UiT

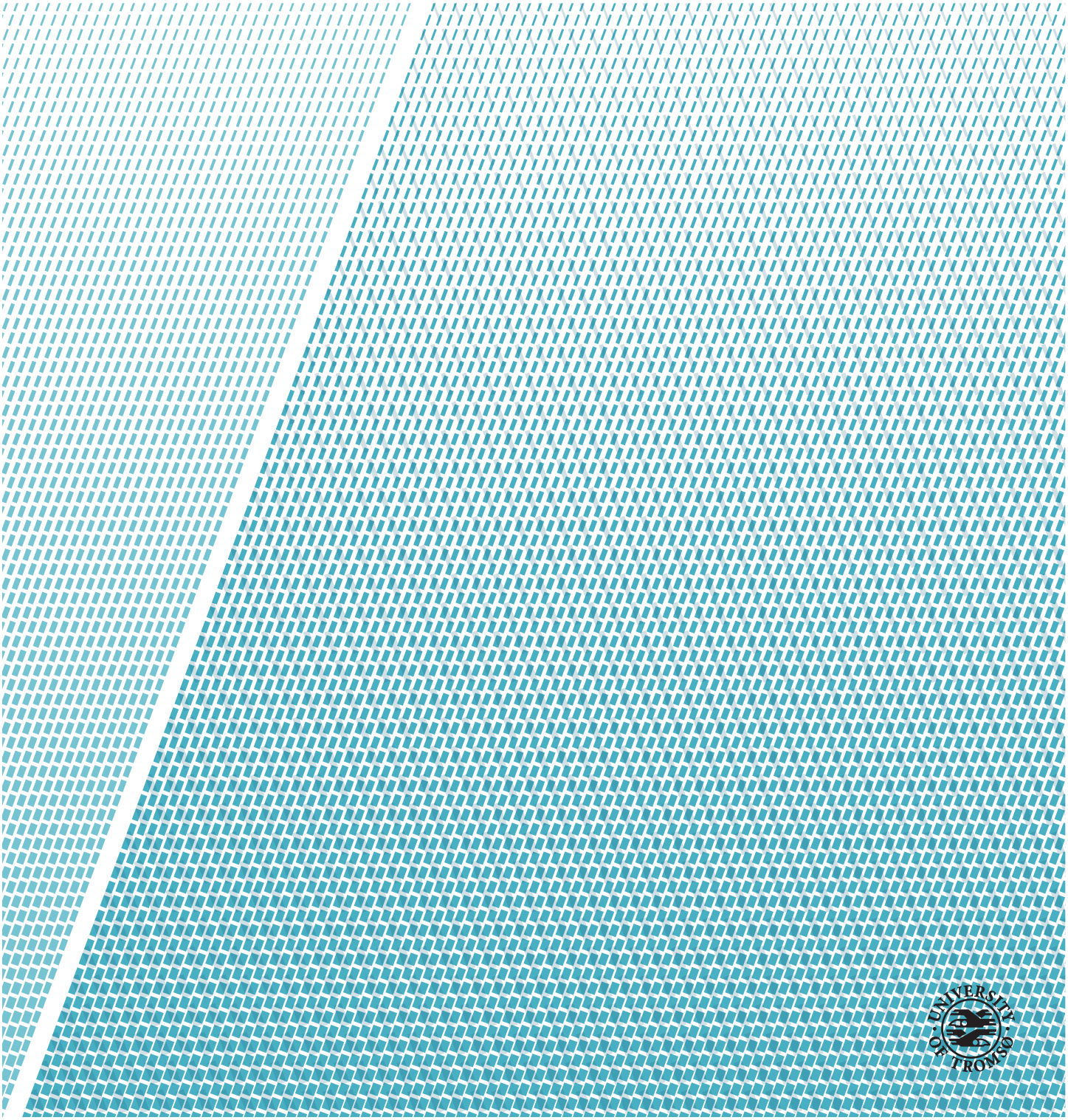
THE ARCTIC UNIVERSITY OF NORWAY

Faculty of Science and Technology

Kvik: Interactive exploration of genomic data from the NOWAC postgenome biobank

Bjørn Fjukstad

INF-3990 Master's thesis in computer science, May 2014



Abstract

Recent technological advances provide large amounts of data for epidemiological analyses that can provide novel insights in the dynamics of carcinogenesis. These analyses are often performed without prior hypothesis and therefore require an exploratory approach. Realizing exploratory analysis requires the development of new systems that provide interactive exploration and visualization of large-scale scientific datasets.

This thesis presents Kvik, an interactive system for exploring the dynamics of carcinogenesis through integrated studies of biological pathways and genomic data. Kvik is designed as a three-tiered application, an architecture that is commonly used for peta-scale applications. It provides researchers with a lightweight web application for navigating through biological pathways from the KEGG database integrated with genomic data from the NOWAC postgenome biobank.

In collaboration with researchers from the NOWAC systems epidemiology group, we have described the requirements for such a system, and by using an iterative approach we implemented Kvik through small development cycles, involving the end-users in the development process. Throughout the project we have gained valuable interdisciplinary experience in developing systems for use in explorative analysis of carcinogenesis.

Through an evaluation of the exploration tasks and workflow of an end-user, we demonstrate that Kvik has the capability of interactive exploration of genomic data and biological pathways.

We believe Kvik is important to enable novel discoveries from the data produced in the NOWAC systems epidemiology project. It provides epidemiology researchers with access to powerful compute and storage resources enabling the use of advanced statistical methods for the analysis. Finally, from our experiences in developing Kvik, we provide use cases and requirements for future analysis, computation and storage systems developed in our research group and by others.

Acknowledgements

First I would like to thank my advisor, Associate Professor Lars Ailo Bongo for his motivation and continuous guidance during the course of this project. I would also like to thank my co-advisor Professor Eiliv Lund for his encouragement and inspiring insights.

I would like to thank Mie Jareid and Karina Standahl Olsen for sharing their biological knowledge and providing invaluable input throughout the project.

Knut Hansen and Nicolle Mode for their help with the NOWAC dataset.

To the HPDS research group for an exciting working environment.

To my fellow students: Einar Holsbø, Jan-Ove 'Kuken' Karlberg, Magnus Stenhaug, Vegard Sandengen, Kristian Elsebø, Michael Kampffmeyer, Erlend Graff, Tom Pedersen, Ida Jaklin Johansen and possibly Martin Ernstsen. Thank you for all the great years at the university!

I would like to thank my parents for their encouragement and warm dinners, and my brother for really going for it!

Finally, Ane Sætrum for her loving support and waiting up for me to get home at night.

Bjørn
Tromsø, May 2014

Contents

List of Figures

List of Tables

List of Abbreviations

ANOVA Analysis of Variance

API Application programming interface

CPU Central Processing Unit

CSV Comma Separated Values

DNA Deoxyribonucleic acid

FTP File Transfer Protocol

GPL GNU Public License

GSEA Gene Set Enrichment Analysis

GSVA Gene Set Variation Analysis

GUI graphical user interface

HCI Human-Computer Interaction

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

JOGL Java OpenGL

JRE Java Runtime Environment

JSON JavaScript Object Notation

KEGG Kyoto Encyclopedia of Genes and Genomes

KGML KEGG Markup Language

NGS Next-generation sequencing

NOWAC Norwegian Women and Cancer

PNG Portable Network Graphics

REST Representional state transfer

RNA Ribonucleic acid

RPC Remote Procedure Call

SD Standard deviation

TSV Tab Separated Values

XML eXtensible Markup Language

1 Introduction

Cancer is the leading cause of death in economically developed countries and the second leading cause of death in developing countries. The number of cancer victims is continuously growing as the world's population is both aging and increasing in size. Contributing to the increase is cancer-causing behavior like smoking, poor diet or little excercise.

The authors of [1] reported that in 2008 there were about 12.7 million cancer cases diagnosed and 7.6 million deaths from cancer around the world. By 2030 the authors estimated that there will be 21.3 million new cancer cases and 13.1 million deaths just because of the growth and aging of the population. Reducing the prevalence of cancer rely new tehcnologies providing new insights and understanding that lead to better treatment and more accurate diagnosis. In 2013 Science awarded the Breaktrough of the Year to Cancer Immunotherapy[2], a treatment method that unleashes the immune system against cancerous tumors. Biological discoveries such as this bring hope and proves that it is possible to transform biological insights into life-saving drugs.

From the discovery of the Deoxyribonucleic acid (DNA) structure by Watson and Crick in 1953[3] to the sequencing of the human genome in 2001 [4, 5] and the massively parallel sequencing platforms in the later years[6], the scientific advances have been tremendous. Today, single week-long sequencing runs can produce as much data as did entire genome centers just years ago [7]. These technologies allow researchers to collect data faster and more efficient,

now making it possible to collect the entire genome from a patient in less than a days work. As of 2010, the number of base pairs sequenced doubled in less than every 6 months [8], leaving both storage capacity and computation speed far behind. And while sequencing is becoming cheaper, the downstream analysis and interpretation of the results is still a major challenge[9].

With these massive quantities of research data, scientists from different fields must collaborate to transform data into knowledge and develop new methods for diagnosis and treatment of cancer. Figure 1.1 illustrates the process of gaining knowledge from data. With increased quantities of data the first step of transforming data into information has become more time consuming and challenging. Even more important is to develop systems for researchers to extract knowledge from the information. To build such systems, computer scientsist require both domain knowledge to understand the problems at hand, as well as insight in managing large quantities of research data and presenting it to end-users. A common approach to present large scale datasets is to generate visualizations and statistics that allow the users to view the data differently.

Figure 1.1: The process of gaining knowledge from data

This thesis presents Kvik, a system for interactive exploration of multi-omics data from the Norwegian Women and Cancer (NOWAC) postgenome biobank. The system was developed in close collaboration with cancer researchers from the NOWAC Epidemiology research group at The Department of Community Medicine.The thesis has a focus on visualizing biological pathways and how cancer researchers can use Kvik to explore the NOWAC dataset.

The name Kvik comes from Norwegian polar history. Kvik was one of Fridtjof Nansen's sledgedogs on the Fram Expedition to The North Pole[10, 11]. The word means means both *brisk* and *quick-witted*, and describes the characteristics of the system.

1.1 Challenges

There are multiple general challenges when designing biological data exploration tools[12]. Solving these challenges needs a wide range of solutions, from new hardware platforms to novel visualization techniques. For a thor-

ough review on visualizing biological data, see the author's report *NOWAC Data Exploration* [13].

In collaboration with domain experts in epidemiological research we have identified three main challenges: i) providing scalable and sustainable storage and computation models for future datasets; ii) providing effective visualizations; and iii) developing intuitive user interfaces.

1.1.1 Storage and Computation

The computational challenges in biological data exploration cover the problems of storing, performing computations on biological datasets.

Traditionally, researchers have used desktop computers to conduct exploratory analyses, but with the growing datasets the computational power of the computers has become a bottleneck. Peta-scale Next-generation sequencing (NGS) datasets require distributed systems to provide the storage capacity and computational power to provide an interactive exploration system.

Some projects do not yet want to publish datasets outside their research group. To prevent unauthorized access, researchers store all data in a secure in-house storage facility, making it necessary for visualization tools to access the data remotely.

1.1.2 Effective Visualizations

Providing helpful visualizations of biological data is key to transforming large quantities of information into knowledge. Not overwhelming researchers with information in the visualizations is a challenge that is still largely unfulfilled and will require the development of truly integrated and highly usable tools [12]. Another aspect of providing effective visualizations is to visualize data at correct zoom level. To understand diseases researchers collect data about different levels in our body, from the atoms of the cells and up to the function of organs. Showing the structure of atoms when looking at entire organs could overwhelm researchers with irrelevant information.

In biology researchers require visualizations that integrate results from literature in textual form, as well as numerical data such as gene expression profiles. A challenge is using many visualization techniques for different data types.

Data cleaning is another challenge of data exploration. Important steps in

the analysis of microarray datasets are the pre-processing stages, where researchers are able to pre-process and normalize data. These stages may alter the final dataset that goes into the visualization software, affecting any downstream discoveries. When researchers perform the data exploration, the optimal parameters going into these pre-processing stages are often unknown until the final visualization steps. Modifying such parameters interactively from a visualization tool may provide helpful insights[14].

There are currently a limited number of tools that allow users to integrate own statistical methods into the data exploration task. Researchers would need to use statistics packages like R[1] to produce datasets that can be visualized later in a separate piece of software.

The display resolution of a researcher's monitor is a limiting component on how much information it is possible to visualize at a time. Especially when dealing with large scale biological datasets, this is a challenge since exploration tools cannot visualize the entire datasets without compacting them. The use of high-resolution displays have opened up for new possibilities for researchers.

1.1.3 Simple User Interfaces

Exploration tools must be intuitive and easy to interact with, making it easy for researchers to use them. Developers must understand how to design a usable graphical user interface (GUI) to make the exploration. If an application is difficult to use, the chances that a researcher will continue using it are small. Additionally, the aesthetics of a system influences the perceived ease of use of the system[15], stressing that a simple and "pretty" user interface is of importance.

Identifying tasks that data exploration systems can automate is a challenge that can potentially speed up the exploration process. Ideally the tools should provide visualizations for tasks requiring human inspection, while automating others. If users tend to use the same outlier removal techniques, the data exploration system could automate this step allowing users to start exploring data faster.

1. r-project.org

1.2 Norwegian Women and Cancer

The NOWAC sustems epidemiology research project is a study designed to identify the possible relationships between lifestyle and the risk of cancer. It started its data collection in 1991. In 2006 the study contained questionnaire information from over 170 000 women. Since then the data collection started in 1998, the NOWAC postgenome biobank has grown to over 60 000 blood samples and 800 biopsies that have been, or will be, analyzed using whole-genome gene expression analysis tools. Additionally the biobank contains information about exposure through questionnaires answered by the participants of the study. More information can be found at site.uit.no/nowac and in [16].

