$$\rho = 1 - 6\frac{\sum_{i=1}^{n} d_i^2}{n(n^2-1)}, d_i = x_i - y_i, x_i \in X, y_i \in Y$$

If tied ranks exist, then the Pearson correlation between ranks should be used for the calculation.The rank of a value is equal to its position in the ascenting order of the values. If equal values exist, the rank is calculated as an average of their positions in the order.

Information specific to the distribution is lost when ranks are used. Consequently, the Spearman correlation does not require any assumptions or underlying conditions related to the distribution for the procedure to be valid. In contrast to Pearson correlation that can detect only perfect linear relationship, a perfect Spearman correlation results when X and Y are related by any monotonic function (Figure 3.3). To perform Pearson correlation it has to be given that both variables are normally distributed. Since no such assuption can be guaranteed for experimental data, the Spearman correlation should be preferred in this context. Spearman correlation coefficient should be used more often, it gives as much information as the Pearson correlation coefficient and is of wider validity, as discussed by Altman [37].

In more general cases though, it is often arguable which correlation coefficient is more reliable. As using ranks is a major advantage of the Spearman correlation, it is a disadvantage at the same time, because it neglets information. Ranks only preserve information about the order of the variable values, but discard the actual values. Because of this information loss, nonparametric procedures like the Spearman correlation can never be as powerful as the parametric methods when parametric tests can be used. That is, when it can be assumed that the observed variables are normally distributed. But on the other hand, the Spearman correlation gives more insurance when this assumption is not correct. There are also certain difficulties related to using Spearman correlation coefficient with very large samples. The problem arising is that it becomes very time consuming, because of the need to rank the data for both variables.

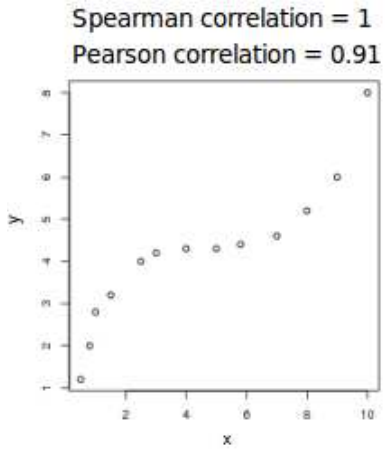


Figure 3.3: Comparison of Spearman and Pearson correlation. Spearman correlation determines that X and Y are perfectly monotonically related $(\rho = 1)$ in contrast to Pearson correlation that does not give the same coefficient $(r = 0.91)$.

Although Pearson and Spearman correlations are widely used for time series data, they are not an optimal solution if a certain time delay of time courses is present. In fact, such time delay can significantly degrade the performance of the correlation methods. Therefore a more flexible method adapted for time delays is introduced– the Dynamic Time Warping (DTW) algorithm.

3.2.4 Dynamic Time Warping

Dynamic Time Warping is an algorithm that measures similarity between two sequences (e.g. time series) that may vary in time or speed. DTW was originally developed for speech recognition [38], but any data that can be turned into a linear representation can be analysed by the algorithm.

For example, DTW can detect similarities in walking patterns even if one person was walking more quickly and the other one slower. A typical wellknown application is the automatic speech recognition, to successfully discover similar sequences even for different speaking speeds. Such feature is important for converting spoken words to text via computer programs. Saved speech patterns are compared to a spoken text in order to recognize single words. By applying the DTW one can detect two words as the same even if a vowel was spoken out in a different way (shorter or longer).

In the following, the Dynamic Time Warping algorithm is introduced, as well as the reasons to use it in context of time series expression data.

Since biological processes may unfold with different rates in response to different experimental conditions or within different organisms and individuals, gene expression time series can variate not only in terms of expression amplitudes, but also in terms of time progression [39]. Figure 3.4 shows two time series profiles once exactly aligned above one another and once aligned by a more elastic non-linear method. The first plot shows that any distance metric (e.g., Manhattan, Euclidean distance, ...) would produce poor curve similarity since the i-th point of one of the time series is placed (compared) with the i-th point of the other. On the other hand, the second plot uses a more flexible alignment, allowing a more intuitive similarity measure with matching of similar shapes even if they are out of phase in the time axes.

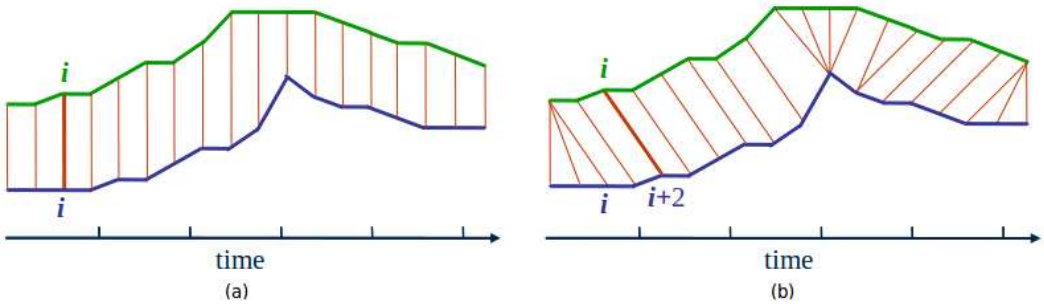


Figure 3.4: An example of a curve comparison on the basis of distance metric (a) and a more elastic alignment (b).Figures adopted from [40].

Given two time series of feature vectors: $A = (a_1, a_2, ..., a_i, ..., a_n)$ and $B = (b_1, b_2, ..., b_j, ..., b_m)$, the two series can be placed on the sides of a grid, with one on the top and the other one on the left. Both sequences start on the bottom left on the grid. A distance metric $d_{i,j}$ can be calculated for each corresponding pair (i, j) in the resulting matrix. The type of the chosen metric highly depends on the requirements of the application. In the context of time series expression data it seems more reasonable to use a correlation coefficient instead of a distance metric since it is supposed to measure the similarity in shape between two profiles. For this puspose the Spearman correlation coefficient was applied with a slight adjustment. Normally the correlation would return also such curves that possess negative correlation. A negative relationship is not of interest for the purpose of EDVis. Therefore a Spearman correlation "distance" is applied between all pairs in the grid, with $d = 1 - \rho$, where ρ is the Spearman correlation. Considering that the Spearman correlation is defined over sequences and not over points, here $d_{i,j}$ stands for Spearman distance between both sequences $(a_1, ..., a_i) \subseteq A$ and $(b_1, ..., b_j) \subseteq B$, with i, j being corresponding pairs in the grid. The Spearman distance will consequently range between zero and two. Zero stands for perfect positive relationship, one for no relationship and two for perfect negative relationship. Now, to find the degree of similarity between two series, one needs to find the path which minimizes the total distance between the sequences and calculate the cost of this path. The correlation between two sequences is introduced by the concept of a *warping path*.

Definition 3.1. *A Warping Path (WP) is a sequence* $P = (p_1, ..., p_k)$ *of tuples* $p_s = (i_s, j_s) \in [1 : n] \times [1 : m]$ *for* $s \in [1 : k]$.

A minimum warping path is consequently a warping path with minimum cost. Figure 3.5 displays the arrangement of two time series on a grid and a corresponding warping path with minimum costs.

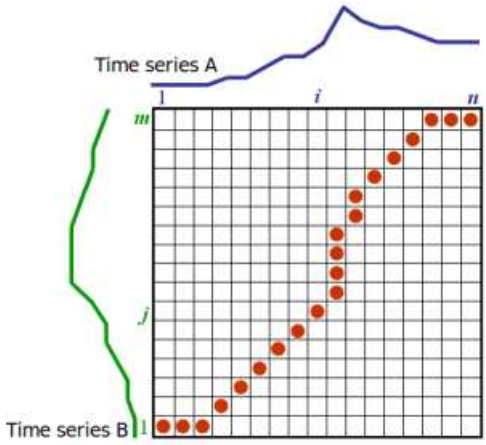


Figure 3.5: A grid with two sequences A and B, corresponding to Figure 3.4(b), placed according to the requirements of the DTW algorithm. The red circles denote the path which minimizes the total distance between A and B.Figure adopted from [40].

Finding this path involves determining all possible routes through the grid and computing the *overall distance* for each of them. Therefore the *overall distance* of the minimum warping path is the minimum of the sum of the costs between the individual elements on the path. Hence, as the length of the sequence grows, the number of possible routes can explode exponentially. Therefore the DTW algorithm proposes restrictions arising from the observations on the nature of acceptable paths through the grid outlined in Sakoe and Chiba[38]:

- Boundary condition: $p_1 = (1,1)$ and $p_k = (n,m)$. The path starts at the bottom left and ends at the top right. This guarantees that the alignment does not consider partially one of the sequences (violation of boundary condition in Figure 3.6 (a)).
- Monotonic condition: $1 \leq i_1 \leq i_2 \leq ... \leq i_k = n$ and $1 \leq j_1 \leq j_2 \leq ... \leq j_k = m$. The path does not go back in time index, both i and j indices increase

or stay the same, they can never decrease. A guarantee that features are not repeated in the alignment is given (violation of monotonic condition in Figure 3.6 (b)).

- Continuity condition: $i_s - i_{s\text{-}1} \leq 1$ and $j_s - j_{s\text{-}1} \leq 1$. The path advances one step at a time. Both i and j can only increase by at most one on each step along the path. In other words, no element of both sequences can be jumped over (violation of continuity condition in Figure 3.6 (c)).

- Warping window condition: $|i_s - j_s| \leq \theta$, where $\theta > 0$. The searched path is unlikely to be very far away from the diagonal. The distance allowed for the path to wander is the warping window width. This also guarantees that the alignment does not try to skip different features and gets stuck at familiar ones (violation of warping window condition in Figure 3.6 (d)).

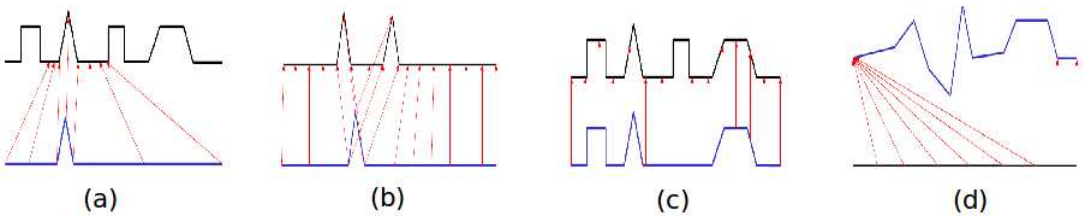


Figure 3.6: Example violations of DTW constraints: (a) Boundary condition; (b) Monotonic condition; (c) Continuity condition; (d) Warping window condition. Figures adopted from [40].

By adopting these restrictions the number of possible moves outgoing from each point in the path is being restricted, thus the number of paths that need to be considered is reduced.

Instead of finding all possible paths through the grid that satisfy the conditions, the DTW algorithm keeps track of the cost of the best route to each point in the grid, which is indeed the power of the algorithm. Therefore a cumulative cost matrix D is calculated with:

$$D(i,j) = \begin{cases} 0 & i = j = 0 \\ min\left\{D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1}\right\} + d_{i,j} & i > 0, j > 0 \\ \infty & otherwise \end{cases}$$

That is, the cumulative distance $D(i, j)$ is the sum of the distance between current elements (specified by a point) and the minimum of the cumulative distances of the neighbouring points. Since both predecessor points off the diagonal are used, the above formulation is a symmetric algorithm. Upon completion, the optional warping path can be traced back in the table by choosing the previous points with the lowest cumulative distance. Since only the minimum cost is of interest for the application, finding the warping path itself is not supported in EDVis. Then $D(n,m)$ is the minimum cost of the best warping path and can be calculated with complexity $O(m*n)$. Algorithm 1 illustrates the Dynamic Time Warping Algorithm, where $d(x,y)$ is the Spearman correlation distance.

Algorithm 1: DynamicTimeWarping

Input : Discrete sequence char a[1..n], discrete sequence char b[1..m]
Output: Similarity measure int

```
begin
    declare int D[0..n,0..m]
    declare int i,j,cost
    for i ← 1 to m do
        D[0,i] ← infinity
    for i ← 1 to n do
        D[i,0] ← infinity
    D[0,0] ← 0
    for i ← 1 to n do
        for j ← 1 to m do
            cost ← d(a[i],b[j])
            D[i,j] ← cost+ minimum(D[i−1,j]),
                          DTW[i,j−1],
                          D[i−1,j−1])
    return D[n,m]
end
```

Undoubtedly the major drawback of the algorithm is its quadratic complexity which grows with the number of time points measured. By applying the warping window constraint one can significantly reduce its running time by limiting the cells that need to be evaluated in the DTW grid. However, the DTW algorithm of EDVis does not make use of a warping window for a reason. A warping window might compromise the alignment accurasy of DTW and show a worse performance than the other methods which is not the puspose of the application.

Variations of DTW

Note that depending on which constraints are adopted for the algorithm, the result of DTW may vary. EDVis implements the classical variant of DTW supporting monotonicity, continuity and the boundary conditions. Considering that the matching performance of the algorithm may suffer if further constraints are not studied carefully, no additional conditions were adopted at this point. However, the latter does not exclude future optimizations driven by some concrete needs, such as lower complexity if extremely large datasets are studied. Some of the DTW variations are discussed in the following[41].

The continuity constraint of the DTW algorithm ensures that each element from $A = (a_1, a_2, ..., a_i, ..., a_n)$ is assigned to an element of $B = (b_1, b_2, ..., b_j, ..., b_m)$ and vice versa. A disadvantage of this condition is that a single element of one sequence can get assigned to many consecutive elements of the other sequence, which leads to vertical and horizontal segments of the warping path (Figure 3.8(a)). Thus, the warping path can get stuck at some position with respect to one sequence, corresponding to a local delay by a large factor or to a local delay by a large factor of the second sequence.

In order to avoid such unwanted effects, one can modify the continuity condition to restrict the slope of the acceptable warping paths. Recall previous continuity constraint $p_{s+1} - p_s \in \{(1,0),(0,1),(1,1)\}\ for\ s \in [1:k] \Longleftrightarrow i_{s+1} - i_s \leq 1\ and\ j_{s+1} - j_s \leq 1$ (Figure 3.7(a)). The new continuity constraint is changed to $p_{s+1} - p_s \in \{(2,1),(1,2),(1,1)\}\ for\ s \in [1:k]$ (Figure 3.7(b)), resulting to warping paths having a local slope within the bounds $\frac{1}{2}$ and 2. The new cumulative cost matrix D can be calculated with:

$$D(i,j) = \begin{cases} 0 & i = j = 0 \\ d(a_1, b_1) & i = j = 1 \\ min\left\{D_{i-1,j-1}, D_{i-2,j-1}, D_{i-1,j-2}\right\} + d_{i,j} & i > 0, j > 0 \\ \infty & otherwise \end{cases}$$

A further restriction applied by the modification of the continuity constraint is, that a warping path between two sequences A and B is defined if and only if the lengths $N = |A|$ and $M = |B|$ differ at most by a factor of two. Furthermore, it is not required for all elements of A to be assigned to an element of B and vice versa. Figure 3.8(b) illustrates the omission of elements of either sequence: a_1 is assigned to b_1, a_3 is assigned to b_2, but a_2 is not assigned to any element.

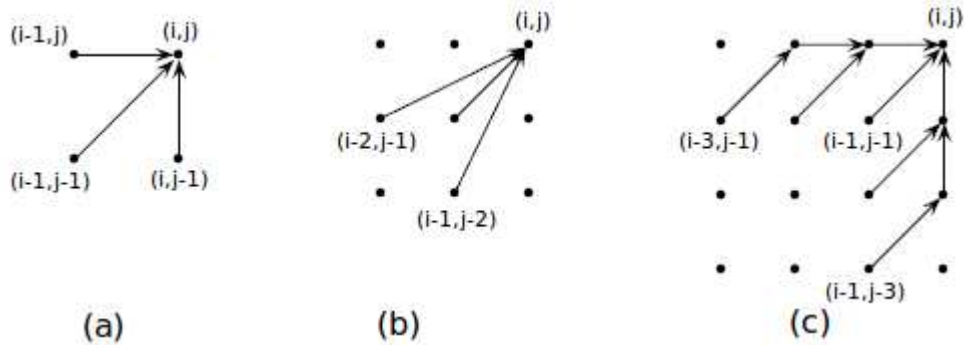


Figure 3.7: Modifications of continuity condition (a) Continuity condition of classical DTW; (b) First modification of continuity condition resulting in the omission of elements in the alignment of A and B; (c) Improved modifications with no element omission and degenarations of the warping path. Figures taken from [42].

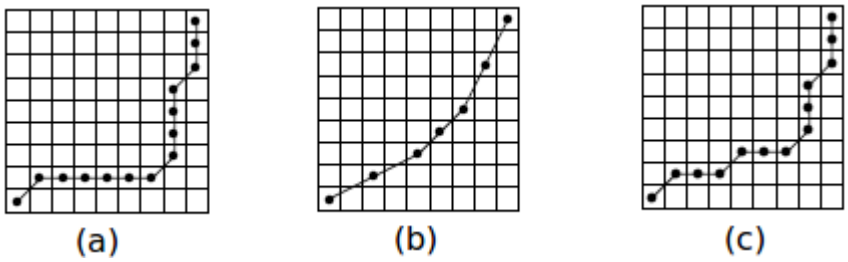


Figure 3.8: Warping paths with respect to modifications of continuity condition (a) Warping path corresponding to condition 3.7(a) with path degeneration; (b) Warping path corresponding to condition 3.7(b) with omission of elements; (c) Warping path with respect to the condition of Figure 3.7 (c). Figures taken from [42].

The omission of elements in any sequence is to be avoided since important features of both variables can be skipped and not taken into account, as a consequence the curve similarity measure of DTW may not be reliable. A more strict continuity condition is introduced in Figure 3.7 (c), which avoids such omission while suggesting constraints on the slope of the warping path. The definition for the resulting cumulative warping path is given by:

$$D(i,j) = \begin{cases} d(a_1,b_1) & i=j=1 \\ min \begin{cases} D_{(i-1,j-1)} + d_{(i,j)} \\ D_{(i-2,j-1)} + d_{(i-1,j)} + d_{(i,j)} \\ D_{(i-1,j-2)} + d_{(i,j-1)} + d_{(i,j)} \\ D_{(i-3,j-1)} + d_{(i-2,j)} + d_{(i-1,j)} + d_{(i,j)} \\ D_{(i-1,j-3)} + d_{(i,j-2)} + d_{(i,j-1)} + d_{(i,j)} \end{cases} & (i,j) \in [1:N] \times j \in [1:M] \backslash \{(1,1)\} \\ \infty & otherwise \end{cases}$$

The slopes of the warping paths resulting by the continuity condition are between $\frac{1}{3}$ and 3. This improvement enforces that all elements of A are aligned to some element of B and at the same time it excludes warping path degenerations.

As already mentioned, the main drawback of the Dynamic Time Warping algorithm is its complexity. A very effective strategy to speed up DTW is to perform

the computations on adjusted versions of the sequences A and B by reducing the lengths N and M of the sequences. One way to reduce the data rate is to process the sequences by a suitable low-pass filter followed by downsampling. Another strategy is to approximate the sequences by some function and then to perform the warping on the adapted data. A very important limitation of this solution, is that one must carefully choose the approximation depth applied on the alignment. If the approximation is chosen too fine, then the gain of speed is insignificant. On the other hand, if the approximation is chosen to be too coarse by decreasing the sampling rate of the two sequences, the resulting path may become inaccurate [41]. Figure 3.9 illustrates this problem.

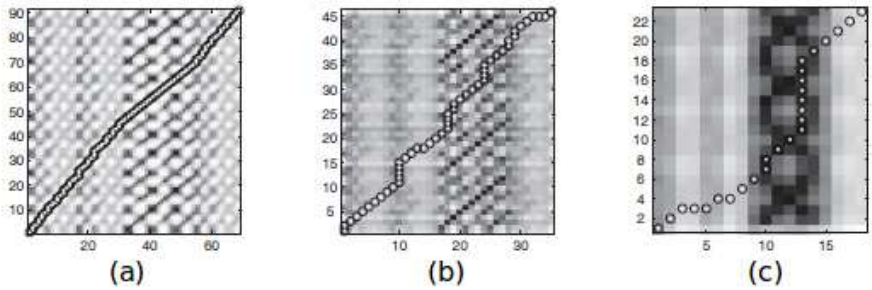


Figure 3.9: (a)Cost matrix without adjustment; (b)Cost matrix after low-pass filtering and downsampling by two; (c)Adjustement with an useless alignment. Figure taken from [42].

3.3 Web Application System Design

The aimed web application EDVis has to support the functionalities described in Section 3.1 given sets of time series expression data. These sets are available in a file form and are therefore inappropriate for the needs of a classical web application, including efficient search, read, write and update operations. Thus, a database is designed and populated with the time series data. Provided data comprises typical values measured by time series experiments (Section 1.1), such as expression values, p values and $\log_2$ ratios. Figure 3.10 shows the architecture of the Zope [43] based web application EDVis, with the common 3 tier web application architecture being adopted. The data storage layer is represented by a

MySQL [44] database- EDVisDatabase (further denoted as EDVisDB). The application logic and the representation layer are realised by a Zope Product named ZPEDVis. Further detailed description of the tool and used technologies is introduced in Section 4.

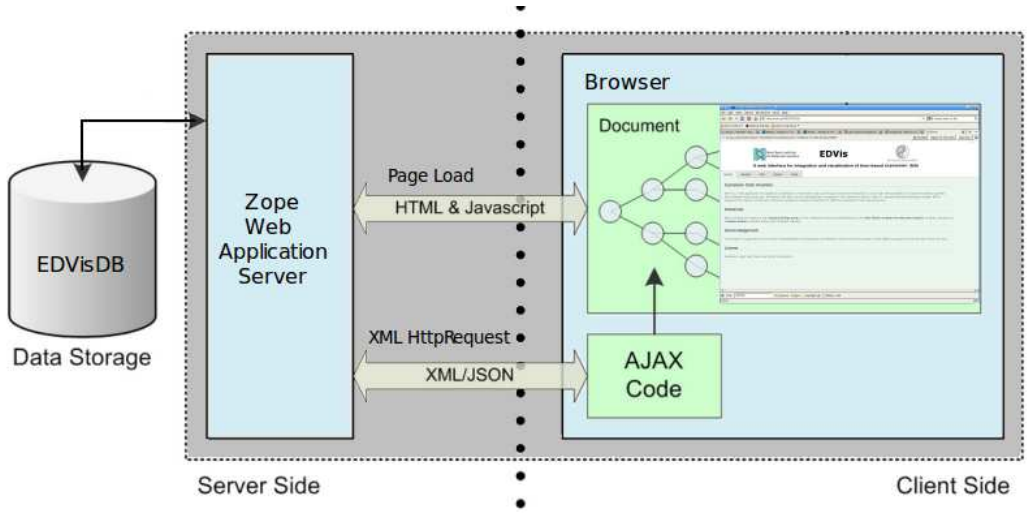


Figure 3.10: EDVis architecture

4 Implementation

4.1 Used Technologies

4.1.1 MySQL

Nowadays, there are a lot of debates about which database is the most efficient, fastest or most reliable. Experience shows that there is no correct answer to the question. The choice of database depends highly on the aimed application and its requirements. Thus, one database might be the perfect solution for an application, but inappropriate for another. MySQL has several advantages in the context of EDVis:

- Support availability: MySQL is one of the most used databases, therefore a large community, resources and books are available for the use of developers.
- Open source/Free to use
- Speed: Combined with MyISAM engine (Section 4.2.2), MySQL is lightweight and very fast.

Therefore MySQL was chosen to be used for the web application since it responds to our requirements with speed being the leading one.

4.1.2 Zope Web Application Server

Zope is a web application server written in the Python [45] programming language. It is an open source, free, object oriented application server designed for the creation of high-performance, dynamic web pages. The creation of web sites with Zope is easy and rapid. Zope gained popularity in the last years because of its speed, flexibility and power. Some of the main advantages of Zope are:

- Separation of data, logic and presentation.
- Storage of website components in objects of the Zope Object Database, unlike common file-based web server systems like ASP or PHP [46] that store websites in files. This feature allows developers to benefit from the advantages of object technologies.

- Simple user interface.
- Requires no configuration.
- Powerful user management system that can scale well to many users with different rights and privileges.
- Zope is free and open source.

Zope was chosen for the development of EDVis because of the advantages mentioned above, personal experience with the technology and the good integration in already existing platforms in the systems biology working group at the Max-Planck-Institute for Molecular Genetics.

4.2 Database Design

4.2.1 ER Model and Table Description

The database EDVisDB integrates all provided data and satisfies the requirements described in Section 3.1. Figure 4.1 depicts the defined tables and relationships. EDVisDB abstracts from the type of data in order to be able to hold information from different experimental sources (such as simulated or experimental data).

In general, experiments are grouped together in experimental groups. The table ExperimentGroup stores information about one particular group. Depending on the type of experiment, a group can be executed on a chip (e.g. in case of a lab experiment) or not, if no chip was used the corresponding field is defined as NULL. A group of experiments has optional fields of data like start date and end date of execution and a working group which made the experiments. A single experiment type belongs to exactly one experiment group, an experiment group can have many experiment types. Each experiment group has an owner and a user role as a part of the requirements for user management. The owner of the group is allowed to delete the experiment group and the user having the corresponding role is allowed to view the experimental group data.

An experiment type consists of name and time of execution. An optional field is the sample which describes the place of the measurement (nucleus, membrane, etc). Depending on the experiment type each experiment type is assigned different

types of expression data: expression values and/or p- values and/or $\log_2$ ratios. A separate table is managed for each type of experimental data.

The ExpressionValue table stores the concentrations of probes to a given time point. A reference to the Probe table is held in order to assign a probe to each measurement.

The PValue table holds information about detection p-values of probes.