The main objective in systems epidemiology is to test the assumption or hypothesis that human carcinogenesis is communicated through blood. Through the *Transcriptomics in cancer epidemiology*[17] research project the experience was that it was necessary to run all genome analyses agnostic or without any a apriori hypothesis. These results motivate the need for specialized analysis, exploration and visualization for systems epidemiology data.

To enable researchers to gain knowledge from the large quantities of information in the NOWAC postgenome biobank, the researchers require new solutions for exploratory data analyses. In particular, visualization tools that integrate data from multiple biological levels (e.g. from genes to a population), linking to cancer databases and merging it together into a single interactive system.

Currently, systems that produce such visualizations are often stand-alone desktop applications managing both visualizations and data locally. Common in industry is moving towards storing data in “cloud” services and providing interfaces to this data in lightweight tools accessible thorugh lightweight web browsers. Since researchers don’t need to install third-party libraries or applications, they can explore data on any device with a modern web browser. Using web applications has other advantages as well. The web application is updated on the server, making it transparent to users. Also, web applications allow users to use a wide variety of operating systems and hardware platform to run the application.

Kvik is an interactive system that combines data intensive computations on multi-level omics datasets and the transformation of information into knowledge and biological insights. It merges different systems and data sources together to provide a single interface for both biologists and statisticians exploring multi-level NOWAC biobank.

Within biology, there are multiple solution for visualizing and presenting research data. One such method is pathway maps, which are graphical representations of biological processes. With pathway maps it is possible to integrate experimental data from e.g. blood samples and the structural organization of biological processes in a single view. Kvik uses this approach.

1.3 Kvik

Through collaboration with Epidemiology researchers at the Institute of Community Medicine at the University of Tromsø, we performed a requirements analysis and identified the following:

Interactivity Kvik should provide helpful visualizations in an interactive fashion. Users should not have to wait an unnecessary amount of time before they receive some feedback or result. Delays less than 0.1 seconds are unnoticeable, but anything more than a second will act intrusive on the users line of thought[18]. With visualizations that take more than one second to load the system should present the user with a progress bar, or an indication that the system has not crashed.

Scalability Kvik should scale to the upcoming petascale datasets. In addition to data handling, the visualization tools should also be capable of visualizing large quantities of data.

Familiarity The visualizations provided by Kvik should follow familiar visualization techniques. E.g. researchers in the NOWAC research group are familiar with the manually drawn pathway maps used in KEGG making it obligatory for Kvik to follow this drawing convention. Other systems such as enRoute[19] have also identified this requirement.

Heterogeneity Researchers are collecting data from a large number of sources, both online databases and local datasets. Kvik should handle the addition of such data sources without any major overhead. It should also be able to process different data sources, from textual sources like KEGG or numeric datasets like gene expression data from the NOWAC biobank.

Expandability Researchers are continuously discovering novel methods to visualize or process data, making it essential for Kvik to be expandable both to processing and representation of data. Kvik should also facilitate data processing in systems such as Hadoop MapReduce[2] or Apache

2. hadoop.apache.org

Spark[3] without any major developer overhead.

Simplicity Kvik should provide simplicity both in terms of the system but also with regards ease of use for the users. Researchers should not have to install a single piece of software in order to run Kvik. This requirement dictates that the implementation uses software already installed on the researchers computers or devices.

Security Since Kvik is a system that will manage datasets containing sensitive data, secure storage is of importance. Kvik must provide an interface to access data from a secure storage facility, possibly behind restrictive firewalls.

To our knowledge, no existing system fulfill all these requirements. There are multiple online resources for visualizing biological pathways, like KEGG [20] or BioCarta [21], but they provide poor interaction support. These tools require users to switch between separate views when selecting genes or compounds in a pathway, making it difficult to keep the same line of thought when exploring the biological pathways. VisANT [22], VANTED [23] and KEGGViewer [24] are both systems for interactively exploring biological pathways, but both of these lack visual cues, like cell walls, that would make the visualizations familiar to the researchers. enRoute [19] and Entourage [25] both included the Caleydo framework [26] provide familiar visualizations and incorporates the possibility to visualize gene expression from multiple heterogeneous data sources. Nevertheless, the Caleydo framework is a standalone application that requires installation on researchers computers, failing the simplicity requirement. The Caleydo framework is not the only system failing the simplicity requirement. VisANT and Vanted are both dependent on users installing the Java Runtime Environment (JRE) in addition to the application, or a Java plug-in to run in the web browser. Pathway Projector [27] is a system that visualizes biological pathways without any installation. It allows users to browse biological pathways similar to viewing maps on Google Maps[4], but fails the security requirement since researchers must upload to their servers for visualization. Chapter 6 describes these systems in more detail.

Kvik fulfill the above requirements as follows:

Interactivity Kvik achieve interactivity by visualizing biological pathways and corresponding expression data in a single view using modern HyperText Markup Language (HTML)5 technology.

3. spark.apache.org
4. maps.google.com

Scalability Kvik separates computational resources and display resources, making it possible to explore large quantities of data even on lightweight clients. Using this separation Kvik is able to make use of large storage clusters and move computation to designated compute nodes.

Familiarity Kvik visualizes biological pathways using the traditional KEGG layout, achieving familiarity with contextual visual cues familiar to researchers.

Hetereogeneity Kvik incorporates multiple heterogeneous data sources into a single data engine.

Expandability With a modular design, Kvik is capable of adapting to both software and hardware improvements, from data sources to screen resolution.

Simplicity The exploration tool in Kvik runs in a modern web browser and does not require any third party plug-ins or applications. Kvik is modular by design, separating components into independent functional units.

Security Kvik uses a designated data engine for storing expression data from the NOWAC study. Kvik will only expose visualizations to the researchers, making it impossible to view or download any of the raw data.

This thesis presents Kvik, an interactive system for exploring the dynamics of carcinogenesis through studies of biological pathways and genomic data. It describes the future direction of the project and how it is designed to handle both new data sources, analysis methods and visualization techniques.

We have evaluated Kvik by measuring the latencies to load the different visualizations. The results show that Kvik is usable for interactive exploration of

In collaboration with researchers from the nowac research group, we demonstrate how Kvik bridges the gap between biologists and statistics in a single exploration tool. From our experiences we demonstrate the importance of an iterative development process, and how Kvik has benefit from this model.

1.4 Contributions

The contributions of this work are:

- A requirement analysis for visualization system for exploring and visualizing data from the NOWAC postgenome biobank.

- The design and implementation of Kvik, a data exploration tool for biological pathways and gene expression data. It provides interactive exploration of biological pathways from the KEGG database integrated with genomic data from the NOWAC postgenome biobank.

- The experimental evaluation of Kvik, demonstrating that researchers can use Kvik for interactive exploration of the full NOWAC biobank and KEGG databases.

1.5 Organization

The thesis is structured as follows. Chapter 2 gives an introduction for computer scientists to get up to speed with the biology that goes into designing and implementing an exploration tool for multi-omics data. Kvik, is outlined in Chapter 3, which also describes the desired workflow of our collaborators. Chapter 4 describes the architecture of Kvik and its three components. The design and implementation follows in Chapter 5. A study of the state of the art tools for visualizing biological pathways is given in Chapter 6. Chapter 7 covers the evaluation of Kvik, both with regards to the performance but also usability for the researchers. Concluding remarks are given in Chapter 8 and future work in Chapter 9.

2

Biological Background

The main goal of the NOWAC study is to understand the impact of exposores on the risk of getting cancer. Making novel scientific discoveries that may lead to understanding carcinogenesis requires the collaboration of multiple sciences. With the collection of multi-level datasets, unique challenges for computer scientists emerge, both regarding storage as well as analysis. Understanding fundamental concepts in biology are crucial for developing effective and intuitive data exploration tools for large scale biological datasets.

This chapter gives a brief introduction in molecular biology neccessary to understand the challenges in exploring the NOWAC biobank. It describes cells, DNA, Ribonucleic acid (RNA), proteins, and biological pathways. Finally the chapter concludes in a description of the challenges faced by our partners from the Institute of Community Medicine at University of Tromsø when analyzing data from the NOWAC study, and how Kvik addresses these. For a more thorough description see the authors's special curriculum report [13].

2.1 Molecular Biology

Carcinogenesis is the development of cancer in an organism. To understand this process and how lifestyle and hormones may impact the risk of having cancer, we need to understand the different building blocks that make an organism.

Cells are the smallest units of a living organism that still preform a function. Cells perform multiple tasks: exchanging materials with their environment; self duplication; transmitting and receiving signals with their environment; and the synthesizing of molecules. The human body specializes every cell and organizes them into tissues. These tissues for form organs which in turn form organ systems. All cells contain the same genetic information, but not everything is actually used. Nucleic acids are responsible for storing, transmitting and expressing the genetic material. There are two types of nucleic acids, Deoxyribonucleic acid (DNA) and Ribonucleic acid (RNA). DNA stores the genetic information, while RNA decodes the information stored within the DNA. Small molecules known as nucleotides compose the strands of DNA and RNA, forming a ribbon like structure. There are different classes of nucleotides that compose the different nucleic acids, identified by four letters. A, G, C and T describes the DNA sequence, while in RNA the letter U replaces the T resulting in sequences of the letters A, G, C and U. Cells organize DNA into two complimentary strands in a double helix structure held together with chemical bonds. These two strands contain sequences of the four letters A, G, T and G, pairing A and T, and G and C between the two strands. Figure 2.1 illustrates a hypothetical strand of DNA. RNA consists only of a single strand, as illustrated on figure 2.2.

```
…gtgcatctgactcctgaggagaag…
…cacgtagactgaggactcctcttc…
```

Figure 2.1: The two strands of DNA. Chemical bonds align the letters G and C, and A and T.

```
…gugcaucugacuccugaggagaag…
```

Figure 2.2: A single strand of RNA

The genetic information stored within cells determine how the body performs different biological processes. The sequences or coding units of DNA, called genes, determine how an organism should synthesize large molecules called proteins. This process from DNA to proteins is called the *Central Dogma of Molecular Biology*, first stated by Francis Crick in 1958[28]. The proteins aid in a number of processes, for example by turning food into energy, the process of cell development, and contribute to the distribution of oxygen in the body. Humans have approximately 20.500 genes that make up the human genome.

The protein synthesis process starts within a specific part of the cell. The first step, transcription, is the flow of genetic information from DNA to RNA. The strands of RNA are then translated into sequences of amino acids, molecules

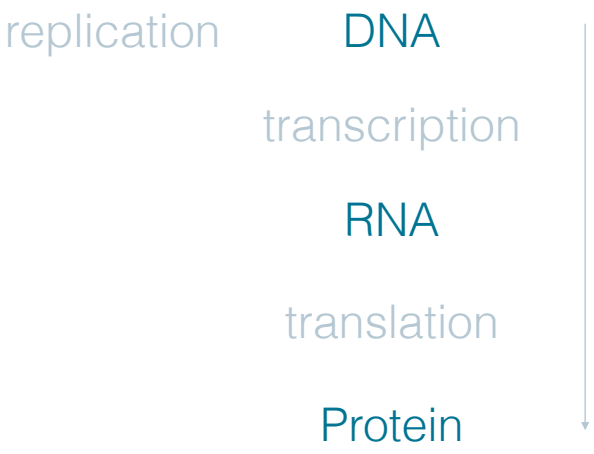


Figure 2.3: The central dogma of molecular biology

consisting of carbon, hydrogen, oxygen and nitrogen, that describes the protein. The protein is then folded into a three dimensional structure and transfered to its destination. Figure 2.3 depicts a high level view of this process. Figure 2.4 illustrates a more detailed example of protein synthesis. It illustrates a sequence of DNA with a gene highlighted on the top row. The second row contains three letter sequences known as codons, with the corresponding transcribed codon highlighted. Each codon encodes for a specific amino acid, except the stop codons, which terminate protein synthesis. The sequence of DNA in this example transcribes the codon `cau`, which in turn translates to the amino acid represented by the letter `h`.

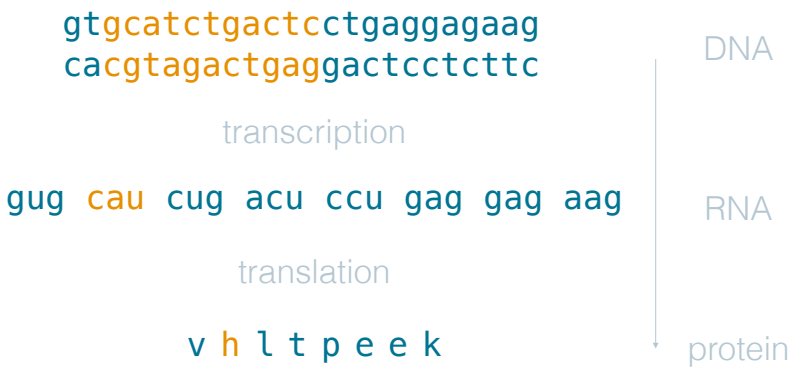


Figure 2.4: The central dogma of molecular biology

Another important part of the central dogma is DNA replication. This is a continuous process within the body that starts with the DNA within the egg cell and continues to produce the ~40 trillion cells [29] that make up the human body. The process takes a single DNA molecule, divides the two strands, and pairs the bases in the individual strands with a complementary bases. In

other words, a special enzyme reads the letters in the strands sequentially inserting e.g., T when it encounters any A's, G when it encounters C's and so on. After the creation of the new DNA molecule it goes through an error correcting process, which should detect and fix any error. Mutations are alterations in the nucleotide sequence in the DNA, either switched bases or entire parts of the DNA strand deleted or added. The error correction process is able to fix most errors, but with the large number of cells in the body some errors will occur. Since 98% of the human genome does not contain any protein coding information [30], mutations may not cause any notable effect to the cells. Nevertheless, mutations can effect either the organism itself or its offspring.