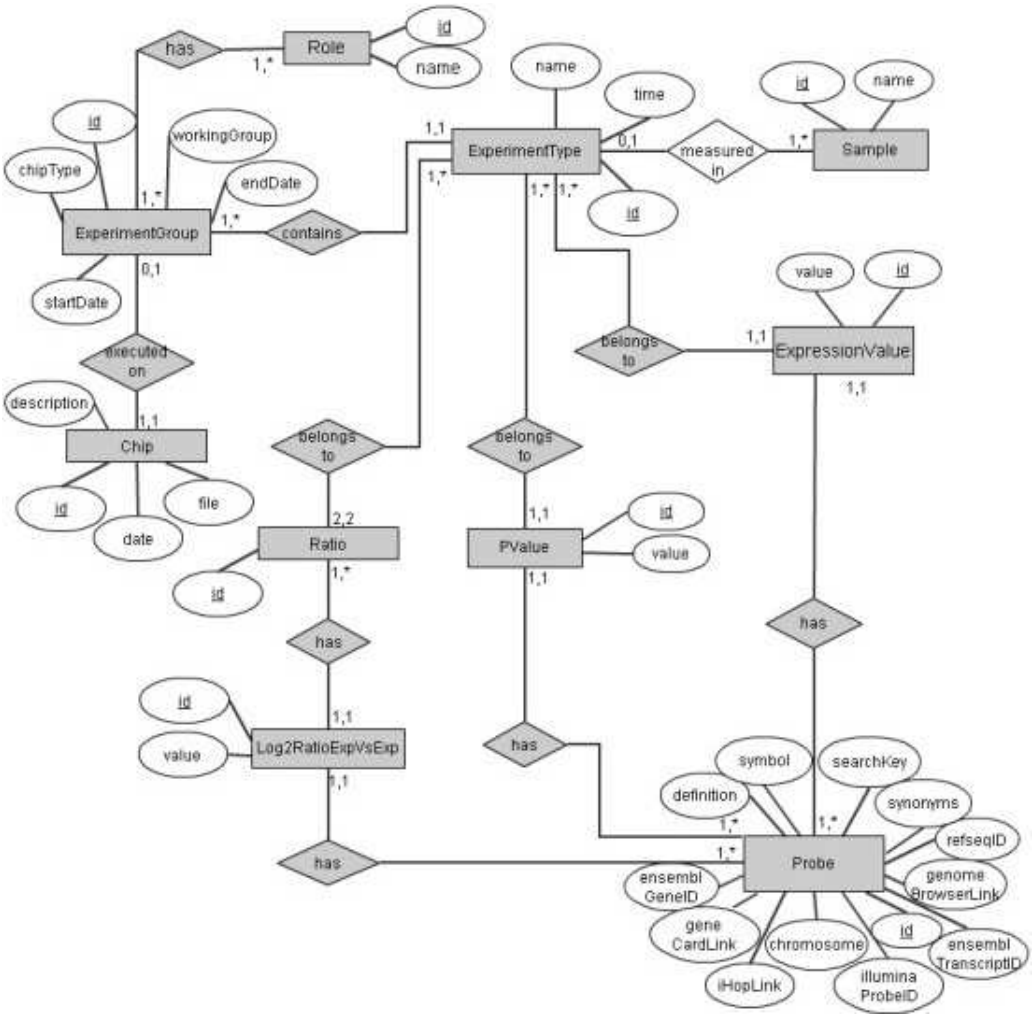


Figure 4.1: EDVisDB ER Model .

The only table of expression data having a different structure is the Log2 Ratios-

ExpVsExp table, since it holds reference to two experiments. Therefore a separate table Ratio is designed to avoid redundancy. The Ratio table stores all experiment type pairs for which a $\log_2$ ratio was calculated. The Log2RatioExpVsExp table stores a foreign key reference to the Ratio table for each table row.

The Probe table stores all measured probes with optional additional information such as links to external databases, synonyms, symbols, etc.

4.2.2 Table Engine and Indexing

One of the central requirements to EDVis is to deliver results in acceptable time. Besides optimization of MySQL queries, the choice of database storage engine is a crucial factor to performance. Table 4.1 shows a feature overview of the default MySQL engine MyISAM and InnoDB.

Table 4.1: Overview of MyISAM and InnoDB features

	MyISAM	**InnoDB**
Locking Granularity	Table level locking	Row level locking
Performance	Faster for less write operations	Faster for more write operations
Transaction Support	No	ACID Transactions, Rollbacks
Foreign Keys	No	Yes
Fulltext Indexing	Yes	No
Storage Requirements	Low disk/memory	Higher storage requirements

The choice of an appropriate database engine is not trivial and depends on the actual application and the requirements. One should weigh out the pros and cons of each engine and decide then which concept is a better solution depending on the demands of the overlying application. In the following I will introduce advantages of both engines and the arguments that led to the choice of MyISAM over InnoDB.

InnoDB is in most cases a better option since it supports transactions, foreign keys and row level locking. Supposing we have an application that has more updates, deletes and inserts than selects or such with many mixed up long lasting select queries and update queries, InnoDB would be undoubtedly the logical solution.

That way one can execute many concurrent select/update queries and benefit from the InnoDB locking mechanism. MyISAM would perform table level locking and cause long response times leading to serious performance problems. Additionally, InnoDB offers high data consistency, reliability and durability as a part of the ACID paradigma.

Despite the obvious advantages of InnoDB, MyISAM can be more appropriate in the context of some applications. It's faster in cases of many selects, supports full-text indexing and works well with default adjustments. InnoDB requires, on the other hand, tuning from expertised administrators to be most efficient.

The winning argument about the choice of MyISAM is that EDVis produces no update queries, many select queries and quite rarely insert queries. Data is uploaded once and used further in the course of events without any updates, it can be deleted eventually afterwards. The main goal is high performance ensuring quick response times over tables which contain millions of rows.

4.3 Prototype

4.3.1 Overview

The first step of the prototype implementation was the database design and data upload. The initial goal was the creation of a raw prototype of EDVis in order to test the web application and to adopt some improvement suggestions from the future users. EDVis is fully documented in English for the international use of the product.

EDVis uses the Model-View-Controller (MVC) design pattern implicitly by working with the Zope Application Server that implements MVC. MVC is a common architecture pattern, that separates data access from view and actions based on user input. Thus, easing the reuse of classes, making applications easier to maintain and test. MVC consists of three classes:

- Model: The model manages domain specific information, it responds to state queries, state changes and notifies views of changes. (Zope Product)
- View: The view is responsible of how information is being displayed. (Zope Page Template pages)

- Controller: The controller accepts and interprets user input and informs the model or the view to change their state accordingly. (also contained in the Zope Product)

At first EDVis was designed to view data from the MoGLI project. The MoGLI project is a systems biology project funded by the Bundesministerium für Bildung und Forschung (BMBF) within its research initiative "Medizinische Systembiologie" (MedSys) . Its aim is the modelling of biochemical reaction systems related to signal transduction processes and gene regulatory networks. Therefore the data was imported in the database from the command line. However, a need to generalize the tool and gather data coming from different projects and sources appeared. The manual import of data was soon proven to be time-consuming and error-prone. Thus, scripts were created for the automatic upload of data. Such import script requires a data file in a predefined format, described in the documentation of EDVis. Data can be uploaded either by an administrator or by the user (in this case the data is handled as user-specific data and cannot be viewed by everyone else).

By using the Zope Web Server EDVis is automatically clearly divided into front-end and back-end. The front-end is designed by:

- Zope Page Templates (ZPT) [47]: templates used by Zope, that can generate HTML [48], XHTML and XML [49] web pages
- jQuery [50]: a powerful JavaScript library that simplifies HTML document traversing and adds a number of dynamic events for easy and quick webpage development
- CSS [51]: Cascading Style Sheets that allow the separation between web-page content and design and ease future design changes

The back-end is programmed in the Python language .

The prototype consists of one Zope Product called ZPEDVis. The ZPEDVis folder contains all python scripts. Furthermore the product is divided into subfolders:

- zpt – contains all zpt pages
- js – contains all javascript files

- css – contains all css files
- etc – contains shell scripts
- pics – contains pictures for the zpt pages

The prototype implements all requirements listed in the concept. It can be reached under http://pybios.molgen.mpg.de/EDVis. Further functionalities can be added in the future, a more extended database search is planned to be implemented.

4.3.2 Functionalities and Layout

The EDVis web interface is divided into tabs, each tab reflecting one of the main functionalities of the tool. Profile search can be performed either through the Plot page or through the Graph page.

Home

The home page shows a short description of the tool and information about the working group and the project financing the development of EDVis. It should serve as a short introduction for those users that use EDVis for the first time and want to get more information about the application.

Plot Creation

One of the main functionalities is the creation of plot graphics of expression data. Plots have to be able to view data from up to two data resources at once. There are three data resources defined (the support of more resources is planned in the future):

- Expression Data: Gene expression data of experiments generated in laboratory
- User Data: Uploaded user data of experiments (e.g. simulated data)
- Protein Data: Protein data of experiments executed in laboratory

The available data in the database is organized in experiment groups, each of them containing many experiments (see Section 4.2 for detailed database description). Concentration measurements, p-values and $\log_2$ ratios can be stored for each experiment and its components. To create a plot graphic one has to choose components, data resource, experiment group, experiments and type of data. The user is allowed to plot data from various experiments, so that comparisons between experiments can be investigated.

First step of the creation is the selection of the components to be plotted. The search page is used to search through the database and select components from the database hits. Figure 4.2 shows a usage example for the search page.

Figure 4.2: Search page– The user can search for components by typing them in the Search List text field or by uploading a file with a list of components. Database hits appear in the Hit field and can be added to the Selection by checking one or more checkboxes and clicking on the arrow.

One can look for a component using component symbols, Ensembl Ids or synonyms. The searched elements can be typed in the designated text input field or uploaded from a tab-delimited file. A list of hits appears with corresponding information about each hit (e.g. external links to databases, component synonyms, etc.). Moreover the user gets feedback about the types of data resources that are available for each hit. One search can contain one or many components. After each search the user can choose search hits and add them to a selection.

Next step is the selection of data resources. Depending on the number of resources, plot options can vary. In case of one data resource the user is able to plot data from one, two or three experiments within the same experiment group. Plotting of more experiments is not implemented in order to keep plots clearly represented and readable. In case of two resources one can plot data from up to two experiments respectively. After having chosen the experiment resources, groups, experiments and data type, the user can perform data plotting. Curves coming from different experiments and different resources have to differ from each other, different colors and line styles are therefore used. Each plot has a color and line style legend, showing the mapping of curves to components. Each component is colored with a different color if the number of components is less than eight, otherwise all components are colored red (choosing different colors is not distinguishable). Curves belonging to different data resources or experiment types have different line types. Thus, one can easily assign a curve to a component and experiment. In addition to that, the user is able to remove curves from the plot, assuming that the graphic might not be readable if too many components were selected.

Users can change some of the plot settings. They are allowed to change time range for each plot, colors and line styles. The change of time range is an important option since experiments in different groups can be measured at different time points. Assume, we have an experiment measured only for one hour and a second one measured for 48 hours. A comparison of components in a plot with a 48 hour scala would be not very helpful. An additional option is the removal of components, supposed that the user would wish to remove some items from the plot. Optionally, plots can be deleted, zoomed or saved together with a legend. Figure 4.3 shows an example plot with corresponding experiment resource, experiment, data and plot legend.

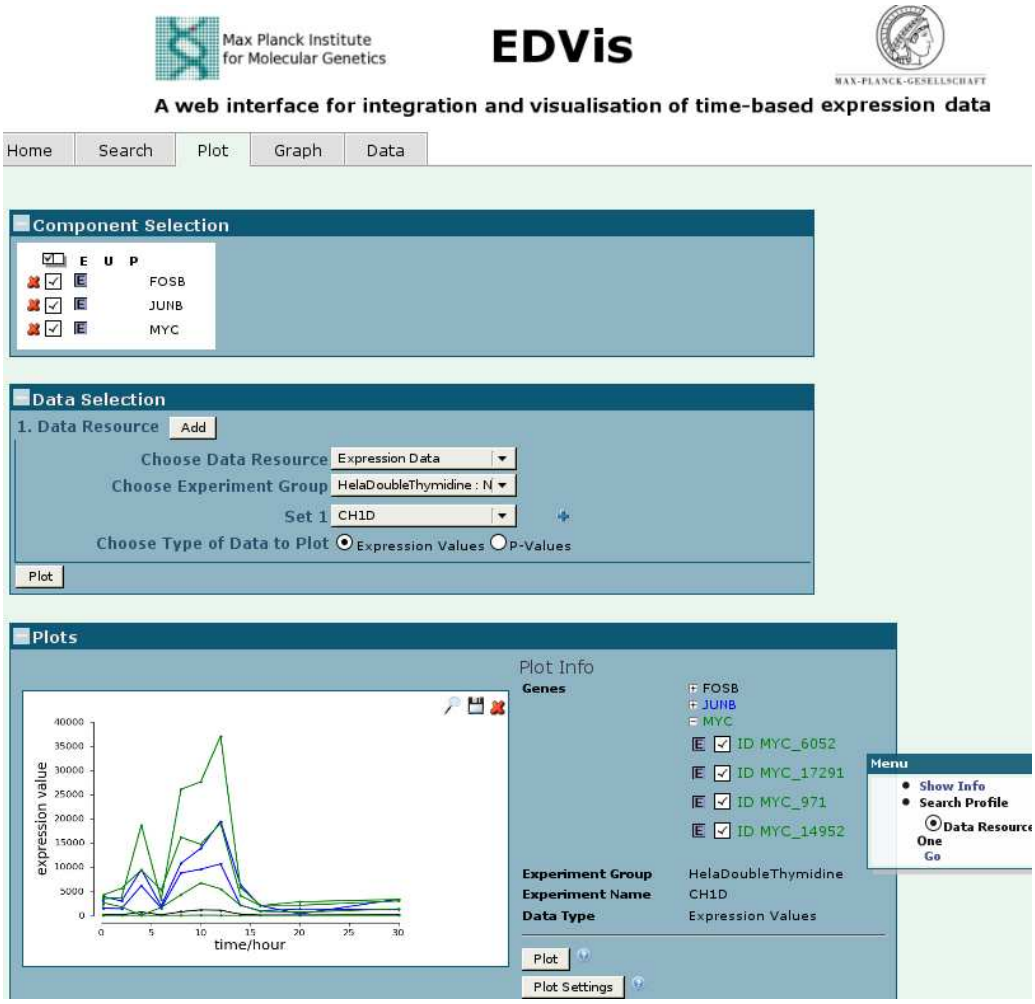


Figure 4.3: Plot page– The user can choose a data resource, experiments and type of data for the plots in the Data Selection field. The Component Selection has already been defined on the Search page. Afterwards, plots can be created by clicking on the Plot button. All plots appear in the Plots together with a legend. A profile search can be started by clicking on the icon in front of each component id.