2.2 Carcinogenesis

As mentioned, carcinogenesis is the development of cancer in an organism. Cancer are results of gene mutations that can occur in any cell anytime. These mutations may cause the cells to grow uncontrollably, resulting in tumors. A tumor is a growing mass of tissue, which could be either benign or malignant. Benign tumors have not invaded surrounding tissues, while malignant tumors have spread to their surrounding tissues interrupting their functions. Cancer is primarily a disease of old age, due to accumulation of mutations over years. Some inherited mutations my increase the risk of cancer considerably, for example the breast cancer genes BRCA1 or BRCA2 that has got much public attention lately.

To understand carcinogenesis, researchers must often study changes in multiple genes that possibly participate in different biological processes. Studying the changes in a number of genes and how this affects the processes they participate in is a challenging task. Not just because of the number of genes and processes, but also the fact that researchers may look for small changes in multiple genes, not necessary big changes in single genes. Another challenge is the time aspect of carcinogenesis, how are the genes expressed prior to diagnosis of cancer? Understanding this process requires the monitoring of genes over time series spanning decades.

The NOWAC biobank was built to understand carcinogenesis through analyses of blood samples. This main objective or hypothesis has been and is still controversial, but the project is considered as "high risk – high gain".

2.3 Genomic Data

To study carcinogenesis researchers must collect genetic data from patients. With Next-generation sequencing (NGS) it is now possible to get the genetic information stored in DNA at low cost in just a day's work.

DNA sequencing is the process of determining the order of nucleotides within a strand of DNA. Researchers must sequence patients DNA to be able to detect mutations that may lead to diseases. Previously sequencing has been both a time consuming and costly process, but with NGS it is now possible of doing what previously took weeks for only a fraction of the cost and time. Figure 2.5 shows how the cost of sequencing a megabase (a million letters from a DNA strand) has evolved over the past 13 years. In addition to the cost of sequencing, the graph illustrates hypothetical data following Moore's Law [31], highlighting the extreme decline in cost starting early in 2008. This decline marks the wide adaptation of the Next-generation sequencing (NGS) techniques, that makes it possible to sequence longer strands of DNA.

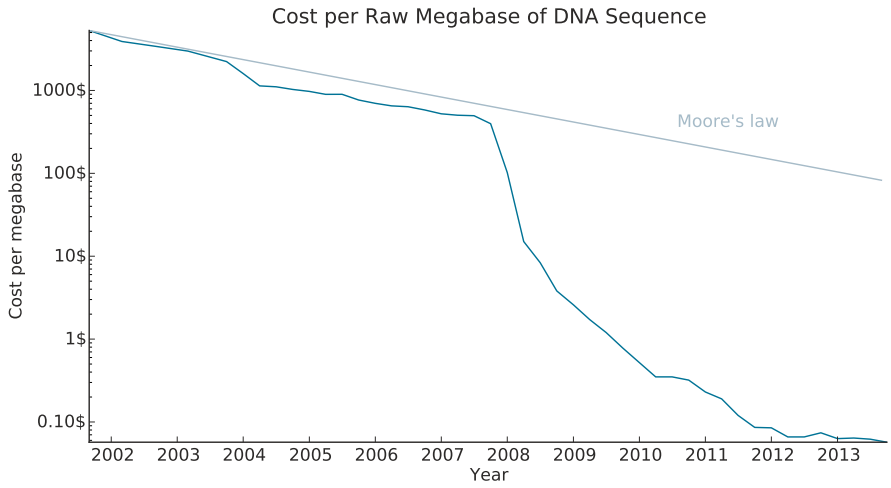


Figure 2.5: Cost per RAW Megabase of DNA Sequence. Figure adapted from http://www.genome.gov/sequencingcosts/

Microarray is a high-throughput technology used in the NOWAC study that can investigate different levels of biological data, from proteins to DNA and RNA. Currently the NOWAC biobank consists of more than 70 000 blood samples now being analyzed using this technology. In the future the researchers plan to use other sequencing techniques such as Deep-sequencing.

DNA microarrays are matrices on some solid surface (often glass) with single stranded DNA probes that correspond to genes. To analyse gene expression

RNA gets extracted from biological samples, pre-processed and placed on the substrate. This material hybridizes, or connects, to probes on the substrates using complimentary base pairing. Complementary base pairing is the process of connecting the letters A and T, and G and C from the probes and the RNA material. After washing away any unbound material, the RNA abundance is quantitated by image analysis [32]. See figure 2.6 for an illustration.

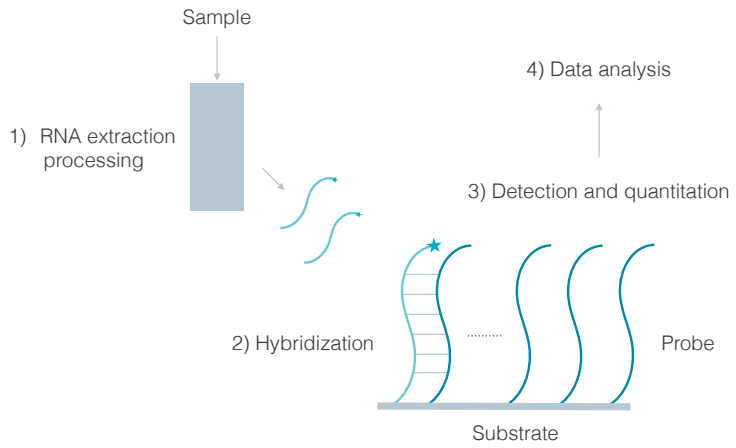


Figure 2.6: Microarray technology. Figure based on figure 8 from [32]

2.4 Biological Pathways

Living organisms are often described as complex biological networks of molecules and interactions between them. These molecules may be genes, proteins or other compounds. A biological pathway is the series of actions along a biological network that leads to a specific effect, such as assembly of new molecules, turning genes on or off, or spur a cell to move [33]. There are different types of pathways, but the most common groups of pathways are metabolic pathways, gene regulatory pathways and signal transduction pathways.

To understand the complex relationships in different organisms researchers use abstract pathway maps. These maps hold information on how genes interact with each other, the assembly of new molecules and signal transmission within an organism. Pathway maps hold large amounts of information and is an invaluable source of information for researchers.

Metabolic pathways are the chemical reactions within an organism. Examples of metabolic pathways are the processes breaking food down to energy or the uptake of oxygen in the blood.

Gene regulatory pathways are responsible for turning genes on and off. Turning genes off will stop an organism from producing the protein it codes for, and turning it on may increase the ammount of proteins produced.

Signal transduction pathways are responsible from moving signals from outside a cell to its interior. The messages sent to the cells may instruct it to move or to perform some action.

With the graphical representation of biological pathways, researchers are able to gain deep insights into complex biological phenomena. Often, researchers compare pathways in healthy persons and persons with a specific illness to reveal similarities or differences. Identifying proteins or genes acting differently in a pathway may reveal the roots of a specific disease. In cancer there are usually a number of pathways and genes affected, requiring the researchers to investigate these in combination.

3

Kvik

Kvik is a system helps researchers gain new knowledge and biological insights from the multi-omics biobank of the NOWAC study. Kvik allows researchers to explore DNA expression profiles integrated in pathway maps. It integrates gene expression data from NOWAC with pathway maps from the popular pathway database KEGG, providing state of the art pathway visualizations to the researchers.

Kvik is a web application that researchers can run in web browsers on their workstations, or even mobile devices. It uses open source libraries and does not need any installation of third party applications or plugins. Figure 3.1 shows the GUI of Kvik. It consists of a pathway viewer on the left, and a gene information panel view on the right. Kvik visualizes pathway maps as static KEGG pathway images with gene expression data visualized on top of the hand-drawn images. Gene expression data is retrieved from the NOWAC biobank as the researcher explores pathways and genes.

Users can select genes from the pathway by clicking on the different gene names, opening an adjacent view that visualizes gene expression profiles and relevant information for the selected gene. The KEGG database also contain relevant information about genes, such as its definition, structure and which pathways the gene is found in. Figure 3.1 illustrates how Kvik composes both pathway visualization and a info panel for detailed inspection of genes. The info panel includes a visualization of similar pathways to the pathway in the main view. This simplifies the navigation to relevant pathways. Kvik measures

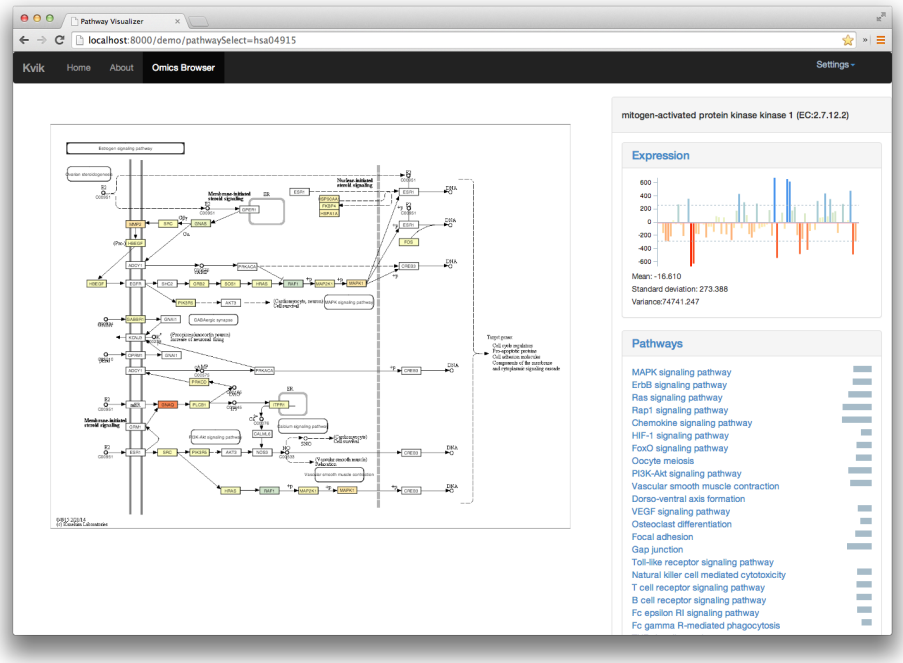


Figure 3.1: Overview of the user interface of Kvik. The user is viewing the Estrogen Signaling Pathway (hsa04914) and inspecting the Mitogen-activated protein kinase kinase 1 gene (hsa:5604)

similarity as the number of common genes between pathways.

Through the visualization of biological pathways researchers can investigate and gain knowledge about different processes within an organism, e.g. the intercellular PI3K pathway that contributes to cell growth [34]. Researchers start data exploration by selecting a pathway of choice to explore. The user is redirected to the main window of the Kvik Browser, where the pathway is visualized. The pathway nodes that correspond to a gene is colored by the difference in gene expression, or fold change, between the cases and the controls in the NOWAC biobank. From this overview the researchers quickly get an overview of the biological process and data from the NOWAC biobank. Now the researchers can explore different hypotheses about data, if a gene is more active in cancer patients how does it affect the other genes in the pathway. Researchers can continue their exploration by clicking on nodes in the visualization. Since Kvik is connected to the KEGG database, the info panel contains the most recent information about the genes. To allow researchers dig further into the NOWAC biobank, the gene view includes a visualization for all gene expression values in the dataset. With this researchers get more in-depth knowledge about signals or trends in the NOWAC biobank. Since can-

cer is a disease that can affect multiple processes and systems in a organism, researchers often want to move between pathways. In Kvik similar pathways is shown in the gene view, indicating pathways that may be of interest to the researcher. With easy access to relevant pathways researchers can quickly get an overview and knowledge from a large number of processes in an organism.

3.1 Development

As mentioned, Kvik was developed in collaboration with researchers from the Department of Community Medicine. The system was developed using an iterative approach with small cycles getting valuable input from the researchers in every cycle. Using this approach we were able to dynamically change the requirements of Kvik throughout the development process. As new features were added to the system, they were presented to the researchers that came with feedback on what worked and what had to be changed.

3.2 Use Cases

Kvik is currently implements visualizing biological pathways and gene expression data. The system is designed to handle additional exploration tasks: i) exploration of the data at gene level; ii) exploration of the data at pathway level; and iii) targeted searches of genes and pathways.

These addition exploration tasks will be implemented by adding statistical analysis tools from the NOWAC systems epidemiology group into the data engine of Kvik. These are features we plan on implementing next.

3.2.1 Exploration of data at gene level

The researchers want to use Kvik to answer two questions: i) which genes are significantly differentially expressed in a pathway; and ii) in which pathways is a specific gene found.

To answer the first question Kvik must provide statistical packages to perform gene-wise linear analysis (or other statistical methods such as Analysis of Variance (ANOVA) or $t-$test) on gene expression data from the NOWAC biobank. It is also important for researchers to modify the parameters going into the statistical analyses ad hoc, e.g. fold change cutoff or p-value cutoff, making

it necessary for Kvik to provide a simple interface to the statistical packages. The statistical analysis performed by Kvik should yield sorted tables of genes and associated results for the analysis.

The second question requires Kvik to query a pathway database. There are multiple online sources available, such as WikiPathways[1], Reactome [2] or KEGG[3]. Kvik should allow researchers to browse biological pathways, visualizing fold change direction (up/down regulation) and p-value (from statistical analyses) in pathway diagrams.

3.2.2 Exploration of data at pathway level

Our collaborators want to answer the following when exploring pathways: i) which pathways are "over-represented" in the list of differentially expressed genes?; and ii) which pathways are significantly regulated according to pathway-level analysis?

Both of these questions rely on advanced statistical methods, such as Gene Set Enrichment Analysis (GSEA), Gene Set Variation Analysis (GSVA), or conditional hypergeometric testing. In addition they rely on taking the output of the gene-wise analysis (using specified cutoff values) as input to these analyses. The analyses must produce lists of pathways that are helpful to the researchers, and allow visualizations of these. Depending on the statistical methods used, the visualizations can use either gene expression values or using another measure to color nodes.

3.2.3 Targeted search

By using targeted searches for either genes or pathways researchers can investigate specific genes or pathways of interest. Researchers can either use gene name or symbol to search for genes. To look up pathways researchers can use pathway name, id, or by searching for keywords such as ”estrogen”. The searches should yield the same table for exploring genes, and the same pathway visualizations.

1. wikipathways.org
2. reactome.org
3. kegg.jp

4

Architecture

Based on the requirements developed in collaboration with researchers at the Department of Community Medicine, Kvik has a three-tiered architecture. To satistfy the scalability requirement, it is apparent that the entire system cannot run on a single computer. Desktop computers does not have the storage capacity nor computational power to perform the necessary statistical analyses on the NOWAC biobank in addition to running an visualization tool. In addion, to meet the security requirement the data should be stored in a secure location, making the user's desktop computers unsuitable.

Kvik is a system for visualizing and browsing of biological pathways and associated genomic data. It has a three tiered architecture consisting of i) the Kvik Browser, an interactive system for visual exploration of biological pathways and genomic data; ii) a Frontend that translates user interactions in the Kvik Browser into queries that retrieves data and visualizations; and iii) a Backend that provides the Kvik Browser with biological data and statistical analyses. It contains both research data from sequencing platforms and also research knowledge from public databases such as KEGG. Figure 4.1 illustrates the architecture of Kvik.

Users run the Kvik Browser on their local system, connecting to a Frontend server to retrieve content to explore. The Frontend translates a request into one or more queries to the Backend. A typical request from the Kvik Browser is to visualize a pathway. The Frontend is responsible for translating this reqeust into different parts, e.g. get a graphical representation of the pathway, retrieve

gene expression data of the genes in the pathway, and merge the two into a visualization to the Kvik Browser. The Backend extracts data from multiple hetereogeneous data sources and provide a simple interface for the Frontend to connect to. This architecture separates compute and display resources, reducing the constraints on compute resources availible at the researchers using the system.

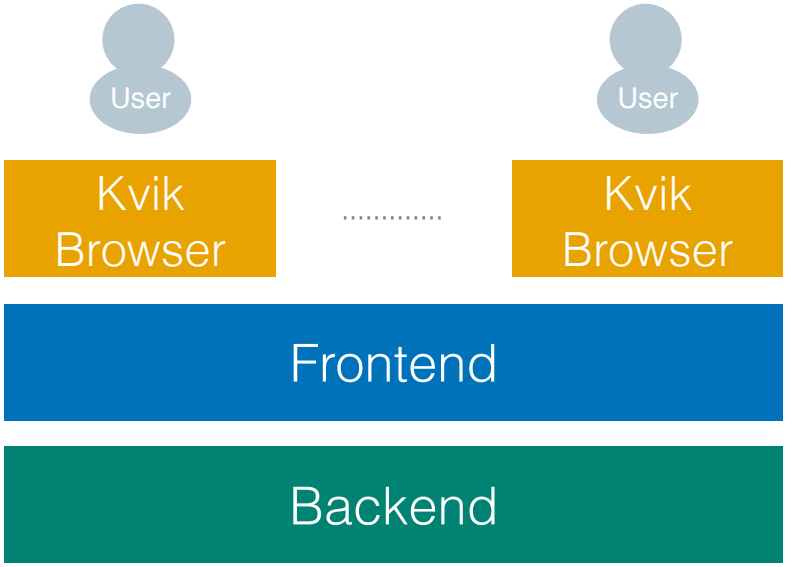


Figure 4.1: Kvik Architecture

4.1 Kvik Browser

The Kvik browser is the point of interaction for users. It should visualize both biological pathways and gene expression, in addition to providing textual information, and a clean and intuitive user interface.

The Kvik Browser consists of two major components, a pathway browser and a gene viewer. The pathway visualization tool provides an interactive pathway browser for researchers to explore biological pathways. With the gene viewer users can dig further into details about specific genes, exploring gene expression data from sequencing instruments and other research data from online databases.

The user interface of Kvik integrates both views and is responsible for translating user input, from a keyboard, mouse or touch interface, to queries to the Frontend. The user interface follows the guidelines in Human-Computer Interaction (HCI) providing feedback to users when the delay time from user

interaction and result exceeds a second [18].

4.2 Frontend

The Frontend is a component that sits between the Kvik Browser and the Backend. It is responsible for translating requests from the Kvik Browser into queries that the Backend can execute. The Frontend exposes an interface which multiple Kvik Browsers can connect to and retrieve data. Requests to the Frontend are translated into one or more requests to the Backend which executes them.

Placing a Frontend between the Kvik Browser and the Backend allows simpler logic in the Kvik Browser. Since the Backend consists of mulitple components, actions from the Kvik Browser may trigger requests for data that is stored in different systems. Additionally since the different data source may require some data parsing, everything is done in a different component than the Kvik Browser.

4.3 Backend

The third component of the Kvik architecture is the Backend. The Backend is the collection of data sources available to the Kvik browser.

In Kvik the Backend consists of two components: i) a database containing biological pathway maps and information about genes and pathways; and ii) a system for managing and performing analysis on gene expression data from the NOWAC biobank.

The gene and pathway database is responsible for providing the Kvik Browser with updated pathway maps and information about genes and pathways. For more effective exploration of biological pathways, our collaborators identified the need for updated research data presented along-side the raw research data and gene expression results. With information about genes and pathways within the Kvik browser, researchers are relieved of the burden of accessing databases through different systems.

The second component is responsible for storing, managing and running statistical analyses on gene expression data from the NOWAC biobank. It is responsible for loading datasets that is available to researchers and responding to queries from the Frontend. Our collaborators design and implement the

statistical analyses, making the primary concern of this component to store and manage the dataset.

5

Design and Implementation

To satisfy the requirements in chapter 1, the design of Kvik follows that of a modern web application. Industry has proven that the separation of data, computation and display resources scales to large datasets and multiple application types. Since our collaborators identified the need for an application that did not require any installation, Kvik is designed and implemented as a web application. Researchers can explore the multi-omics biobank of the NOWAC cohort by simply visiting a URL, without having to install any third-party application or plugin. This design allows updates to Kvik without users having to perform any other work than refreshing a web page. This is especially helpful for in-development apps that are continuously changing. With the design Kvik can also update datasets in the background without any user interaction. This relieves researchers of managing datasets which can concentrate on more important tasks.

The architecture of Kvik consists of three separate parts, the Kvik browser, the Frontend and the Backend. The Kvik Browser is designed and implemented as a web application hosted on a web server and run in the user's web browser. The web server hosts static pages for the web application that is populated with content from the Frontent. Once a user has downloaded a a webpage, the Kvik Browser interacts with the Frontend to retrieve gene expression data and other information. The Backend consists of components for databases and

analysis engines, currently the KEGG databases for information about genes and pathways and the NOWAC Data Engine for exploring the NOWAC biobank. Figure 5.1 illustrates how the different components of Kvik are organized. Typically the Kvik Browser would run on a researchers workstation or laptop, while the other components run in a cluster environment to provide storage and computation capabilities beyond desktop computers.

The separation allows us to easily add new functionality such as new analysis methods or new visualizations.

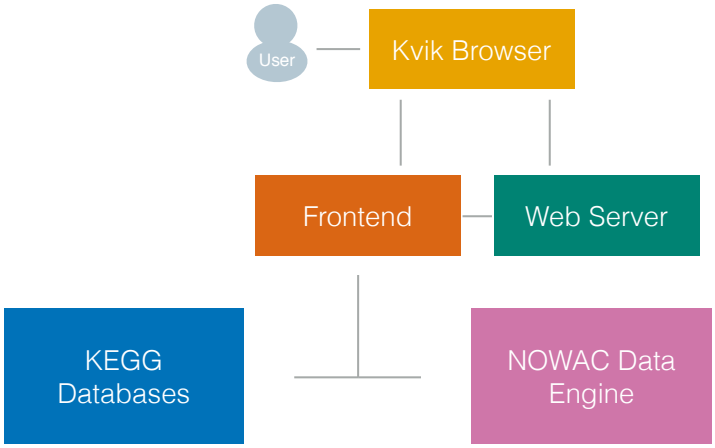


Figure 5.1: Kvik design

5.1 Kvik Browser

The Kvik Browser is the single point of interaction for users of Kvik. It provides a web application for exploring gene expression data and biological pathways. The Kvik Browser is a graphical tool that leverages HTML5 to provide interactive visualizations without users having to install any third party software. With a HTML5 web application there are multiple advantages. First, users receive updates to the application by simply refreshing Kvik Browser. Second, with HTML5 users don't need to install any third party plugin or application, e.g. Java or Flash. HTML5 is a technology that allows the application to run on mobile devices as well, which might be interesting for researchers presenting or discussing results out of the office.

An illustration of the Kvik browser is shown on figure 5.2. It consists of a pathway visualization (left) integrated with a gene viewer (right). When users interact with the pathway on the left, updated information about the genes they inspect is shown on the right.

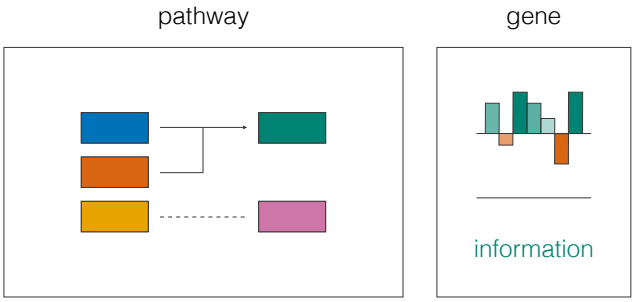


Figure 5.2: An illustration of the Kvik Browser

Users connect to the Kvik Browser by visiting a URL in their web browsers, redirecting them to the web server hosting the web application. The server hosting the web application uses the standard libraries in the Go programming language, serving static pages and generating dynamic content using the template package[1]. The Omics Browser itself uses the Javascript libraries Cytoscape.js[2] and D3[3] for generating visualizations of biological pathways and gene expression data. During the first prototypes of the Kvik Browser the pathway visualization tool was implementing using three.js [4] using WebGL. three.js is a visualization library that is suitable for visualizing large graphs since it utilizes WebGL for rendering. Since KEGG pathway maps are relatively small and the library lacks basic drawing functionality, Cytoscape is easier to use for visualization of graphs such as pathway maps. With the popularity of D3, implementing visualization is easy to get started with because of its extensive list of examples[5]. D3 could have been used for visualizing pathways, but since user interaction must be handled manually when using the HTML5 canvas, Cytopscape is a more suitable alternative since this feature comes out of the box.

5.1.1 Visualizing Biological Pathways

The requirement in Chapter Our collaborators wanted a visualization tool that could integrate pathway maps from KEGG with the NOWAC biobank. From the familiarity requirement 1, Kvik merges the static pathway images from KEGG with Cytoscape graph visualizations, drawing nodes on top of the static images. This approach is used in other system such as Entourage [25].

1. golang.org/pkg/text/template
2. cytoscape.github.io/cytoscape.js
3. d3js.org
4. threejs.org
5. github.com/mbostock/d3/wiki/gallery and bl.ocks.org/mbostock

When users request to view a pathway, the Kvik Browser submits a query to the Frontend. The Frontend generates a visualization of the applicable KEGG pathway and returns it to the Kvik Browser. From the nodes in the pathway the Kvik Browser performs a second request to retrieve gene expression data for the genes in the pathway visualization.

Kvik uses KEGG Markup Language (KGML) representations of biological pathways to generate the pathway visualizations. The KGML is an exchange format used to represent KEGG graph objects, especially the KEGG pathway maps [35]. KGML follows an eXtensible Markup Language (XML) like syntax describing the entities (nodes) and reactions (edges) of the pathway map. Figure 5.3 illustrates the syntax and structure of a KGML file describing the Estrogen signaling pathway (hsa04915). This is the KGML file that is used to construct the pathway figure 3.1.

```
<pathway name="path:hsa04915" … >
  <entry id=1 name="hsa:2009" … >
  .
  .
  .
  <relation entryid1=1 entryid2=3 … >
  .
  .
  .
</pathway>
```

Figure 5.3: Illustration of the KGML syntax

The description includes information about the pathway itself, and a list of entries and relations. The entries (nodes) describes genes, proteins or another compounds, and the relations (edges) describes the reaction between the entries, e.g. activation of a gene. Entries also describe the location of the nodes, allowing Kvik to render nodes in the same location as in the KEGG pathway image.

The KGML file is lacking both edge routing information [36] as well missing nodes. In addition, the manually curated pathway images does not display every edge found in the KGML representation. Figure 5.4 illustrates the differences between hand-drawn pathway maps (top) and automatically generated images from KGML representations (bottom). Hand-drawn pathway maps bundles edges together and adds visual cues to the reader, e.g. the two vertical lines in the center of the figure. Since our collaborators identified the importance of familiar visualizations, Kvik only uses the nodes from the KGML representation, and draws them on-top of the pathway image from KEGG. Figure 5.5 illustrates this approach.