An important feature available on the plot page is the search for similar profiles among all available ones. The profile search page is accessible over the icon in front of each component id. It's opened in a new window and contains one plot

with the chosen profile. The user can choose a criterion for the search on the right of the plot: Spearman Correlation, Euclidean Distance, Pearson Correlation or DTW Algorithm. The result of the search is sorted by rank and viewed on the right. An example with a result of a profile search is shown in Figure 4.4.

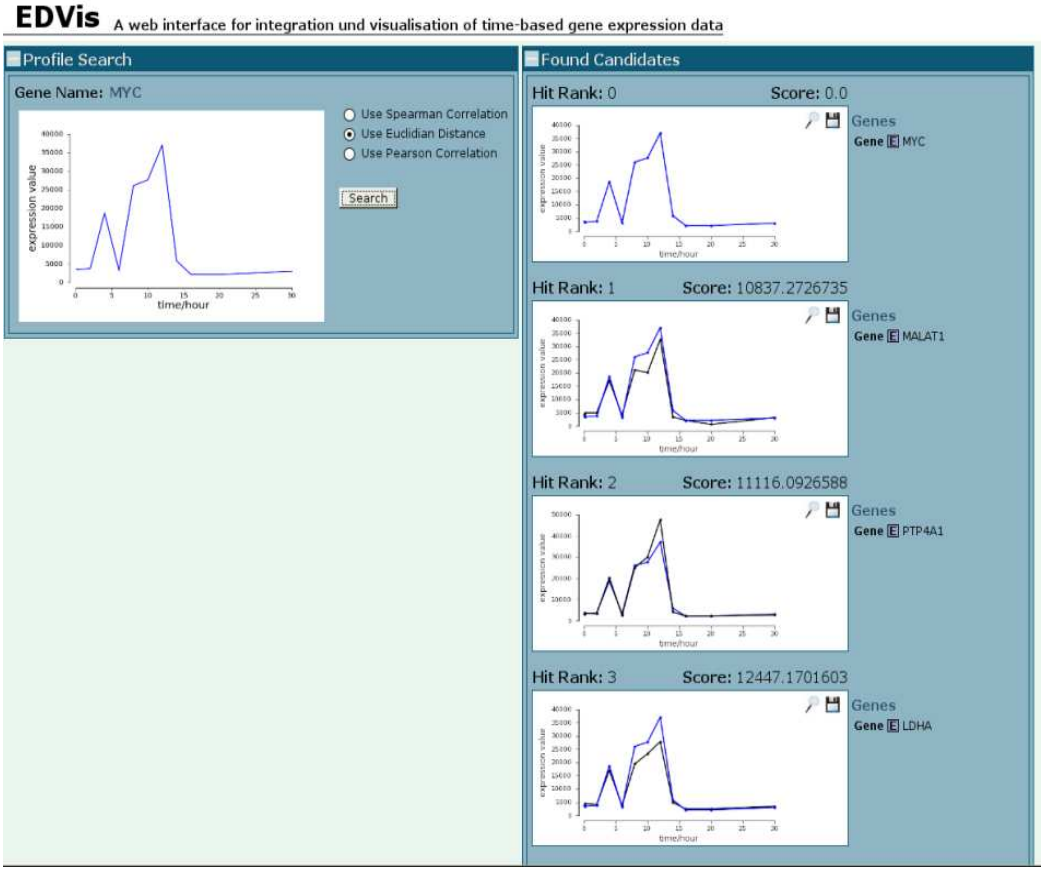


Figure 4.4: Profile search– This screenshot of EDVis shows an example of an expression profile search using the Euclidean Distance. The results on the right are sorted according to the distance to the original profile with best hits being plotted first. The example shows that this method returns curves very similar to the original profile.

Graph Visualization of Interactions

A further feature of EDVis is the graph visualization of interactions. Such graphs can be used to better comprehend complex relations between components. A graph is created either by an input mask or by an uploaded file. The graph reflects different types of components as well as different types of interactions. Additionally an input file format is defined since there is no current standard that satisfies the condition of support of all interaction types that could come into question. The graph itself is created via Scalable Vector Graphics (SVG). Such XML file format offers many advantages over most used graphical formats. To name a few, SVG :

- is scriptable, DOM-based events such as mouseover and mouseclick are supported
- can be animated
- supports hyperlinking
- is independent of resolution, the SVG graphic adjusts itself to the resolution of the output device
- is easy to create and edit
- is easy to read
- can be styled by CSS style sheets

These SVG features are of importance for the web application considering that graphs will be created, that have to be linked, colored, changed, etc. There are a few more demands that the graph should meet. Different types of components and interactions have to be distinguishable. The user has to be able to change the graph and add or remove interactions.

Moreover the graph has to be organized hierarchically. That way plot graphics consisting of the components of each hierarchy can be created beside the graph. The graph is created by the dot layout algorithm implemented by the Python graphviz [52] package. The algorithm orders nodes in layers, avoids edge crossings, minimizes edge length and aims edges in one direction (top to bottom and left to right). The layout allows to gather the nodes of a layer and create a plot for each layer. With the use of the SVG jQuery plugin the graph is styled and

manipulated. Furthermore an interaction network can comprise components from all three data resources, therefore the user can choose experiments and type of expression data for each resource.

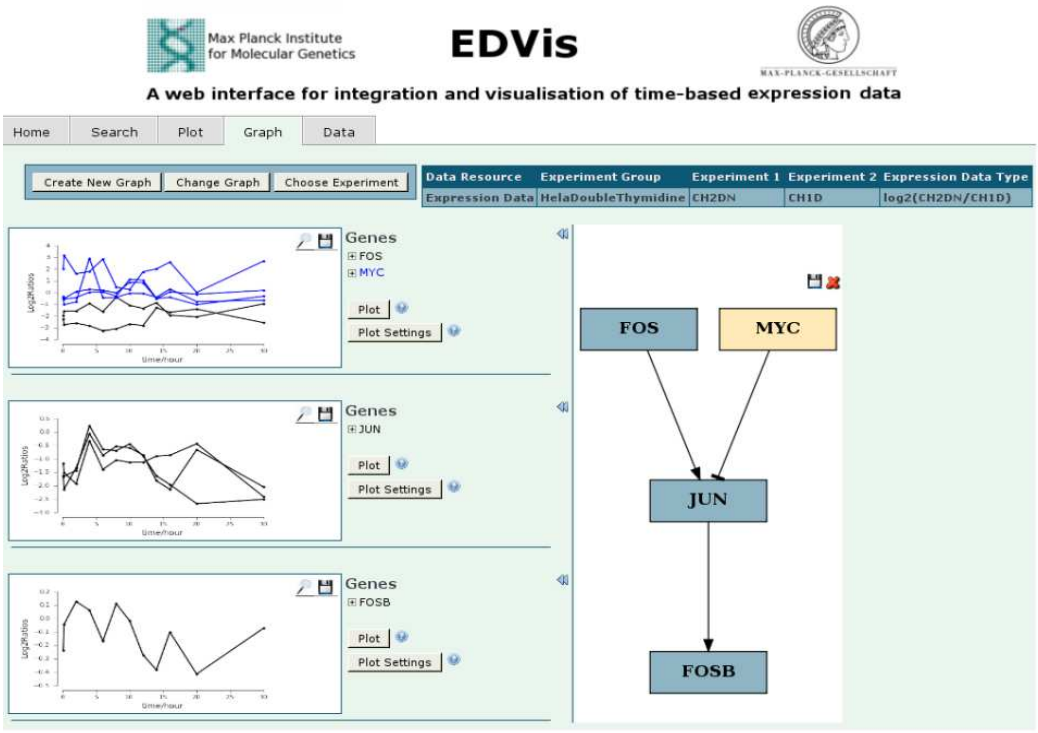


Figure 4.5: Graph page– The screenshot shows an example of a gene regulatory network defined by the user. It has been created by choosing Create New Graph. The plots on the left correspond to each layer and nodes on the layer of the graph, as well as to a chosen resource,experiment and data type. The graph can be expanded either by the user or by using the connection to CPDB in order to import interactions from the database.

The profile search is supported by the graph as well. Such search can be performed through the icons of the plot legend. A profile hit can be added to the graph providing the user is interested in adding a similar profile to the graph. The database search is carried out after a click on a graph node. The user gets a list of interactions with the node involved. Analogously, interactions can be added to

the graph and the data can be viewed in the graph plots if such is stored in the EDVis database. The network graph can be exported in the predefined file format and exchanged among users of EDVis.

User Management

EDVis supports user management according to the requirements since uploaded data is strictly secure and cannot be viewed by unauthorized persons. The Content Management System (CMS) of the Zope Server comes with User Management, Role and Rights support. Therefore only minor adjustments had to be made on the database to satisfy this requirement. The CMS can differ not only users, but also assigns roles and rights to each user. The user registration is not to be executed automatically for security reasons. An administrator has to register the new user to the system and provide a password that can be changed later on. Additionally, he has to assign roles to the user. The Zope Product has to care of the distinguishment between functionalities and roles. One functionality can be available for a certain role, but not for another. Thus, users can be divided into groups with different roles. Further user management functionalities are not required, but can be added at later point. Having added user roles to users, one can implement functionalities available only for specific roles. Currently all users can dispose of the functionalities of EDVis, the only limitation is the data viewed by each user. Therefore, the table Role was defined. Each experiment group is assigned with one or many roles, if the logged user possesses one of the roles, he is allowed to view and download the experiment group data.

Data Manipulation

To meet the needs of the end users, EDVis has to offer data manipulation: data upload and download and deletion of uploaded data.

Data can be uploaded via file with predefined format. The description of the file format appears in the documentation of EDVis. Each file corresponds to one experiment group. Data can be uploaded from a predefined file containing a component symbol, an id column and data columns for expression values, p values and $\log_2$ ratios. Such uploaded data is defined as user data and is to be viewed or deleted only by the user that has uploaded it. After having uploaded experiment groups, the user is the owner of the group and is the only one who is allowed to

access the data. These groups can be viewed and deleted only by the owners of the data.

In order to download data from the database, the user has to possess at least one role. Depending on the role, he is allowed to download data from experiment groups assigned to the particular role. Data can be downloaded for a subset of components (selection) and experiments from the same or different experiment groups.

Help

EDVis is obliged to deliver short documentation in form of help menu for the future users. The help menu appears on the upper right corner of the web application and contains short guidance on how to use the offered features and on the format of required upload files.

4.4 Optimization

In the course of implementation some problems concerning running time occurred: the database structure wasn't optimal which lead to slow queries and slow creation of plots; some complex queries had to be rewritten to get most of the performance.This chapter describes the approaches to overcome these complications.

One of the ways to optimize query performance is to rewrite nested queries with many table joins. Query optimization can be of great importance for the performance if queries are not planned properly. One and the same query can take several hours or a second if it's been planned through carefully. Nested queries are sometimes not only hard to read and understand, but also slower. However, query optimization did not bring any significant improvement (only in the millisecond range). Therefore the database design had to be changed in a way that data queries would return results in up to 1-2 seconds.

One of the database design changes that lead to significant performance improvement was the change of the id for the Probe table. The initial primary key for this table was a unique varchar id that came with the uploaded data. Such ids work well with tables containing several thousands of rows. The tables of EDVisDB comprise of millions of rows, a table join is not efficient. Instead of the slow varchar primary keys an auto increment int primary key was used. The main

concerns about using varchar primary keys were that varchar keys require more space, therefore finding matches would cost more byte comparisons (e.g. in table joins).

5 Evaluation

This section concentrates on the evaluation of different ascepts of the implemented web application.