Kvik uses the open-source JavaScript library Cytoscape.js, a graph theory li-

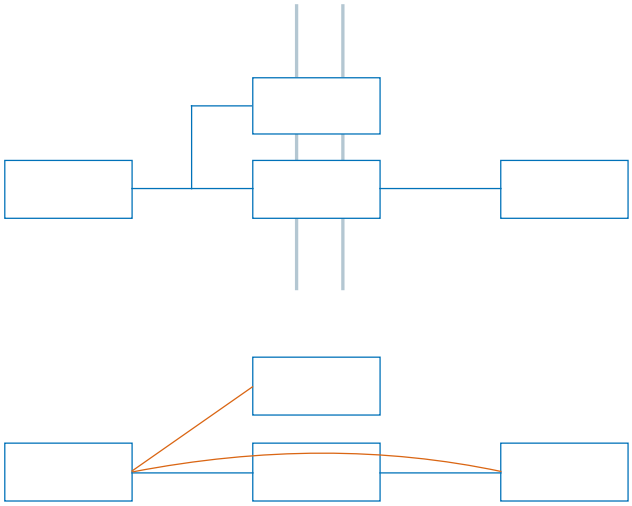
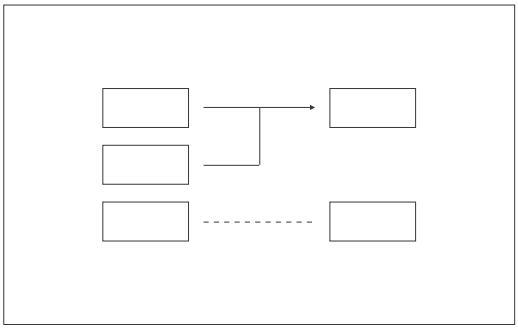


Figure 5.4: Difference between KEGG image and KGML representation

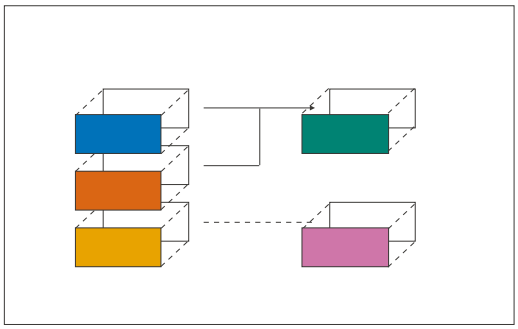
brary for analysis and visualization [37], to generate the interactive pathway graphs. Cytoscape.js shares its name with the popular network visualization software Cytoscape, and is the successor of Cytoscape Web. Cytoscape uses the HTML5 canvas to render graphs, making it suitable for visualizing large networks. Also the Cytoscapee.js library contain helpful graph analysis methods that may be interesting in the future, e.g. highlighting neighboring genes. Kvik generates the pathway visualizations from the entries in the KGML representation as well as a backgound node holding the pathway image from KEGG. It draws the background node onto the HTML5 canvas first, before adding other nodes on top of the background image. When the nodes are added to the image, they are colored accordning to gene expression values from the NOWAC Data Engine. The process of generating the pathway visualization is shown on figure 5.5

The Kvik Browser uses diverging Color Brewer palettes [6] to indicate up or down regulation of a gene. To fetch expression values and other information about the NOWAC dataset, Kvik uses the jQuery [7] JavaScript library that communicates with the Frontend through Hypertext Transfer Protocol (HTTP) GET requests.

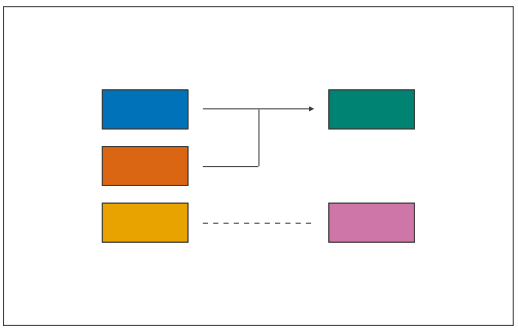
6. colorbrewer.org
7. http://jquery.com/



(a) Original static pathway image from KEGG, placed as a background node



(b) Overlaying graph nodes from the KGML representation of the pathway



(c) Final visualization

Figure 5.5: Visualizing gene expression data on KEGG pathway maps

5.1.2 Visualizing Gene Expression Data

In addition to the coloring of nodes in the pathway maps, Kvik is capable of visualizing gene expression profiles for the entire underlying dataset. When users want to inspect a single gene, the Kvik Browser opens an information panel containing a visualization of the gene expression profile using the D3 JavaScript library. Bar plots visualizes the difference between cases and controls. As with the gene expression values added to the pathway maps, Kvik uses the same approach to retrieve the gene expression profiles.

5.1.3 Visualizing Research Data

The Kvik browser adds information from the KEGG database to the info panel that opens when a user selects a gene. This info panel contains information such as the description of a gene and other background information about it. The Kvik Browser also adds a list of pathways this specific gene is a member of. To indicate their similarity to the pathway in the main view, a small bar

is visualized adjacent to the pathway names. Similarity is measured as the number of common genes between two pathways.

5.1.4 Extensibility

With the modular design of the Kvik Browser it is possible to add features in the future. If the Kvik Browser must visualize large quantities of data, it may be interesting to move to visualization libraries that leverage GPUs to provide more interactive visualizations.

5.2 Frontend

The Frontend is the component that connects the Kvik browser and the Backend, providing an interface for the Kvik Browser to retrieve expression data and other information. The Frontend provides a Representional state transfer (REST) Application programming interface (API) for Kvik Browsers to connect to. The RESTful interface consists of three groups of resources: i) a data resource; ii) a information resource; and iii) a visualization resource. Table 5.1 contains a detailed description of the resources and their methods.

The Frontend is designed as a stand-alone process that exposes an HTTP REST interface. It uses the GoRest[8] framework to implement the REST interface.

5.2.1 Data Resource

The data resource provides the Kvik Browser with an interface to the NOWAC Data Engine. The Frontend exposes two methods for either retrieving or submitting data. To make the queries to the NOWAC Data Engine simpler, the Query Engine forwards the query to the NOWAC Data Engine and returns the result to the client.

5.2.2 Information Resource

The information resource contains the interface to the KEGG databases. It provides the Genomic Browser with information on e.g. the name or description of a gene, the amino-acid sequence of a protein or the common genes shared

8. code.google.com/p/gorest

Table 5.1: The RESTful interface of the Frontend

Resource	Description
GET $/dataengine/\{query\}$	Performs the given query on the NOWAC Data Engine
POST $/dataengine/\{data\}$	POST the given data to the NOWAC Data Engine
GET $/info/\{item\}$	Returns all information possible for the given item
GET $/info/gene/\{id\}/pathways$	Returns a list of pathways a gene is member of
GET $/info/gene/\{ids\}/commonPathways$	Returns a list of common pathways between the specified genes
GET $/info/pathway/\{id\}/name$	Translates pathway id to readable name
GET $/info/pathway/\{ids\}/commonGenes$	Returns a list of common genes between the given pathways
GET $/vis/pathway/\{id\}$	Returns a visualization of the specified pathway
GET $/vis/gene/\{id\}$	Returns a visualization of the specified gene

between two pathways.

The information resource may issue mulitple queries to the KEGG databases. For example, returning a list of common genes shared between two pathways require the Frontend to issue two queries to KEGG. To retrieve a list of genes in pathway A, and one to retrieve the list of genes in pathway B. Then the union of these are returned to the Kvik Browser.

5.2.3 Visualization Resource

The visualization resource collects data and generates visualizations for the Kvik Browser. It combines the data and the information resource to generate visualizations of both gene expression data and biological pathways as seen on figure 5.2.

5.2.4 Extensibility

The extensibility of the Frontend allows new data sources to be added with little development effort. Adding new data sources to the Frontend only requires

adding or modifying the methods in the REST interface and implementing the required action. A suggestion from our researchers is to add information from GeneCards[9] in the gene information panel. Updating the information resource by adding an additional query to GeneCards allows the Kvik Browser to view this updated information without any modifications.

5.3 Backend

5.3.1 KEGG

Kvik uses KEGG as the primary source of information. The KEGG database is frequently updated, and is freely availible for academic users via their REST API. There is also an File Transfer Protocol (FTP) subscription availible, but at a fee. To avoid this fee, and having to manage the database and keeping it updated, Kvik uses the KEGG REST API. Through the REST API the entire KEGG database is available, but because of their location in Japan, the average response times for retrieving information about a pathway is over a second. For an interactive exploration system this is unacceptable, making it necessary for the Backend to host a local cache with requests to the KEGG database. The cache writes every response from KEGG to disk, allowing the Backend to retrieve results locally before contacting KEGG in case of a cache miss.

The KEGG component of the Backend is implemented as a Go package that can be used in any of the components of Kvik. The KEGG component is responsible for executing queries to KEGG and parsing the responses. The KEGG databases responds with data stored as Tab Separated Values (TSV) for information about genes and pathways, KGML files to describe pathways and PNG pathway images.

5.3.2 NOWAC Data Engine

The NOWAC Data Engine is a stand-alone service that is suitable for computer cluster environments, which can provide both storage and computational resources. Currently we run in on a single server since it has enough resources. The NOWAC Data Engine consists of three layers: i) a RESTful Interface that is used by the Frontend; ii) a stand alone Execution Engine for performing statistical analyses; and iii) a data management component responsible for reading raw data from the NOWAC biobank and making it accessible for the execution engine.

9. genecards.org

Table 5.2: RESTful Interface of the NOWAC Backend

Resource	Description
GET $/gene/\{id\}$	Returns all gene expression data on the given gene
GET $/gene/\{id\}/avg$	Returns the mean difference in gene expression between cases and controls in the dataset
GET $/gene/\{id\}/std$	Returns the standard deviation of the differences between cases and controls in the dataset
GET $/gene/\{id\}/var$	Returns the variance of the differences between cases and controls in the dataset
GET $/background/\{CC\ id\}/$	Returns background information about the specific case-control id
POST $/setscale/\{scale\}$	Allow users to specify which scale they want the results in, logarithmic or absolute

When a query arrives at the NOWAC Data Engine, it performs a three-stage process of handling the request. First the Data Engine extracts the applicable data from the data management layer. If the request specified that the data should run through a computation step, the execution layer performs the computation. When the computation completes the Data Engine returns the result of the query back to the Frontend.

Interface

The NOWAC Data engine offers a HTTP REST API, that exposes both methods for extracting raw expression data, as well as methods for performing statistical analysis on the expression profiles. Table 5.2 shows an overview of the resources and methods.

The Interface is implemented with the gorest package. It is responsible for retrieving the applicable data for the query and performing the correct Remote Procedure Call (RPC) on the Execution Engine. When the Execution Engine completes the call it return the results to the Kvik Browser.

Because of the security requirement, the Backend exposes only aggregated gene expression values to the Kvik Browser. No raw gene expression data is ever transported out of the Backend.

Execution Engine

The Execution Engine is as a stand-alone component that receives input data, performs some computation on that data and returns it on completion. This Execution Engine has support for executing multiple statistical analyses provided by our collaborators.

Because of the limited statistical libraries in Go, the NOWAC engine has a separate Execution Engine accessible through RPC. This RPC module is implemented in Python, exposing a JavaScript Object Notation (JSON)-like RPC interface over ZeroMQ[10]. Using python packages such as RPy[11] it is possible to add open source analysis tools like Bioconductor[12] to the RPC interface with little developer overhead. With the ability to execute R code our collaborators can upload their methods and view the results within the Kvik Browser. Since the current data sizes are relatively small the RPCs request contains both the method to be performed as well as the data to perform the operation on. Separating the computational parts of the implementation allow future versions to add a more complex statistical pacakge without affecting the other components of the system.

Data Management

The datasets used in the NOWAC study are both scientifically and commercially valuable, making it necessary to store it in a secure environment. Upon initialization, Kvik loads the necessary datasets from disk and stores the entire biobank in-memory during execution. Being in-memory, Kvik allow fast queries for either gene expression data or extracting subsets of the dataset.

The NOWAC Data Engine manages the gene expression data naively. Since the Kvik Browser performs queries either for a specific gene or a case-control pair, the NOWAC Data Engine uses in-memory maps[13] to store the data.

The NOWAC biobank is stored as Comma Separated Values (CSV) files on a compute node on Stallo, a super computer at The University of Tromsø. These files are parsed and read into memory by the NOWAC Data Engine on initalization. Since the raw data files use Illumina probe identifiers and not gene names from the KEGG database, the probe ids are translated using the lumiHumanIDMapping R package[38]. The Data Engine can add more datasets

10. zeromq.org
11. rpy.sourceforge.net
12. bioconductor.org
13. Maps as in the Go map type that implements a hash table

by simply reading more input data. Currently the implementation has support for parsing CSV files with gene expression values, but new parsers can be added to the management component to support more formats.

NOWAC Dataset

Kvik uses a subset of the NOWAC biobank as the main source of gene expression data. The dataset consists of gene expression profiles collected at time of diagnosis for 77 women. In addition to the 77 women, expression data for 77 controls were also added to the biobank. The rows describe the gene expression for individuals. Using the Illumina Human WG Version 3 chip our collaborators collected expression levels for 9101 genes, resulting in the dataset outlined on table 5.3. In addition to the gene expression values, background data on the samples is also collected. These hold information on which stage of cancer the patients were in, the type of cancer and details about the lab environment the samples come from. See table 5.4 for an illustration of the background dataset.

Table 5.3: Dataset layout

CC Id	Probe n_0	Probe n_1	$\cdots$	Probe n_{k-1}
...	...	...	...	...
1234	123.4	432.1	$\cdots$	213.4
1234_1	321.4	234.1	$\cdots$	312.4
...	...	...	...	...

Table 5.4: Background dataset layout

labnr	CC Id	case_ctrl	ET	EN	stage
...	...	...	...	...	...
...	...	...	...	...	...

5.3.3 Extensibility

The Backend is not restricted to only using the KEGG databases and the NOWAC biobank. Additional sources can, and will, be added in later versions of Kvik. The REST interface allow implementations in other programming languages or systems. Since the NOWAC Data Engine is a stand-alone component it is possible to introduce storage backends such as HDFS or Hbase and execution

engines such as Spark[14]. To incorporate these into an interactive exploration system, such as Kvik, the systems must provide fast response times but will be an interestin topic with the growth of the future datasets.

14. spark.apache.org

6

Related Work

In the last decades the methods for visualizing and exploring biological data has improved greatly. The need for tools that can manage the growing datasets and allow fast exploration of these is growing. Especially tools that generate massive amounts of data, like high-throughput DNA sequencers, has driven the need of novel data exploration tools. Today there are a wide spectrum and large number of different tools, but biology research projects often require custom built solutions for their specific problem set. For a more thorough review of different visualization techniques and systems, see the author's special curriculum report [13].