At first the matching success of profile similarity techniques was compared on the basis of published time series experiments and derived gene clusters which are known to have similar functions and profiles. This method appeared to carry a certain bias and could not be considered reliable. Therefore, I chose to use another reliable method of evaluation by comparing the success of all metrics against Gene Ontology (GO) categories. The Gene Ontology project delivers consistent descriptions of gene products across different databases by standardizing gene and gene product attributes. GO gathers information from many databases including some of the world's major repositories for plant, animal and microbial genomes[25]. Three controlled ontologies that describe gene product properties are provided by the project. These are divided into three domains (description taken from the GO):

- cellular component– parts of a cell or its extracellular environment;
- molecular function– elemental activities of a gene product at the molecular level;
- biological process– operations or sets of operations with a defined beginning and end, pertinent to the functioning of integrated living units: cells, tissues, organs, and organisms.

Therefore, given a set of genes, one can test in which common categories a subset of the genes is present. Thus, selecting a subset with a common function. The molecular function and biological process domains were used for the evaluation since they reflect an activity, the cellular component term is not of interest for the comparison. Section 5.1 describes in detail how the evaluation was proceeded and what results were derived thereby.

Usability is one of the features of great importance for web applications, especially for those that aim profit, such as online shops. Usability tests are used to measure at which extend the user experiences the communication with a tool as satisfactory, to discover unclarity in the workflow, to detect erros, etc. A usability

[25] Visit `http://www.geneontology.org/GO.consortiumlist.shtml` to view the full list of GO members.

test has been created for EDVis and carried out by the future users of the program. The test of EDVis by incompetent users was not considered helpful and was therefore not taken into account. Furthermore the running time of central functionalities was evaluated as a part of the requirement for user-friendliness.

5.1 Comparison of Implemented Methods for Curve Similarity Definition

Hereby results of the implemented methods for curve similarity discovery, that were gathered on the basis of a published set of time series data are analysed. The template matching performance may vary considerably, depending on the method applied. The main goal is to evaluate the performance of the introduced approaches: Euclidean distance, Pearson correlation, Spearman correlation and DTW algorithm.

The set of test gene expression time series is coming from the *Saccharomyces cerevisiae* (Yeast) Identification of cell cycle-regulated genes experiment by Spellman *et al.* (1998) [53] consisting of 6223 genes. The authors, being one of the pioneers in the microarray experiment technology, have sought to create a catalogue of yeast genes which are similarly regulated within the cell cycle, in other words: groups of genes with similar functions within a group and within a period of the cell cycle. They have managed to isolate eight hundred genes which share similar expression profiles in eight clusters using the clustering algorithm of Eisen *et al.* (1999) [54] and have shown for a subset of two hundred ninety-seven genes to share common functions and common expression profiles within a group. Recall that genes sharing similar profiles are thought to share similar functions. The published genes of this subset were used by EDVis to compare the results of each implemented method in order to evaluate their matching performance.

Given eight gene clusters with known functional similarity and profile expression similarity, all genes of each cluster were used as template genes and tested for matching by each method. The number of selected best matching genes was set to the length of the corresponding cluster since the cluster length varies. Then each best matching list has been analyzed: a percentual success was computed for each method within each cluster using the harmonic mean. Figures 5.6 depict the percentual success of all adopted methods within the eight clusters. Finally, the weighted harmonic mean over all clusters for each technique was determined.

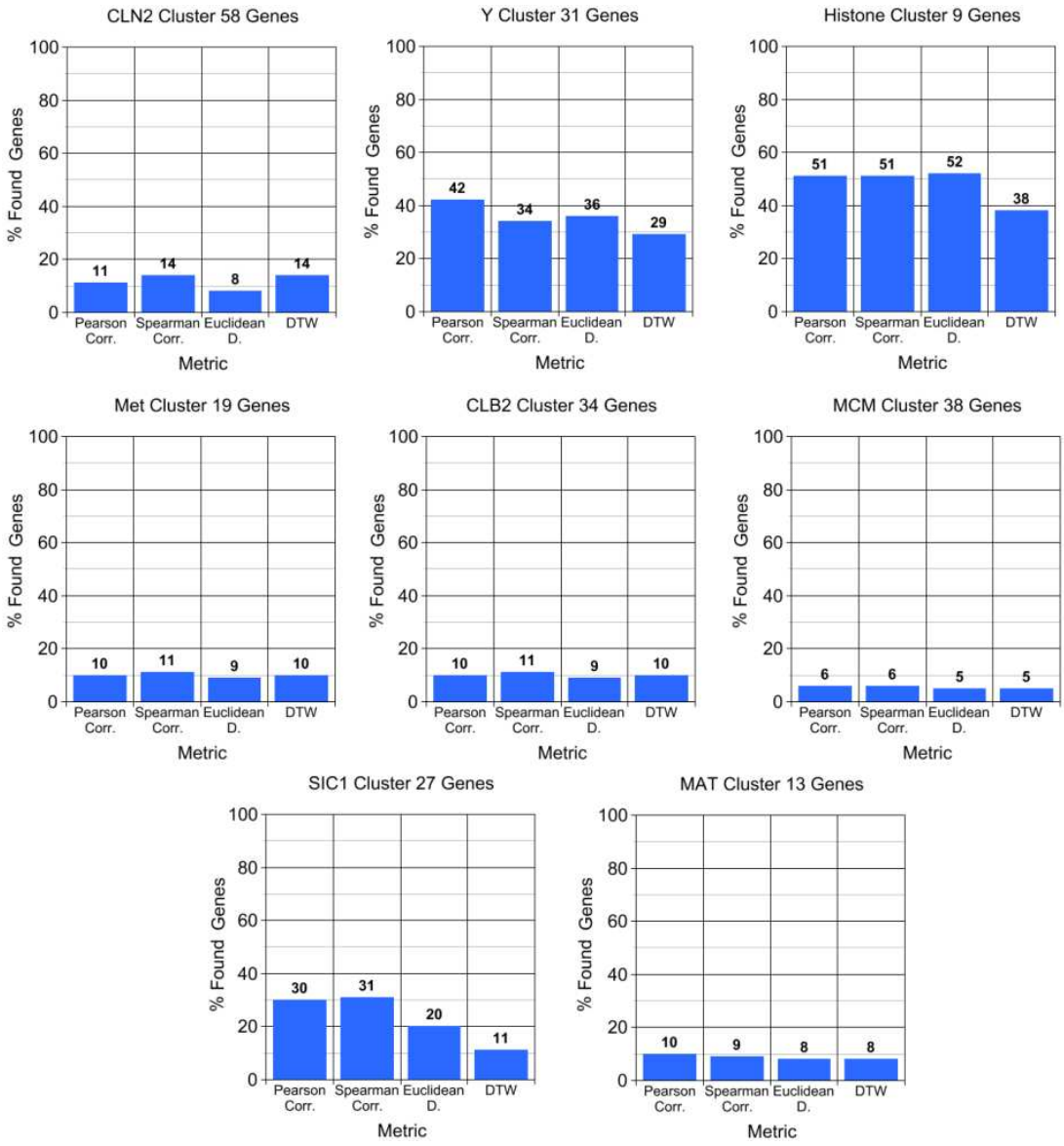


Figure 5.6: Percentage of found genes within all eight clusters for all four methods.

The use of the weighted harmonic mean is reasonable, since the length of each cluster is different. Thus, the weight taken into account is equal to the cluster size. Figure 5.7 shows the final overall success for each of the metrics.

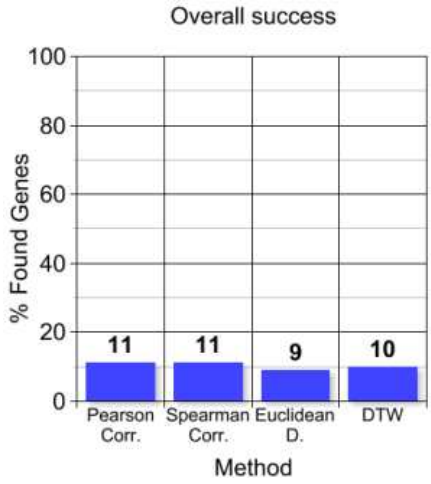


Figure 5.7: Overall hit rate for implemented metrics in all clusters.

The Pearson and Spearman correlations have been able to select most hits in each cluster with an overall success of 24% each. As expected the Euclidean distance was proven to be the worst technique for finding similar curves in the context of time series with only 10% success.

However, in the Y cluster and Histone cluster it seemed to have significant precedence. This is probably due to the fact that the genes in these clusters have very similar profiles with little “distance” between shapes. Since Pearson and Spearman correlations would consider also similar profiles which lie further from each other, its is understandable, that they would return also such curves with a high rank in contrast to Euclidean distance that would match only similar curves with the least distance. Despite the expectations of best performance, the DTW algorithm appeared to show worse performance than both Pearson and Spearman correlations. The reason seems to lie in the fact that the authors have first selected the clustered eight hundred genes with an algorithm that differs from the features of the DTW algorithm and takes into account the Pearson correlation. Although only a biologically meaningful subset of two hundred ninety-seven genes has been tested, a certain bias is still present and cannot be eliminated. In this case the advantages of the Warping algorithm cannot be shown. However, the DTW algorithm is not necessarily inappropriate in the context of time series data. An interesting observation and a possible application area would be the comparison of time series with phase shifting.

In order to remove a possible bias and to evaluate the performance of the methods in an objective manner, another approach was applied: a comparison against GO terms. A cluster of 9 genes (Histone cluster) known to have similar functionality in yeast [53] was used: HTB1, HHT2, HTA1, HHT1, HHF2, HTB2, HHF1, HTA2, HHO1. Figure 5.8 shows the parallel profile plot for Histone cluster in Spellman *et al.* (1998) [53] experiment sets described above.

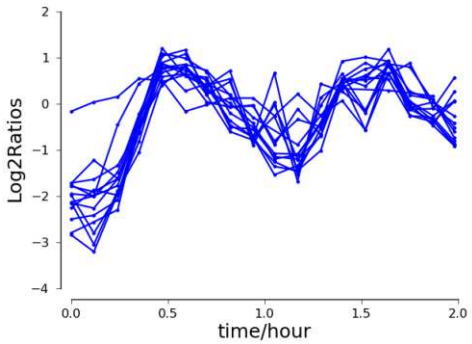


Figure 5.8: H-cluster plotted for Alpha Experiment of *Saccharomyces cerevisiae* (Yeast) Identification of cell cycle-regulated genes experiment set by Spellman *et al.* (1998) [53]

All genes of the H-cluster were used as templates and for each of them the first matching 50 genes were determined, repeatedly for each of the implemented metrics described in Section 3.2. Then the resulting hit lists were analysed by the FunSpec (Functional Specification)[26] [55] tool in order to determine which GO categories are statistically overrepresented in each list. FunSpec inputs a list of yeast gene names, and outputs a summary of functional classes, cellular localizations, protein complexes, etc. that are enriched in the list. Only functional classes were taken into account for the performed analysis since common gene functions are of interest. An example FunSpec output for gene HTB1, Spearman Correlation and DTW is introduced in Table 5.2. First column denotes found GO category together with GO id. Second column k reflects the number of selected hits in the category and third column f – the total number of genes in the category.

[26] http://funspec.med.utoronto.ca/

Table 5.2: An example output of FunSpec with GO categories for gene H1B1, Spearman correlation and DTW algorithm. The rest of the FunSpec output is not displayed at this point for clarity.

	Spearman Correlation			DTW		
Gene	Category	k	f	Category	k	f
HTB1	dolichyl-phosphate-mannose-protein mannosyltransferase activity [GO:0004169]	3	17	dolichyl-phosphate-mannose-protein mannosyltransferase activity [GO:0004169]	3	17
	microtubule motor activity [GO:0003777]	3	18	nucleotide diphosphatase activity [GO:0004551]	2	4
	mannosyltransferase activity [GO:0000030]	3	35	phosphodiesterase I activity [GO:0004528]	2	4
	transferase activity, transferring glycosyl groups [GO:0016757]	4	82	nucleoside-triphosphate diphosphatase activity [GO:0047429]	2	9
	1,3-beta-glucanosyltransferase activity [GO:0042124]	2	12	mannosyltransferase activity [GO:0000030]	3	35
	structural constituent of cytoskeleton [GO:0005200]	3	55	microtubule motor activity [GO:0003777]	2	18
	chromatin assembly or disassembly [GO:0006333]	8	28	plus-end-directed microtubule motor activity [GO:0008574]	1	1
	nucleosome assembly [GO:0006334]	8	31	chromatin assembly or disassembly [GO:0006333]	7	28
	negative regulation of transcription from RNA polymerase II promoter, global [GO:0045816]	2	5	nucleosome assembly [GO:0006334]	7	31
	microtubule nucleation [GO:0007020]	3	23	GDP-mannose biosynthetic process [GO:0009298]	2	3
	protein amino acid O-linked glycosylation [GO:0006493]	3	25	negative regulation of transcription from RNA polymerase II promoter, global [GO:0045816]	2	5
	mitotic spindle organization and biogenesis in nucleus [GO:0030472]	3	30	protein amino acid O-linked glycosylation [GO:0006493]	3	25
	protein amino acid glycosylation [GO:0006486]	3	38	protein amino acid glycosylation [GO:0006486]	3	38
	protein amino acid N-linked glycosylation [GO:0006487]	3	[illegible]	[illegible]opolymer biosynthetic process [GO:0043284]	5	13[illegible]
	biopolymer biosynthetic process [GO:0043284]	[illegible]	[illegible]	[illegible]l mannoprotein biosynthetic process	[illegible]	[illegible]

An overall success for each gene and method was calculated using the harmonic mean. Table 5.3 sums up the obtained results in hit percentage. Colored cells reflect the method with best performance for a corresponding gene, whereas a lighter color stands for no significant difference to the rest of the methods (less than 1.5% is considered not significant).