6.1 KEGG

KEGG is a freely available collection of 19 databases integrating genomic, chemical and systemic functional information [20]. One of its core systems is the PATHWAY database which as of January 2014 is containing 456 manually drawn pathways [39]. The PATHWAY database contain molecular interactions and reaction networks for metabolism, human diseases and cellular processes. The KEGG website hosts pathway maps for researchers to investigate. These maps have hyperlinks on every node allowing further inspection of either pathways, genes, reactions or enzymes. From the KEGG website, users can download pathways as Portable Network Graphics (PNG) images or text files containing the KGML representation of the pathway.

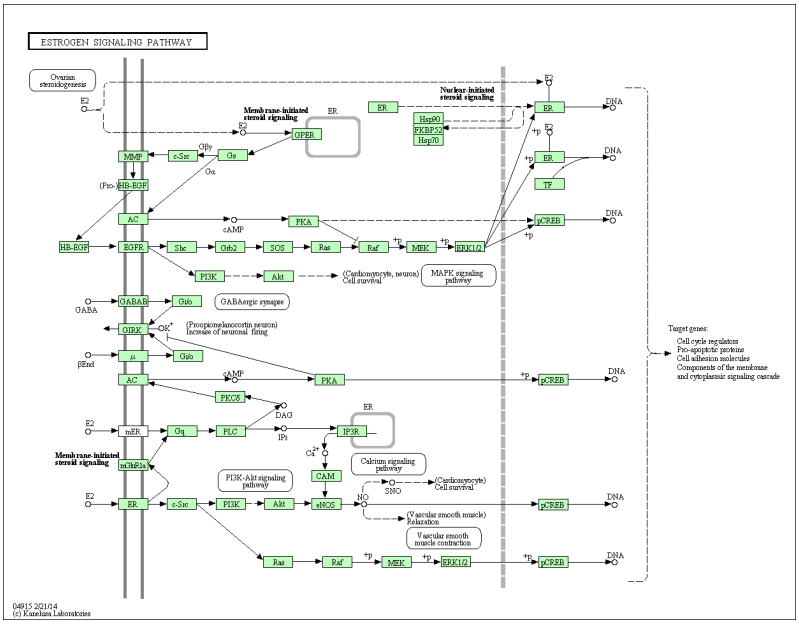


Figure 6.1: Estrogen Signaling Pathway from KEGG. Image from rest.kegg.jp/get/hsa04915/image

KEGG allow developers to contact the entire database through a FTP [1] site and a REST interface [2] . The FTP site requires an expensive license (5,000 USD) but the REST interface is free of charge. There are multiple exploration tools that use KEGG as a data source, either using the static pathway images or the KGML interface to construct own pathway visualizations. The decision to use KEGG as the primary data source for the visualization system in Lantern was the combination of the open REST interface, our collaborators prior experience and the number of pathways available in KEGG.

6.2 BioCarta

BioCarta is a interactive online resource targeted to the life science research community [21]. It is a resource much similar to KEGG that allows resarchers to view pathway maps and inspect single genes. Opposed to KEGG, the pathway maps found in BioCarta are visually different in that they are more similar to traditional text-book illustrations. Similar to KEGG, researchers can inspect genes by clicking hyperlinks opening information about the genes in a separate windows. Unfortunately BioCarta does not have any availible

1. Found at ftp.genome.jp
2. Found at rest.kegg.jp

API, making it necessary to download pathway images manually. Figure 6.2 shows the CARM1 and Regulation of the Estrogen Receptor pathway from BioCarta.

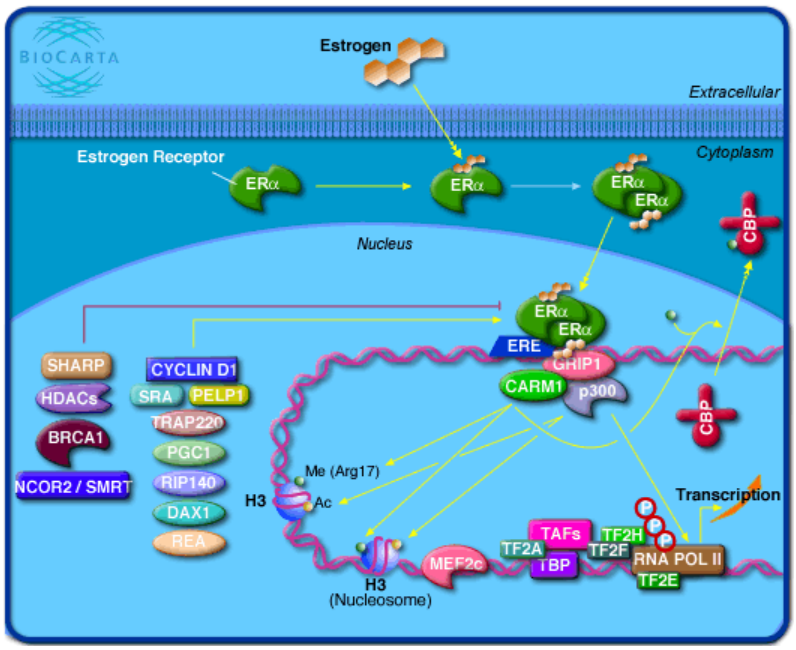


Figure 6.2: CARM1 and Regulation of the Estrogen Receptor. Figure from http://www.biocarta.com/pathfiles/h_carm-erPathway.asp

6.3 Caleydo

Caleydo is an open source visual analysis framework targeted at biomolecular data [40]. As of February 2014 Caleydo consists of four projects: StratomeX [41], enRoute [19], Entourage [25] and LineUp [42]. Caleydo has the capability of visualizing multiple biological pathways simultaneously. Originally using a 2.5D view for arranging pathways in a stack with edges interconnecting different pathways, highlighting common nodes [26]. From version 3.1.0 the Caleydo framework incorporated the Entourage visualization technique for visualizing multiple pathways [43].

Caleydo make use of the existing pathway layout supplied by KEGG and BioCarta for visualizing single pathways. It renders pathways as 2D graphs with nodes and edges placed at coordinates from the KEGG and BioCarta layout. In the final pathway visualization Caleydo render the pathway graphs on top of images from KEGG and BioCarta, hiding internal edges. Using this technique the visualizations contain contextual information from the static images, but

it is possible to generate views containing multiple pathways.

Caleydo can color nodes according to gene expression levels, but cannot change position from the original layout. To visualize mulitple pathways, Caleydo render these as stack of images interconnected by edges between shared nodes in the different pathways. In the original version of Caleydo, researchers could store important pathways in a designated view for quick access.

Caleydo is a open source framework written in Java using the Java OpenGL (JOGL) library for rendering, requiring that the system has got Java installed. The source code is availible at GitHub [3] and is frequently updated.

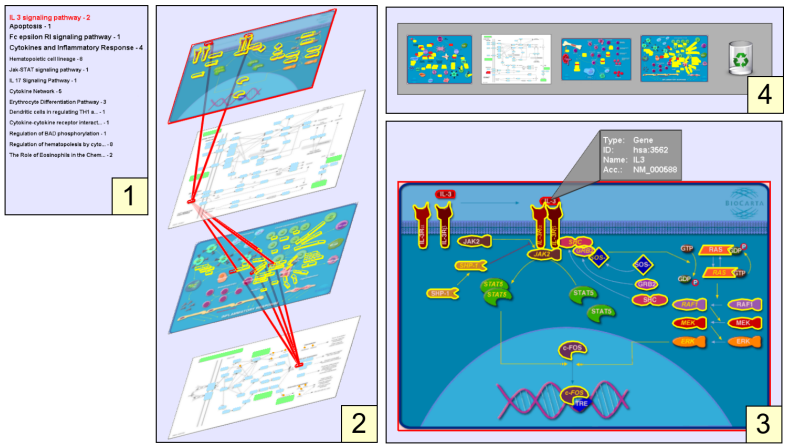


Figure 6.3: Navigation of Pathways in the original version of Caleydo. 1 shows possible pathways to explore, 2 illustrates the stacked view, 3 a detailed view of a pathway, and 4 stored pathways for later inspection. Figure from [26]

6.3.1 StratomeX

StratomeX is a integrative visualization tool that allows investigators to explore relationships of candidate cancer subtypes [41]. The system visualizes datasets as columns and cancer subtypes as bricks within the columns. StratomeX uses heatmaps, parallel coordinates or histograms within the bricks to visualize heterogeneous datasets To indicate relationships across datasets, it displays ribbons between the different columns.

3. github.com/Caleydo/caleydo

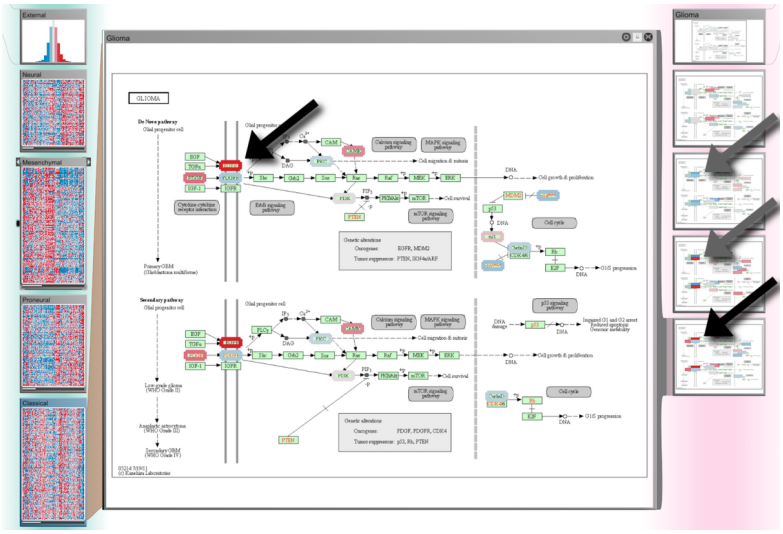


Figure 6.4: Cancer subtypes in the context of pathways. Illustrates how StratomeX visualizes expression data and biological pathways. Figure from [41].

6.3.2 enRoute

enRoute is an exploration tool that allow researchers to explore experimental data from paths that are dynamically extracted from biological pathways [19]. It visualizes biological pathways by extracting graph nodes from the KGML description of a pathway and overlaying them on top of the static pathway image from the KEGG database. enRoute highlights selected nodes within a pathway and visualizes the associated experimental data for side-by-side comparison. The system supports visualization of expression levels inside the nodes of the pathway maps.

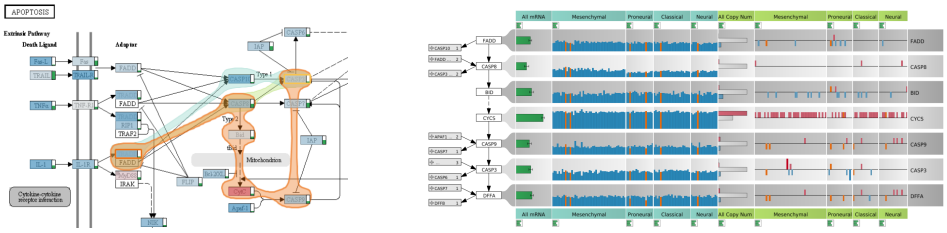


Figure 6.5: Pathway and expression data visualized in enRoute. Figure from [19].

6.3.3 LineUp

LineUp is an exploration technique for visualize multi-attribute rankings [42]. The authors use LineUp it to investigate how different universities rank ac-

cording to some property, for example their academic reputation or their world wide ranking has evolved over time. To our knowledge LineUp is not used for genomic data yet, however the authors plan to use it to rank genes, clusters and pathways in the future [42]

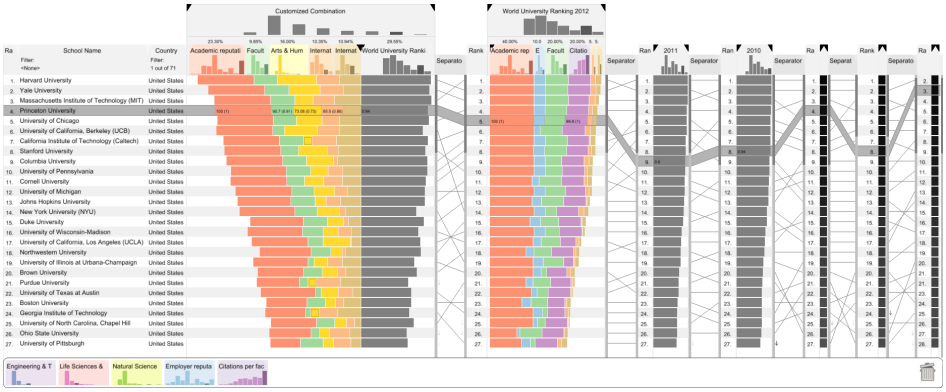


Figure 6.6: Rankings of the top Universities visualized in LineUp. Figure from [42].

6.3.4 Entourage

Entourage is a visualization technique that provides contextual information when visualizing multiple related pathways. It uses a single focus pathway for main interaction and exploration, and visualizes only what is important to researchers from other related pathways. Bubble Sets [44] highlights the selected nodes within a pathway. Entourage visualizes subsets of related pathways to give context information about the location of user selected genes within related pathways.

Since pathway maps generally are large in size, this technique allow users to view and interact with a large number of pathways given the relative small monitor size commonly used by researchers. Entourage uses the enRoute technique to visualize experimental data [25].