Table 5.3: Overall success of Euclidean distance, Pearson and Spearman correlations and DTW algorithm for genes of the Histone cluster.

Gene	Euclidean Dist.	Pearson Corr.	Spearman Corr.	DTW Alg.
HTB1	0.141133896261	0.105177800568	0.128267937469	0.170870626526
HHT2	0.101066816395	0.10003705076	0.0960622117181	0.091572144211
HTA1	0.180492251595	0.144583152047	0.09588952641	0.180365885081
HHT1	0.0941441778007	0.123691099476	0.154359218676	0.158544955388
HHF2	0.0921358771522	0.104241692784	0.154005989122	0.164978737212
HTB2	0.105965463108	0.112334051969	0.122116689281	0.169816406601
HHF1	0.109120611999	0.108745684695	0.110039736572	0.1227815121
HTA2	0.110890357159	0.10451122528	0.0822669104205	0.207920792079
HHO1	0.212121212121	0.368421052632	0.545454545455	0.165975103734

Recall the assumptions made in Section 3.2:

(1) DTW is expected to provide most hits among the implemented metrics or at least as many as the Spearman correlation.

(2) DTW combined with Spearman correlation is expected to perform better than Spearman correlation.

(3) Spearman correlation is at least as good as the Pearson correlation.

In 6 out of 9 of all cases the DTW algorithm managed to get the most hits confirming the assumption that the Dynamic Time Warping algorithm would return better results than any of the other techniques (1). However, in half of the cases in which DTW was best performing, the difference to the Spearman correlation result was less than 1.5% percent. Despite this fact, none of the assumptions is violated, on the contrary, assumption (2) is confirmed. Furthermore, DTW got in 7 out of 9 of all cases better results than the Spearman correlation once more confirming the thesis that DTW combined with Spearman correlation achieves better results (assumption (2)). In 6 out of 9 of all cases the Spearman correlation got more hits than the Pearson correlation (assumption (3)). The Euclidean distance turned out to get best results for HHT2 and HTA1, nevertheless with no leadership compared to the other approaches. For HHT2 the difference is less than 1% to any other method and for HTA1 the difference to DTW is only 0.01%.

The evaluation against GO terms has managed to show that the DTW algorithm is of great interest in the context of time series data. By adjusting its parameters

one can achieve different results and use the algorithm for various application areas. The systematic parameter adjustment is not a subject of the thesis and was therefore not considered in the analysis. Nevertheless, a closer observation is to be carried out at a later point. Possible use cases are noted in the discussion. Moreover, the running time of the algorithm is discussed in the following, as well as ways to speed up the performance of Dynamic Time Warping.

5.2 Usability

The concept of usability is an abstract term depending on the subjective point of view of each and every user working with a product (here referring to a software product). It is often difficult to define a software as user-friendly, but a widely used definition of usability responds to three criteria: effectiveness, efficiency and user satisfaction [56].

The requirement of product effectiveness is fulfilled if the program manages to accomplish the tasks and goals that it is supposed to achieve, e.g. a correct computation. The efficiency is related to the time that is needed to achieve the task. If the user needs to install additional software or to go through many documentation pages in order to be able to work with the software, then the efficiency of a product is in most cases evaluated negatively. Hence, the criteria for effectiveness and efficiency are closely related.

The effectiveness, the achievement of a goal, depends on the level of working progress efficiency that is experienced by the user, otherwise the process is stopped before the aim is achieved. The user satisfaction is a quite subjective term, that measures to which extend a user is satisfied with their work with a software. Design, layout and navigation play a crutial role next to the form of presentation for the user satisfaction.

According to Jakob Nielsen, a former researcher at Sun Microsystems, five users are enough to obtain best results for an usability test and everything above that is a waste of resources [57]. He defends the thesis that many test iterations are more valuable than a large number of test persons. Nowadays, his statement is broadly adopted by many researchers. What is most important about an usability test is that it is carried out at an early stage and that it is performed multiple times after each prototype change of a product. More test users would only keep discovering the same errors or issues.

Methods of usability evaluation can vary from audio/video recording including eye tracking to paper prototypes and questionnaires. The applied method depends on the goal of the software and first of all on the type of information that has to be gathered. For the purpose of EDVis, that is obtaining feedback on the product, a user survey with scenario of list of tasks was designed. The results of which are introduced in the following. The complete usability test can be found in supplementary data of the thesis. The testing group consists of future users of the software or such people familiar with the topic domain. Incompetent test persons were considered inappropriate, mainly because the software is too specific to be tested by arbitrary persons.

The users were asked to accomplish a list of tasks and to answer a list of questions afterwards. The survey ran for two weeks over which 10 responces were collected. The results of the questionnaire are introduced in the following. Some questions were grouped together and formed four main evaluation criteria that can be evaluated with respect to a level of user satisfaction, these are: workflow, feedback, ease of use and design.

Figure 5.9 illustrates the results of the survey regarding the listed criteria. 70% of all users rated the workflow rather positively, whereas several users expressed the need of more explanations, such as tooltips. All features except for the profile search could be understood, it seemed that some users did not understand how to start a profile search, indicating that this feature might need to be enhanced or a more thorough explanation on how to start it has to be provided. However, it has to be noted that some users admitted not to have read the help menu. 70% of all users expressed the need for a more detailed feedback from the system. Undoubtedly, this indicates the need of more hints, tooltips and warnings on wrong input and is an area that needs improvement. The design, as such, was rated very good from most users, whereas one user noted that the creation of large networks might become problematic in respect of clarity. That is undeniably a major issue for every vizualisation program dealing with large networks and is a research topic itself. It is definitely an important point to work on, but it could not be addressed in the course of the thesis. Nevertheless, it is a matter to be considered at a further level in the improvement of the application.

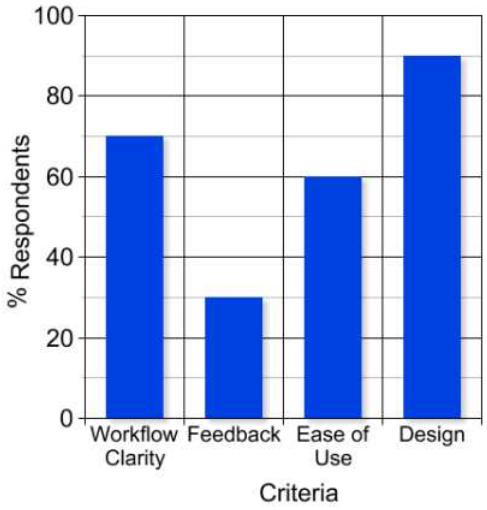


Figure 5.9: Evaluation results of usability test: four criteria are plotted on the x-axis, the y-axis represents the percentage of respondents that experienced the corresponding criterion as positive

60% of the users rated the application as easy to use, some have commented that the help menu was very well organized and easy to interpret. Others preferred to see more hints rather than reading a help menu for going through software documentations is not a common practise or is generally avoided. One thing to keep in mind is that one cannot always reasonably use a software for the first time without having to go through a help menu or documentation. Still, this remark was taken into consideration since it was mentioned numerous times. Next to rating several aspects of the web tool, the usability test was used to discover unnoticed errors and unclarity. Although the usability test discovered lack of clarity in certain areas, the overall perception was positive regarding the usefulness of the application. The test was used to improve the user-friendliness of the EDVis prototype and is going to be executed iteratively after further changes in the future.

5.3 Running Time

As acceptable response time is an inseperable part of software usability, it has to be kept in certain bounds. EDVis is required to execute most used features

in acceptable time for the user. For there are no standards in this area, there is no clear definition on what acceptable response time means. Nevertheless, the broadly embraced practice has been about the same for forty years [58]:

- 0.1 second is the limit for the user to feel that the system reacts immediately, no system feedback is necessary at this point except for displaying the results.
- 1.0 second is the limit when the user would already notice a delay, but still their attention is focused on the application and their flow of thought stays uninterrupted. Here, again no system feedback is necessary, although the closer the response time gets to 1.0 the more does the user lose the feeling of operating directly on the data.
- 10 seconds is the limit for users to stay focused on a task. Crossing that limit means that the user will perform other tasks while waiting for the computer to finish. So, a feedback on when the computer is going to be ready is required, since users will be notified on what to expect.

For operations taking more than 10 seconds it is reasonable to provide a running progress feedback, e.g., in a form of a status bar. This way the user is informed for the approximate time needed for the computer to accomplish a task or for the absolute amount of work done. In cases where it is not known how much work has to be done and no prediction on response time can be made, it is still useful even highly required to provide a less specific progress indicator, such as a spinning ball or any sign that indicates that the system is working.

Considering these recommendations, EDVis was developed correspondingly. Simple operations like searching, adding to a list, removing elements of a list were kept in the bounds between 0.1 and 1 second. More complex tasks such as the creation of a plot and a graph, that are expected to take more time were optimized to last for as short as possible, whereas a progress indicator is provided. Such operations depend mainly on the speed of a database query. Section 4.4 has already referred to that problem, hence, the issue is not further discussed. After database query optimization, the plot creation has become noticeably faster and lies between 0.40 and 0.47 seconds for plots containing one to ten genes. Note that plots containing more than ten genes are not expected to be created often since they are not representable and are difficult to interpret. The same applies to the network graph. Considering that a reasonable network, that is to be observed in detail by

a user may not span hundreds of nodes, the test for graph creation ran on up to 64 nodes.

The running time needed for plot and graph creation was investigated via numerous tests with variable number of genes. EDVis is started and running on a machine with an eight quad CPU (Quad-Core AMD Opteron(tm) Processor 2354 @ 2,2 GHz). The plot creation was tested with ten iterations on randomly chosen experiments for a growing number of genes. Figure 5.10 (a) illustrates the average time needed for the creation of a plot with the growth of plotted genes. Figure 5.10 (b) depicts average time needed for the creation of a graph with the growth of nodes. Note that time needed for the browser to display results may vary depending on the user web browser. Since no prediction can be made on that matter, the browser time is kept out of consideration. Obviously, both tests have shown that the plot and graph creation are kept in more than satisfactory time bounds: below one second for the common case.

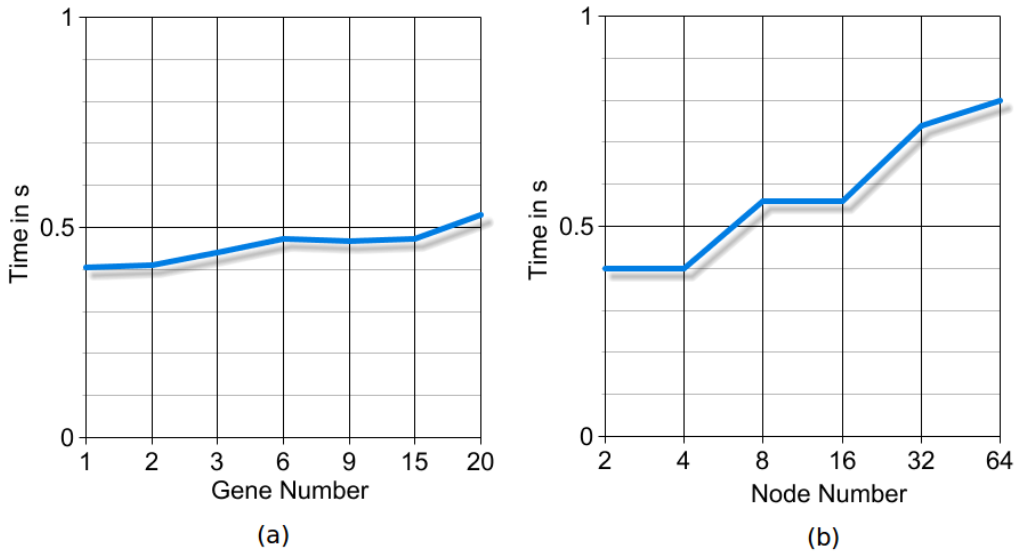


Figure 5.10: Running time of (a) plot creation and (b) graph creation

Furthermore, the running time of all profile search methods was investigated.

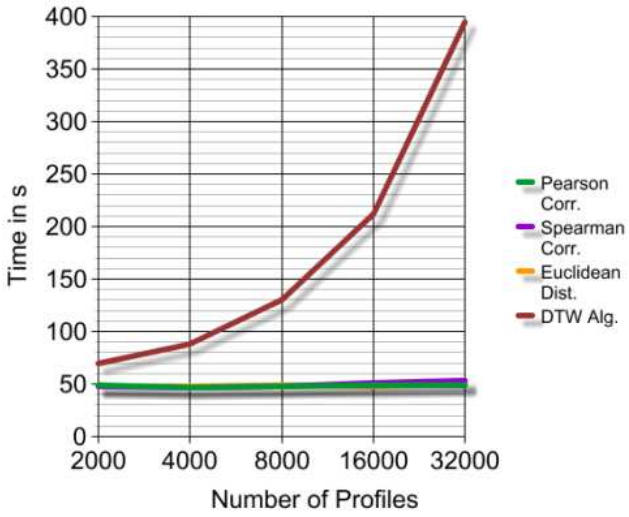


Figure 5.11: Worst case running time of profile search.

Figure 5.11 shows the time needed for each method to complete calculation in the worst case with the growth of compared profiles and for experiments with the most measured time points.

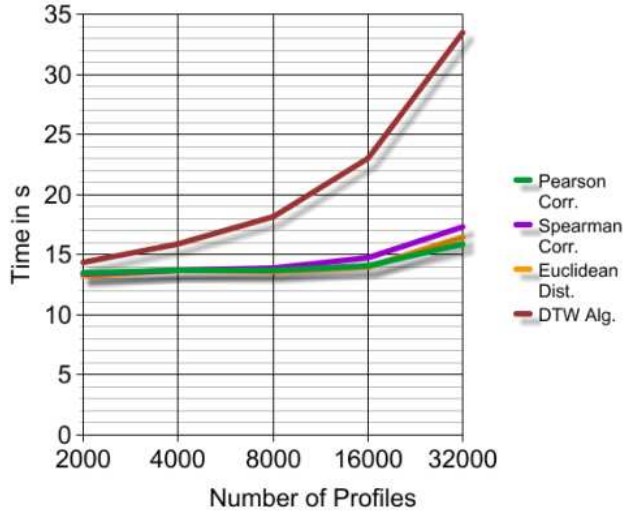


Figure 5.12: Best case running time of profile search.

The same investigation was performed for the best case, with the least number of measured time points. Results are shown in Figure 5.12. Note that the database query was also added to the calculation. As expected the DTW algorithm needs at most time to complete for its complexity grows quadratically. It lies in $\mathcal{O}(V.m.n)$, where m and n are the lengths of compared sequences and V is the overall number of alignments (number of gene profile comparisons). Thus, with the growing number of compared profiles and a large number of measured time points, the running time of DTW rises significantly, whereas the running time of all other methods grows linearly. This can be seen in both graphics: Pearson correlation, Spearman correlation and Euclidean distance increase only slightly their running time with the growth of compared profiles, at the same time the curve of DTW grows obviously a lot faster. DTW needed around 6 minutes to terminate for the comparison of 32000 profiles with 13 time points in the worst case and half a minute for the same number of profiles and 4 time points. The other methods needed only approximately 50 seconds and 17 seconds in worst and best case respectively. At this level no difference between the running time of the correlation metrics and the Euclidean distance is made for it is insignificant in comparison to DTW. One can easily recognize the major drawback of the algorithm–namely its complexity.

Keeping in mind the amounts of data that have to be calculated, the fact that profile comparison is not frequently executed and that not many experiments use a large number of measured time points since this is a major expense factor, such disadvantage can be accepted. Furthermore, the results of the calculation stay in foreground, not the running time. Section 3.2.4 discussed this issue and the ways to optimize the running time of the algorithm. However, considering the fact that results may be influenced, the alternatives were not taken into account yet.

6 Discussion

This section discusses possible application domains of the implemented web tool next to biological time series experiments. Furthermore, future improvement features are introduced, that would bring new functionality to EDVis and help with the evaluation of experiment data.

6.1 Related Problem Domains and Portability

Although EDVis was developed for the use in biological research, the tool can be applied in other domains with similar success by adjusting some of its components. In the following, I will discuss other scenarios for the application of EDVis.

As the conceptual methods adopted in the development of EDVis do not constrain the data on which they can be applied, the tool is theoretically applicable for different kinds of experiments producing large amounts of time series data. Certainly, the front-end of the application has to be adjusted, as well as maybe some of the underlying database tables. The tool can be adapted with little effort to any type of time series data in order to proceed evaluation where the adopted similarity methods are also of interest.

6.1.1 Validation of simulated data

Consider weather prediction validation with the help of the proposed analysis methods of EDVis. Weather forecasts generate large amounts of predicted time series data, such as solar activity, rain amounts, wind strength. Such forecasts are generated with carefully designed numerical weather prediction systems based on simulations. In order to validate the reliability of a forecast, simulated data is compared to really observed weather parameters measured in time [59]. It is an usual practise to compare these time series with a correlation metric, such as Pearson correlation. The higher the correlation, the more reliable is one model and the generated prediction.

In general, EDVis is in the position to validate different kinds of simulated time series data against observed data using either of the methods I have proposed within my thesis. The applied method would of course depend on the distribution of the

data. The only condition is that the time series are organized in the predefined table format, so that they can be smoothly uploaded to the database.

6.1.2 Prediction

One of the main features of EDVis next to the proposed validation of simulated data is the prediction using correlation methods, Euclidean distance or the DTW algorithm. In the context of biology, the application can predict possible gene function by comparing profile courses. A high degree of profile similarity would mean possible similarity of gene function. The same principle can be used in any domain with similar observations.

Suppose, the concentrations of differents gases and gas compounds in the atmosphere is measured in time. It is known that a higher concentration of CO_2 regulates temperature positively, assume that we are interested in finding other gases that have the same function, namely a rased concentration of the gas is related to higher temperature and a lower concentration – to lower temperature. One can compare the concentration profiles of all gases measured in the same time interval and make the assumption about gases highly correlated with CO_2, that they are also greenhouse gases. This can be achieved via the profile search function with one of the implemented similarity measures in EDVis.

A study has already proven the correlation between CO_2, CH_4 and earth temperature. It was derived from multiple ice cores that record atmospheric conditions and climate for the last 650,000 years. Figure 6.1 shows the obvious correlation between the gases and earth temperature in the past 600,000 years. The black center line measures deuterium (a hydrogen isotope that is a stand-in for historical temperature). The red line shows the profile of CH_4 in time and the blue line – the time course of CO_2.

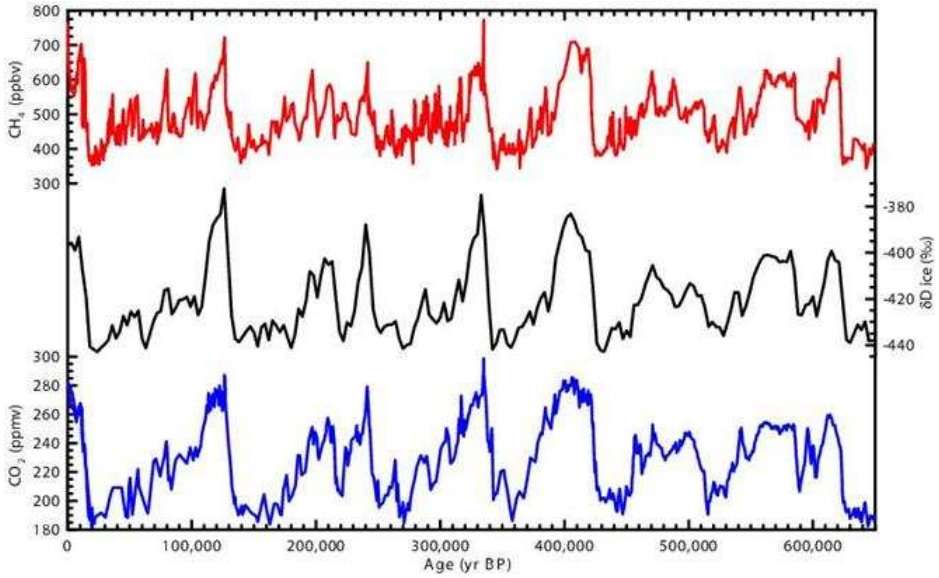


Figure 6.1: Correlation between CO_2, CH_4 concentrations and temperature[60].

6.2 Future Work

Although the requirements set to EDVis to fulfill have been accomplished, there is still room for improvement in terms of new features and enhancements to the system.

6.2.1 Adjustable Parameters of Dynamic Time Warping Algorithm

Consider the Dynamic Time Warping algorithm. For it can be modified via several parameters, one can use it in different application fields. One parameter is the distance metric. The implementation of DTW realized in the thesis uses the Spearman distance, but many other distance measures can be studied and applied: such as Manhattan and Euclidean distance, Pearson and Chebychev correlation.

Another adjustable parameter is the warping window constraint. With the evolution and modernization of technology, time series experiments are expected to become cheaper. Data size, i.e. length of a time series is still limited by the cost of experiments. In the future, the measurement of more time points may become

a common practise, which will cause even larger data sets to evaluate. Recall the complexity of DTW, its running time will grow as a consequence. The warping window constraint or a properly adjusted continuity condition can be used in that case as discussed in 3.2.4 for the faster processing of larger data sets. However, one has to keep in mind, that the extreme usage of this possible feature may have a negative effect on the accuracy of the final result. Therefore closer studies on the size of the warping window are required.

Another adjustable parameter of DTW that was not mentioned by now is an offset parameter. This parameter allows the sliding of time series against each other on the time axes. By now EDVis supported the search of similar time cources within one and the same experiment, that is within the same species. Assume, that given a gene in one species, one attemps to identify a set of genes in another species with a potentially similar function. Many biological processes are conserved between species, e.g. the cell cycle. Nevertheless, the duration of different cell phases may differ between species. One way to correct this is to position corresponding cell phases against each other by sliding them via a defined offset. This approach called a "causality search" has been proposed in [61]. By specifying a non-zero offset one may slide the sets of expression profiles against each other along the time axes, thus identifying genes with similar curves, but shifted in time. The technique can be used not only between time series of different species, but also between time series of different individuals of one species, since different individuals affected by a common disease may progress at varying rates.

6.2.2 Further Improvements

Next to enhancements on the Dynamic Time Warping, improvements regarding the network graph are planned. Although the chosen display design of the network graph is very convenient for relatively small networks (for a couple of tens of nodes), a display of hundreds of nodes becomes problematic. Considering the layered hierarchy of the network, and the fact that time cources of one layer are plotted together in one plot, a growing number of nodes lying on one level would cause a not interpretable plot. A more flexible way to display large networks has to be carefully considered. One option is to use the same network hierarchy, but to display a separate plot for each node on user's behalf. The advantage of this method would be on first place, that only curves of interest will be created and not all time courses have to be plotted as is the case now. Second, the overall display clarity is improved. The disadvantage is that the curves of one level can no longer

be compared in one plot. For this purpose one will have to create comparative plots separately without parallel display of the network. Further disadvantage is that plots have to be created separately one by one by the user. Still, one can combine both approaches and let the user choose which display method they prefer.

Certainly, in the process of application of the tool, a row of further improvement and enhancement suggestions will occur.

7 Conclusion

Time series experiments measuring the expression of thousands of genes simultaneously provide a method of high-throughput data collection necessary to obtain the scope of data required for understanding the complexities of living organisms. The visualization and analysis of the collected data play a crutial role for the correct data interpretation.

EDVis is a web application that enables the integration and visualization of time series data in form of time course plots and regulatory networks. The tool available at http://pybios.molgen.mpg.de/EDVis, also provides methods for data evaluation by implementing several tecniques for time course comparison: Euclidean distance, Pearson and Spearman correlation and Dynamic Time Warping algorithm. Finding genes with similar profiles, leads possibly to predicting new gene function, since genes with similar time courses are believed to be functionally related. This discovery helps in bringing new enlightment in the field of biological observations and further investigation decisions.

The evaluation of the time course comparison approaches led to the result that Dynamic Time Warping returns most accurate output, but however its complexity is a problem to consider. Its running time can be accepted for time series of current researches, but with the growth of measured time points it may become a problem to address. This issue as well as further enhancements are to be inspected for further improvement and adaptation of the web application.

References

[1] Ziv Bar-Joseph, et al. (2003). *Comparing the continuous representation of time-series expression profiles to identify differentially expressed genes.* Proc. Natl Acad. Sci. USA, 100, 10146–10151.

[2] Singh, R., Palmer, N., Gifford, D., Berger, B., and Bar-Joseph, Z. (2005). *Active learning for sampling in time-series experiments with application to gene expression analysis.* ACM, 119, 832-839.

[3] Ziv Bar-Joseph (2004). *Analyzing time series gene expression data.* Bioinformatics 20(16), 2493-2503

[4] Sadava, et al. (2008). *Life: The Science of Biology, Eighth Edition.*

[5] Levine, M. & Tjian, R. (2003). *Transcription regulation and animal diversity.* Nature, 424, 147-151

[6] Wierling C, et al. (2007). *Resources, standards and tools for systems biology.* Brief. Funct. Genomic Proteomic, 6, 240–251.

[7] Edward R. Dougherty et al. (2007). *Genetic Regulatory Networks.* EURASIP Journal on Bioinformatics and Systems Biology, 2007: 17321.

[8] Fujita et al. (2007). *Modeling gene expression regulatory networks with the sparse vector autoregressive model.* BMC Systems Biology, 1, 39.

[9] Dianne Cook, Heike Hofmann, Eun-Kyung Lee, Hao Yang, Basil Nikolau, Eve Wurtele (2007). *Exploring Gene Expression Data, Using Plots.* Journal of Data Science, 5, 151-182.