6.4 VisANT

VisANT is a freely available open source application for integrating biomolecular interaction data into a cohesive, graphical interface [22]. It is a stand-alone desktop application that integrates multiple online datasets and databases (for example KEGG) into a single application allowing thorough exploration of pathways and systems biology. The core of VisANT is its graph visualization system, capable of visualizing pathway or other interaction networks. To

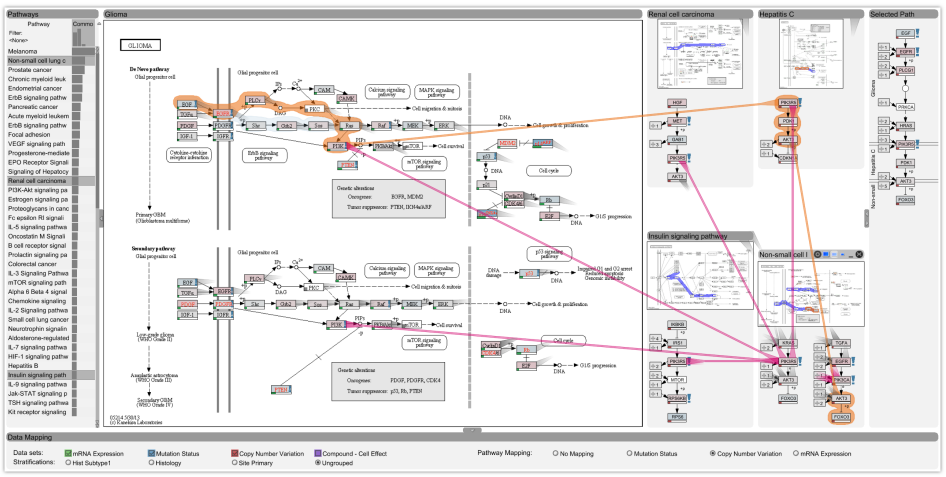


Figure 6.7: The Glioma pathway visualized in Entourage. Figure from [25].

construct graphs from different pathways it uses KGML files from KEGG which describes nodes and edges. The pathways are then reconstructed and visualized using multiple layout algorithms including force-direction, or a KEGG based layout using coordinates specified in the KGML files. VisANT is a general purpose network visualization system and lacks support for visual guidelines such as cell walls found in static KEGG images.

It is a three tiered system consisting of: a (i) a data layer for storage; (ii) a data control layer for interacting with the multiple data sources in the data layer; and (iii) a data presentation layer for visualization and presentation of data. It is a Java application requiring the JRE to run, and uses a VisANT backend to deliver pathway maps and other information to the user. This backend is responsible for downloading an integrating multiple databases including KEGG [45]. The VisANT application has been regularly updated since 2003 with the last update in August 2013.

6.5 KGML-ed

KGML-ED is a tool for exploring and editing pathways generated from KGML files supplied by KEGG. It provides the functionality of editing existing pathways and exporting KGML versions of these for later inspection. KGML-ED allow users to start exploration at an overview of multiple pathways and expand those of interest. It integrates mulitple pathways into the same view to explore their relationships.

It is a stand-alone Java application that runs on any system with the JRE installed and requires users to manually download KGML files to visualize them in KGML-ed.

It also requires that users download database files and store them locally for example to lookup compound names from KEGG identifiers. The software is freely available for download online, but has not seen updates since mid 2007.

6.6 KEGGViewer

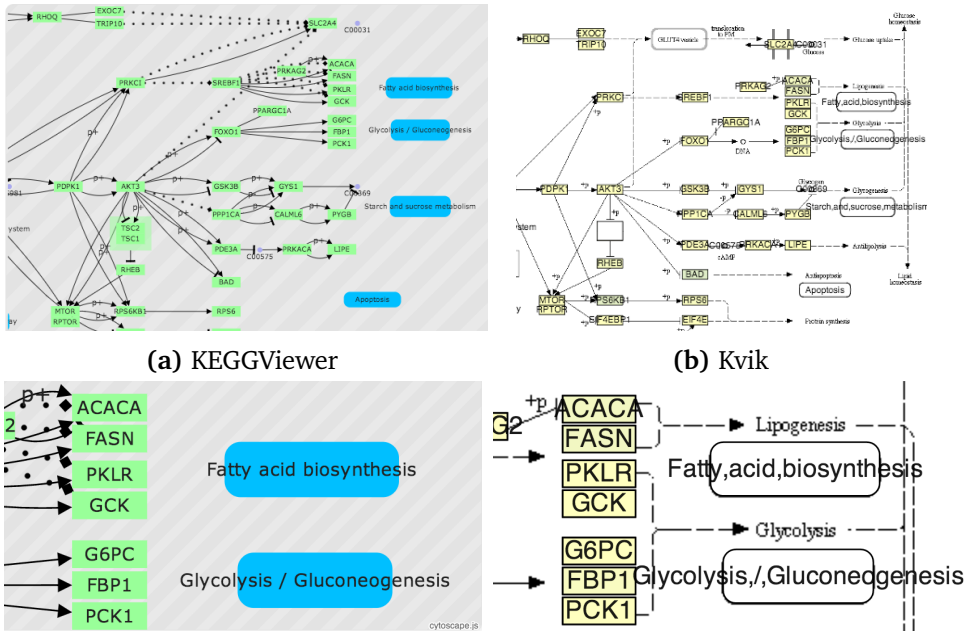
KEGGviewer is a BioJS [46] component to visualize KEGG pathways and to allow their visual integration with functional data [24]. It uses the KGML representations of pathways from the KEGG REST API to build pathways, and visualizes them in a web browser using the Javascript library Cytoscape.js. It makes it possible for researchers to embed pathway maps into web pages with minimal developer effort. Since KEGGViewer only uses the KGML representation to generate the visualizations, they are lacking both in contextual information as well as nodes and edges. Figure 6.8 illustrates the difference in the final visualization of the same pathway using KEGGViewer and Kvik.

Since KEGGViewer is an open source tool, we downloaded and evaluated the system against Kvik. The usage examples of KEGGViewer visualizes the insulin signaling pathway.[4] Visualizing this pathway in KEGGViewer and Kvik revealed that the KEGGViewer used an average of 2.03 seconds (SD $= 1.05$ seconds) while Kvik only used an average of 0.46 seconds (SD $= 0.67$ seconds) to load the visualization. The main contribution to the load time in KEGGViewer is the latency to the KEGG servers. KEGGViewer performs a HTTP GET reqeust every time it visualizes a pathway, while Kvik caches the request to KEGG in the Backend. This reduces load times to less than a second.

6.7 VANTED

VANTED is a tool for the vivsualization and analysis of networks with related experimental data [23]. It allows visualization and exploration of biological networks, including KEGG pathways, along with experimental data imported either from local files or remote databases. Pathways are visualized as networks that can be arranged in a wide variety of layouts including force-

4. genome.jp/kegg-bin/show_pathway?hsa04910



(a) KEGGViewer **(b)** Kvik

(c) KEGGViewer zoomed in **(d)** Kvik zoomed in

Figure 6.8: The Insulin signaling pathway visualized by KEGGViewer and Kvik. Both contextual information and nodes and edges are lost in the KEGGViewer visualization.

directed layout, tree layout, circle layout or expression matrix layout [47]. Unlike other tools for visualizing biological networks, VANTED enables users to visualize statistical data within the nodes or edges in the network. For example gene expression data can be visualized on top of nodes in a pathway.

VANTED is an open source application written in Java released under the GNU Public License (GPL), and requires the JRE to run. It is based on the graph library Gravisto, and supports extension written using BeanShell or jRuby [23].

6.8 Pathway Projector

Pathway Projector is a pathway browser that uses the Google Maps API for navigating integrated pathway maps based upon the KEGG Atlas [27]. It allows users to navigate from an overview of the complete KEGG Atlas and down to gene and enzyme level through panning and zooming displaying content in a single view. In contrast to multiple other systems for visualizing pathways,

Pahtway Projector does not require any installation of additional software, like the JRE, except the standard web browser you find on practically every computer or smart device.

7

Evaluation and Use Case

The main goal of the evaluation is to evaluate the design choices and implementation of Kvik. To do this, Kvik is evaluated using three metrics i) latency, ii) resource utilization and iii) system scalability. In addition we conducted an informal evaluation of the system with regards to usability from an end-user's perspective.

Kvik is a system for interactive exploration of multi-omics data, making latency the most important metric to evaluate. The latency is measured as the time from a user clicks or interacts with the system, until the resulting visualization or information is shown.

It is also helpful to measure resource utilization when using the Kvik Browser. Since the Kvik Browser only visualizes the data and handles user input, it is expected that it does not use excessive memory or CPU power. In addition, the proposed design of the backend systems should also handle the data from the NOWAC cohort and the KEGG database.

There are several factors that have an impact on the observed latency and resource usage in Kvik:

Pathway Size The number of entities (genes, proteins or compounds) in a pathway affects the latency since the Kvik Browser gathers information from the Backend for every entity. In addition to the information, gene expression values are fetched from the NOWAC Data Engine.

Dataset Size Kvik uses a subset of the NOWAC cohort which contains 9102 expression values for 154 individuals. Increasing the dataset size, e.g. the number of patients, has an impact on the start-up time of the Data Engine. Executing the simple statistical analyses (mean, Standard deviation (SD) and variance) is not affected by the dataset size. With more advanced statistical analyses this is expected to change in the future.

KEGG Response Time Kvik relies on the KEGG database for all background information about genes and pathways. Since Kvik uses the freely availible REST API the response time has significant impacts on the end-to-end latency of the system.

7.1 Experimental Setup

Kvik was tested and evaluated using commodity hardware. All experiments were run on a 2013 Mac Mini with a 2.66 GHz Intel Core i7 processor, 16 GB 1600 MHz DDR3 SDRAM and a 1 TB Fusion Drive, connected to two 1920x1080 Acer P246HBD displays.

The experiments were run in OS X 10.9.2 running Firefox version 29.0. To evaluate the visualizations, the JavaScript library Benchmark.js[1] was used. The experiments on the backend systems uses the Go testing package[2] in Go version 1.2rc3.

The experiments that measures the end-to-end latency in the Kvik Browser uses real data form the NOWAC databank and the KEGG database. This dataset contained 9102 expression values from 154 individuals. To measure how the NOWAC Data Engine scales to larger datasets it was necessary to generate random datasets for the experiments. This is due to the present unavailability of larger datasets. The following sections give a detailed description of the datasets.

7.2 Experiments

To evaluate the latencies of the visualizations in Kvik, we measure the load times of different pathways, and the load times of different genes in the gene inspection view. To measure how well the NOWAC Data Engine scales, it was

1. benchmarkjs.com
2. golang.org/pkg/testing

measured both with regards to load time of the data, as well as memory consumption. Finally, to evaluate the need for a local cache in the Backend, the latencies of the visualizations in Kvik were measured without caching enabled.

To be able to run many experiments and get accurat results, it was necessary to automate user actions. We used the Benchmark.js library to execute the generation of different visualizations automatically. The JavaScript library executes a given section of source code a given number times, and does not require any modifications to the source code being measured[3]. The results for the visualization latencies are for 200 samples. Evaluating the Kvik Browser is done by measuring runtimes and sending them to an external server for further inspection. Most web browsers have the capability of profiling web sites, but since they require manual user interaction, this approach was not chosen. Profiling web pages manually is just too tedious to get statistically significant results.

Accurate measurements of the performance of the NOWAC Data Engine was retrieved through multiple runs of the Go testing package. As with the latency experiments the results are for 200 samples.

7.2.1 Load Pathway

To measure the load times relative to the pathway size, four pathways were chosen. Table 7.1 lists the pathways and their sizes.

Loading a pathway requires the Kvik Frontend to retrieve the both the pathway description and image from KEGG, in addition to the gene expression for the different genes in the pathway. The latency is measured from the user clicks on a pathway to visualize in the Kvik Browser, until it is rendered in the web browser.

Table 7.1: Pathways used to evaluate Kvik

Id	Name	Number of nodes
hsa04630	Jak-Stat signaling pathway	35
hsa04915	Estrogen signaling pathway	74
hsa4151	PI3K-Akt signaling pathway	120
hsa05200	Pathways in cancer	267

Figure 7.1 shows the distribution of the measured latencies to load each of the

3. Note that asynchronous functions require special setup

four pathways. From the histograms it is possible to conclude that the latencies follow a log-normal distribution. From this observation the results on table 7.2 are presented using geometric mean and geometric (or multiplicative) SD to give a more correct description. Note that because of the geometric measures, it is important to notice that 68.3% confidence interval are not defined by $\bar{x} \pm s$ where $\bar{x}$ is the mean and s is the standard deviation confidence but $\bar{x}^{\star\, x}/\, s^{\star}$, where $\bar{x}^{\star}$ is the geometric mean and $s^{\star}$ is the geometric SD[48].

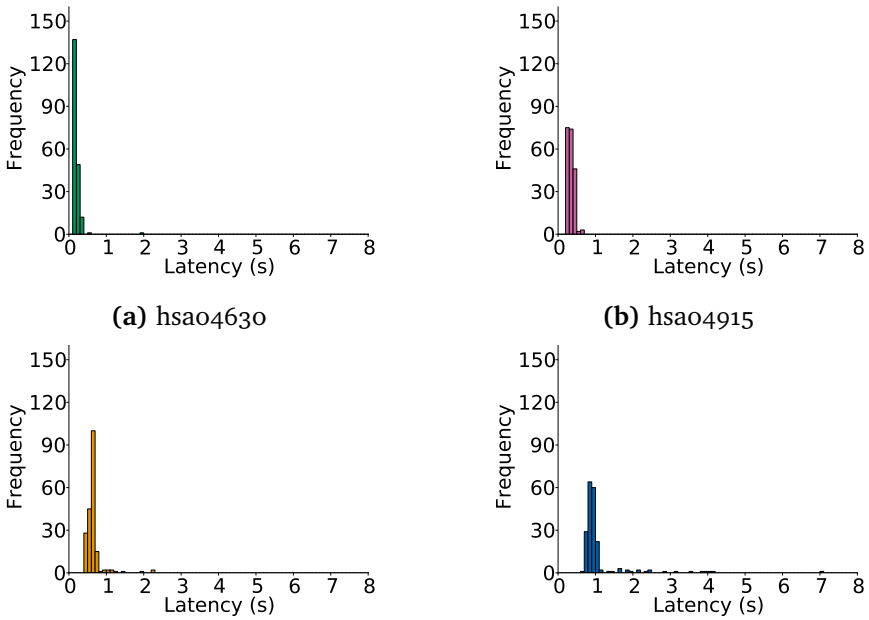
Table 7.2 shows the results for loading different pathways. The results show that for relatively small pathways the average load times are around 0.3 seconds. For larger pathways the load times increase to about a second. Note the large variance in the measured latencies. For all pathways the SD is more than 1.2 seconds indicating in the similar variability in load time for all pathways.