[10] Yvonne E. Pittelkow, Susan R. Wilson (2007). *h-Profile plots for the discovery and exploration of patterns in gene expression data with an application to time course data.* BMC Bioinformatics, 8, 486.

[11] Colin S. Gillespie, Guiyuan Lei, Richard J. Boys, Amanda Greenall, Darren J. Wilkinson (2010). *Analysing time course microarray data using Bioconductor: a case study using yeast2 Affymetrix arrays.* BMC Research Notes, 3, 81.

[12] Shannon,P. et al. (2003). *Cytoscape: a software environment for integrated models of biomolecular interaction networks.* Genome Res., 13, 2498–2504.

[13] GML: A portable Graph File Format, Technical Report, http://www.infosun.fim.uni-passau.de/Graphlet/GML/gml-tr.html.

[14] XGMML Schema, Technical Report, http://www.cs.rpi.edu/~puninj/XGMML/DOC/xgmml_schema.html

[15] Dahlquist,K.D. et al. (2002). *GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways.* Nat Genet., 31, 19–20.

[16] Keseler,IM et al. (2005). *EcoCyc: a comprehensive database resource for Escherichia coli.* Nucleic Acids Res., 33, D334-1.

[17] Kanehisa,M. and Goto,S. (2000). *KEGG: kyoto encyclopedia of genes and genomes.* Nucleic Acids Res., 28, 27–30.

[18] Waugh et al. (2000). *PathDB: A metabolic database with sophisticated search and visualization tools.*In Proc. of Plant & Animal Genome VIII Conference, San Diego, CA.

[19] Jenssen,T. et al. (2001). *A literature network of human genes for high-throughput analysis of gene expression.* Nat Genet., 28, 21–28.

[20] Hu,Z. et al. (2007). *VisANT 3.0: new modules for pathway visualization, editing, prediction and construction.* Nucleic Acids Res., 35, W625–W632.

[21] Jon Ferraiolo, Fujisawa Jun, and Dean Jackson. Scalable vector graphics svg 1.1 specication. Technical report, http://www.w3.org/TR/SVG/ W3C.

[22] Paquette J, Tokuyasu T. (2010). *EGAN: exploratory gene association networks.* Bioinformatics 26(2), 285-6.

[23] David R. Bickel (2004). *Degrees of differential gene expression: detecting biologically significant expression differences and estimating their magnitudes.* Bioinformatics 20(5), 682-8.

[24] Atanas Kamburov, Christoph Wierling, Hans Lehrach, Ralf Herwig (2008). *ConsensusPathDB–a database for integrating human functional interaction networks.* Nucleic Acids Res., 37, D623–D628.

[25] Vastrik,I., D'Eustachio,P., Schmidt,E., Joshi-Tope,G., Gopinath,G., Croft,D., de Bono,B., Gillespie,M., Jassal,B., Lewis,S. et al. (2007). *Reactome: a knowledge base of biologic pathways and processes.* Genome Biol., 8, R39.

[26] Romero,P., Wagg,J., Green,M.L., Kaiser,D., Krummenacker,M. and Karp,P.D. (2005). *Computational prediction of human metabolic pathways from the complete human genome.* Genome Biol., 6, R2.

[27] Kerrien,S., Alam-Faruque,Y., Aranda,B., Bancarz,I., Bridge,A., Derow,C., Dimmer,E., Feuermann,M., Friedrichsen,A., Huntley,R. et al. (2007). *IntAct–open source resource for molecular interaction data.* Nucleic Acids Res., 35, D561–D565.

[28] Salwinski,L., Miller,C.S., Smith,A.J., Pettit,F.K., Bowie,J.U. and Eisenberg,D. (2004). *The Database of Interacting Proteins: 2004 update.* Nucleic Acids Res., 32, D449–D451.

[29] Chatr-Aryamontri,A., Ceol,A., Palazzi,L.M., Nardelli,G., Schneider,M.V., Castagnoli,L. and Cesareni,G. (2007). *MINT: the Molecular INTeraction database.* Nucleic Acids Res., 35, D572–D574.

[30] Mishra,G.R., Suresh,M., Kumaran,K., Kannabiran,N., Suresh,S., Bala,P., Shivakumar,K., Anuradha,N., Reddy,R., Raghavan,T.M. et al. (2006). *Human protein reference database–2006 update.* Nucleic Acids Res., 34, D411–D414.

[31] Breitkreutz,B.J., Stark,C., Reguly,T., Boucher,L., Breitkreutz,A., Livstone,M., Oughtred,R., Lackner,D.H., Ba¨ hler,J., Wood,V. et al. (2008). *The BioGRID Interaction Database: 2008 update.* Nucleic Acids Res., 36, D637–D640.

[32] Elkon,R., Vesterman,R., Amit,N., Ulitsky,I., Zohar,I., Weisz,M., Mass,G., Orlev,N., Sternberg,G., Blekhman,R. et al. (2008). *SPIKE–a database, visualization and analysis tool of cellular signaling pathways.* BMC Bioinformatics, 9, 110.

[33] Luigi Cerulo, Charles Elkan, Michele Ceccarelli (2010). *Learning gene regulatory networks from only positive and unlabeled data*. BMC Bioinformatics, 11:228

[34] Viet-Anh Nguyen, Pietro Lió (2009). *Measuring similarity between gene expression profiles: a Bayesian approach.* BMC Genomics, 10(Suppl 3),S14.

[35] H. Lohninger (2006). *Fundamentals of Statistics*. Retrieved September 10,2010 from http://www.statistics4u.info/fundstat_eng/cc_corr_coeff.html.

[36] H. Lohninger (2006). *Fundamentals of Statistics*. Retrieved September 10,2010 from http://www.statistics4u.info/fundstat_eng/cc_corr_spearman.html.

[37] Altman D.G. (1991). *Practical Statistics for Medical Research.* Chapman & Hall, London, p. 285-288.

[38] Sakoe,H. and Chiba, S. (1978). *Dynamic programming algorithm optimization for spoken word recognition.* IEEE Trans. on Acoust., Speech, and Signal Process., ASSP 26, 43-49.

[39] Jo Criel, Elena Tsiporkova (2005). *Gene Time Expression Warper: a tool for alignment, template matching and visualization of gene expression time series*. Bioinformatics 22 (2): 251-252.

[40] Elena Tsiporkova . *Dynamic Time Warping Algorithm for Gene Expression Time Series*. Power point presentation. Retrieved September 11,2010 from http://www.psb.ugent.be/cbd/papers/gentxwarper/DTWAlgorithm.ppt.

[41] Müller, M., Mattes, H., Kurth, F. (2006). *An Efficient Multiscale Approach to Audio Synchronization.* Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR), Victoria, Canada, pp. 192-197.

[42] Müller, M. (2007). *Information retrieval for music and motion.* Springer-Verlag. ISBN: 978-3-540-74047-6.

[43] The Zope2 Book, Manual, http://docs.zope.org/zope2/zope2book/index.html.

[44] MySQL 5.1 Reference Manual, Manual, http://dev.mysql.com/doc/refman/5.1/en/.

[45] The Python Language Reference, Manual, http://docs.python.org/reference/.

[46] Php : Hypertext preprocessor. Technical report, http://www.php.net/.

[47] The Zope2 Book, Appendix C: Zope Page Templates Reference, Manual, http://docs.zope.org/zope2/zope2book/AppendixC.html.

[48] Dave Raggett, Arnaud Le Hors, Ian Jacobs, HTML 4.01 Specification, Technical Report, http://www.w3.org/TR/1999/REC-html401-19991224/.

[49] Xml schema, Technical report, http://www.w3.org/XML/Schema.

[50] jQuery Documentation, Manual, http://docs.jquery.com/Main_Page.

[51] Wendy Chisholm, Gregg Vanderheiden, Ian Jacobs, CSS Techniques for Web Content Accessibility Guidelines 1.0, Technical Report, http://www.w3.org/TR/2000/NOTE-WCAG10-CSS-TECHS-20001106/.

[52] http://www.graphviz.org/.

[53] Spellman et al. (1998). *Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization.* Mol Biol Cell, 9, 3273-97.

[54] Eisen et al. (1998). *Cluster analysis and display of genome-wide expression patterns.* Proc. Natl. Acad. Sci. USA, 95, 14863–14868.

[55] Mark D. Robinson, Jörg Grigull, Naveed Mohammad,Timothy R. Hughes (2002). *FunSpec: a web-based cluster interpreter for yeast.* BMC Bioinformatics, 3:35.

[56] Sven Heinsen, Petra Vogt (2004). *Usability praktisch umsetzen.* Hanser, ISBN 3-446-22272-3.

[57] http://www.useit.com/alertbox/20000319.html.

[58] Miller, R. B. (1968). *Response time in man-computer conversational transactions.* Proc. AFIPS Fall Joint Computer Conference Vol. 33, 267-277.

[59] Tigran V. Khotsanyan, Vladimir G. Sahakyan, Irina A. Safaryan (2010). *Statistical Postprocessing of the Output from Numerical Weather Prediction System.* Mathematical Problems of Computer Science 33, 59–65.

[60] http://www.realclimate.org/epica.jpg.

[61] Aach J., Church G.M. (2001). *Aligning gene expression time series with time warping algorithms.* Bioinformatics 17, 495-508.

List of Figures

List of Tables

List of Abbreviations

ACID Atomicity, Consistency, Isolation, Durability

ANOVA Analysis of Variance

BioPAX Biological Pathway Exchange

BMBF Bundesministerium für Bildung und Forschung

CMS Content Management System

CPDB Consensus Path Database

CSS Cascading Style Sheets

DNA Deoxyribonucleic acid

DOM Document Object Model

EDVis Expression data visualizer

EGAN Exploratory Gene Association Networks

GenMapp Gene Microarray Pathway Profiler

GML Graph Modelling Language

GO Gene Ontology

iHOP information Hyperlinked Over Proteins

KEGG Kyoto Encyclopedia of Genes and Genomes

LGPL Library GNU Public License

MedSys Medizinische Systembiologie

MVC Model-View-Controller design pattern

PICR Protein Identifier Cross-Reference

RNA Ribonucleic acid

SBML Systems Biology Markup Language

SIF Simple Interaction Format

SVG Scalable Vector Graphics

VisAnt Visual Analysis Tool

XGMML eXtensible Graph Markup and Modeling Language

XML Extensible Markup Language

ZPT Zope Page Templates

Supplementary Data

Usability Test EDVis

This form is used to determine weak points, bugs, inconveniences in the use of EDVis (Expression Data Visualiser) experienced by the user in order to register quality defects and to examine the usability of the web application.

Please accomplish following tasks and answer the questions:

Tasks

1. **Plot Creation**

- Create a plot containing components of your choice.
- Change its settings.
- Remove/add components.
- Search for Similar Profiles.
- Save plot.

2. **Network Graph Creation**

- Create a graph.
- Change it.
- Choose data resources and experiments and create corresponding plots.
- Search for Similar Profiles and add a profile of your choice to the graph.
- Search for Interacions in CPDB and add interactions of your choice to the graph.

3.Data Manipulation

- Download data from one/many experiments
- Upload user data

Questions

1. **Was the workflow smooth so that you always knew how to continue with the task?**

   ☐ yes ☐ no ☐ not sure

   If no, please list the places where you didn't know how to go on.

2. **In case you needed help, was the Help menu useful?**

   ☐ yes ☐ no ☐ not sure

   Please note where you had difficulties in understanding the Help menu.

3. **Does the web application reflect your needs?**

   ☐ yes ☐ no ☐ not sure

   Please write down possible functionalities that you're missing.

4. **Did you have to go through steps that were in your oppinion not necessary or too complicated to understand?**

   ☐ yes ☐ no ☐ not sure

   If some of the steps was too complicated, please describe to what extent. How should it look like in your opinion?

5. **Is all information you need to accomplish your tasks clearly displayed and understandable?**

   ☐ yes ☐ no ☐ not sure

   Note missing information and the place where it is needed.

6. **Did you have to ask somebody else for help?**

   ☐ yes ☐ no ☐ not sure

   If yes, describe the problem.

7. **Did you get enough feedback from the system where you needed one or thought you should get one?**

   ☐ yes ☐ no ☐ not sure

Note where you missed some more feedback.

8. **Was the workflow of the tasks reasonable?**

   ☐ yes ☐ no ☐ not sure

   Where would you change the workflow or change the sequence of the steps?

9. **Does the programm do something that you weren't expecting?**

   ☐ yes ☐ no ☐ not sure

   If yes, what?

10. **Were there unacceptable long response times that led to confusion or irritation?**

    ☐ yes ☐ no ☐ not sure

    If yes, where.

11. **Does the design of the menu and the different components (graph, plots, etc) satisfy your expectations?**

    ☐ yes ☐ no ☐ not sure

    Note your design suggestions if not satisfied.

12. Were you sure that the program is working (calculating something) though there was a long waiting pause?

☐ yes ☐ no ☐ not sure

Please not the situations where you werent't quite sure.

13. Did you get a feedback if you entered a wrong input or missed to deliver the needed input for the program?

☐ yes ☐ no ☐ not sure

List the places in the program where you missed such feedback.

14. Did you experience some unexpected erros/bugs?

☐ yes ☐ no ☐ not sure

Please describe the bugs and the step sequence that led to the error.

Thank you for your cooperation!