Table 7.2: Time to load pathway visualization

Id	Geometric mean	Geometric SD
hsa04630	0.20s	1.30s
hsa04915	0.34s	1.20s
hsa04151	0.62s	1.26s
hsa05200	1.00s	1.48s

Figure 7.2 illustrates the cumulative distribution of the measured latencies for the different pathways. For the three first pathways over 95% of reqeuests are completed within a second. With the large hsa05200 (Pathways in cancer) pathway, 95% of the requests are completed after about 2 seconds. As reported in [18], response times more than 1 second is intrusive on a user's line of thought. In Kvik all visualizations and events display a progress indicator letting users know that the system is performing some work. According to our collaborators the load times of the pathways were satisfactory and the visualizations were presented within a reasonable amount of time. Still, this pathway has a long tail of requests that is not served after several seconds. We believe the garbage collector in Firefox is responsible for the long tail. Larger pathways allocate more memory and if this is not handled correctly there might be issues with memory management. We have not investigated this hypothesis yet.

To reduce the latencies client-side caching could have been used. Kvik loads the pathway images from KEGG every time it visualizes a pathway. These images rarely change within a session and may have been cached in the client web browser. Since the pathway images are relatively small in size HTML5 can easily provide sufficient storage for these. As an example, the



(a) hsa04630 **(b)** hsa04915

(c) hsa04151 **(d)** hsa05200

Figure 7.1: Distributions of the measured latencies to visualize pathways

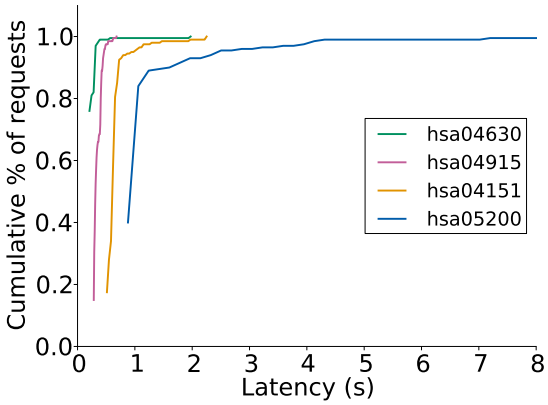


Figure 7.2: Cumulative distribution of the measured latency

image of the large hsa05200 pathway is approximately 93KB large. With the 5MB available local storage it is possible to store more than 5 000 copies of this pathway image (2% of all KEGG images).

7.2.2 Inspect Gene

The second visualization evaluated in Kvik is the gene inspection view. This panel opens when users click on a gene and visualizes gene expression for the entire dataset as well as other information from KEGG and a overview of pathways of interest. To evaluate this visualization, four different genes were chosen. These genes differ from each other by the number of pathways they are a part of. Table 7.3 lists the different genes and the number of pathways they are a part of. This is an interesting number since the Kvik Browser has to visualize more data with a higher count of pathways.

As in the previous experiment, the latency from user input to the display of a visualization is measured. In this case, from a user clicks on a gene in the pathway visualization and until the information panel is opened.

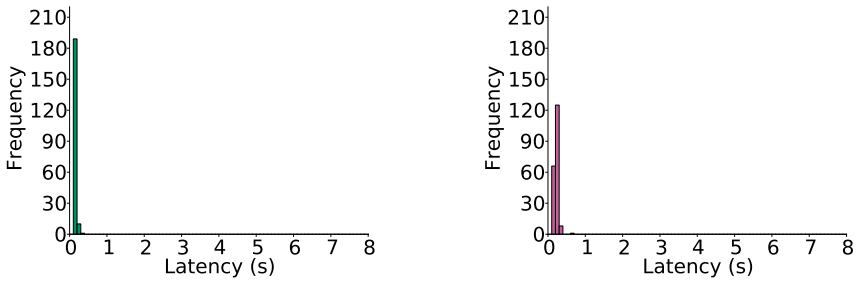
Table 7.3: Genes used to evaluate Kvik

Id	Name	Number of pathways
hsa:4313	MMP2	6
hsa:3303	HSPA1A	12
hsa:6654	SOS1	32
hsa:5604	MAP2K1	55

Figure 7.3 show the distribution of the measured latencies to load the gene information panels. As for the first two genes, all requests complete within a second. For the other two the variance is large and the results show a high percentaage of requests not completing until after 5 seconds. Figure 7.4 illustrates the cumulative distribution, and the long tail of requests for the last two genes is apparent.

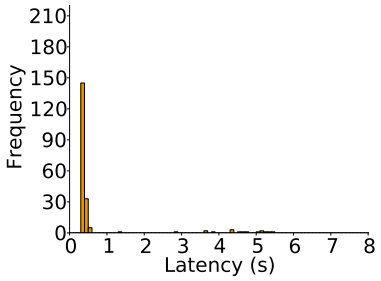
The latencies to the gene visualizations follow a similar log-normal distribution as the pathway visualizations. Table 7.4 include the results for the different genes. Although the averages fall under a second, the long tail of requests that fall outside a second is a serious issues. The load times of the info panel should be consistent for a gene, not ranging from under a second up to eight seconds. As with the pathway visualizations we believe that the long tail of requests is due to garbage collection in Firefox. We have not investivate this yet.

As with the pathway visualizations, client-side caching could improve performance.

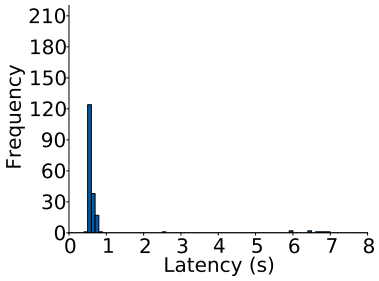


(a) hsa4313

(b) hsa3303



(c) hsa6645



(d) hsa5604

Figure 7.3: Distributions of the measured latencies to load gene information

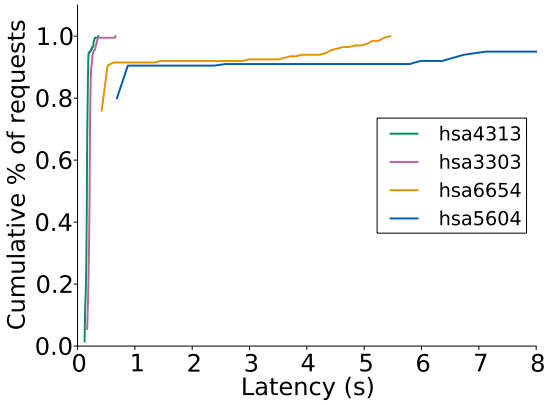


Figure 7.4: Cumulative distribution of the measured latency

7.2.3 Load Dataset

To evaluate the NOWAC Data Engine it was necessary to generate test datasets. From the discription in table 5.3 additional case-control pairs were added to increase the dataset size. The number of genes in humans are expected to stay the same, while the number of cases/controls will increase when new samples are added to the NOWAC biobank. The gene expression values for the cases/controls were generated randomly. It does not matter what the values

Table 7.4: Time to load gene details

Id	Geometric mean	Geometric SD
hsa:4313	0.16s	1.18s
hsa:3303	0.21s	1.18s
hsa:6654	0.47s	1.98s
hsa:5604	0.75s	2.11s

are.

The experiment datasets ranged in size from the original subset, and up to a dataset containing 9102 gene expression values for 3080 case-control pairs. The complete NOWAC cohort contains blood samples from 6000 women, making the 20x experiment dataset approximately 50% of the available gene expression data.

In the experiment the NOWAC Data Engine is initialized using different datasets. The initalization reads the dataset from a disk, generates the in-memory structures and initializes the REST interface. We measure the time taken from the NOWAC Data Engine starts to read the dataset until it can receive requests from the Frontend.

Table 7.5 show the results from initializing the NOWAC Data Engine. The initialization reads the dataset from disk and stores it in main memory. To load the original dataset consisting of 9102 genes and 77 case-control pairs takes on average 2.63 seconds and uses 353 MB of memory. As the results show both memory usage and load time appears to scale linearly. From the results it is apparent that the NOWAC Data Engine can hold the experiment datasets in memory. Datasets that grow larger than available DRAM will probably degrade performance, making it necessary to introduce either enough resources or use other distributed systems.

Table 7.5: Time to load NOWAC dataset

	Time		Memory	
Size	Mean	SD	Mean	SD
1x	2.63s	0.13s	353.55 MB	0.04 MB
2x	5.39s	0.26s	703.54 MB	0.06 MB
5x	13.94s	0.29s	1652.38 MB	0.10 MB
10x	28.06s	0.28s	3320.06 MB	0.19 MB
20x	58.18s	0.68s	6579.74 MB	0.85 MB

The statistical analyses in the NOWAC Data Engine operates on the columns of the gene expression table. Because the number of rows will add up to 3080 for the 20x dataset, the statistical analyses has not been evaluated. Since the analyses only perform simple tasks like calculating an average or standard deviation, this was left out.

7.2.4 KEGG Caching

We must cache requests to the KEGG database for two reasons: i) the reponse times; and ii) licensing issues with the REST API. First, since Kvik is an interactive system that uses KEGG extensively, e.g. performing over 200 requests to load the hsa05200 pathway, response times of several hundred milliseconds for every request are unacceptable. Second, as the API restrictions read: *"This service should not be used for bulk data downloads."*[4] it is necessary to limit the traffic to their servers.

To measure the impact caching has on the Kvik Browser, the latency experiments in 7.2.1 and 7.2.2 were run, each 200 times. The visualization of the large hsa05200 pathway generates 267 requests on information about the entities in the pathway, 1 request for the KGML representation of the pathway and 1 request to fetch the static pathway image. Visualizing this pathway 200 times results in over 53 000 requests to KEGG, clearly violating the terms of use and introducing unnecessary long latencies.

From table 7.6 we see the huge benefit in caching KEGG responses. For every pathway measured there is over a 140x speedup in load times when caching KEGG responses. For genes the benefit is smaller, but still significant. The size of the cache is small. Visualizing the four experiment pathways resulted in 693 elements in the cache adding up to 5.7MB used disk space server side.

7.2.5 Resource Consumption

To measure the load on the client computer when using the Kvik Browser, we measure CPU and memory utilization by Firefox when using the Kvik browser to load the four experiment pathways.

Figure 7.5 show a timeseries of CPU and memory usage when loading the different pathways from the previous experiments.

The results show that loading a large pathway compared to a smaller pathway

4. from kegg.jp/kegg/rest

Table 7.6: Comparing load times with and without caching of KEGG requests

Method	Caching Enabled	Caching Disabled
Load pathway hsa04630	0.20s	53.18s
Load pathway hsa04915	0.34s	59.63s
Load pathway hsa04151	0.62s	123.99s
Load pathway hsa05200	1.00s	142.30s
Inspect gene hsa4314	0.76s	5.35s
Inspect gene hsa3303	1.05s	4.92s
Inspect gene hsa6654	1.90s	5.10s
Inspect gene hsa5604	1.73s	5.32s

does not have any impact on CPU nor memory usage. Overall the Kvik Browser uses at peak 50% CPU and less than 4% memory when it is loading the visualization, and then settles down after the visualization has been rendered. It is safe to say that the Kvik Browser does not hog the CPU or use excessive memory.

Since the Kvik Browser is implemented using the HTML5 canvas, it is also interesting to evaluate the system on mobile devices. Early trials show promising results, but we still have left to investigate the performance thoroughly.

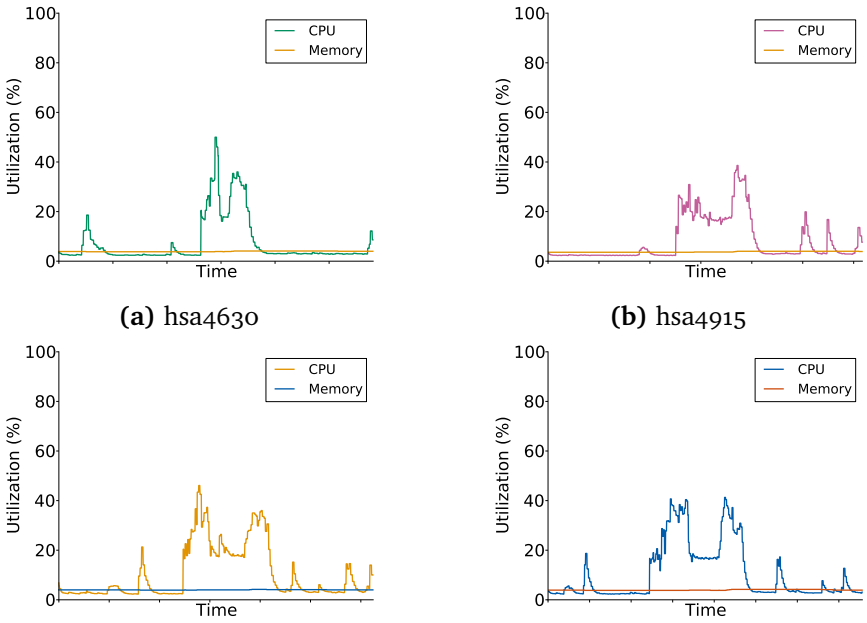
7.2.6 Comparison of different hardware

The previous experiments were run on high-end systems. To compare Kvik on lower-end systems we used a 5 year old mid-2009 Macbook Pro. The Macbook Pro had a 2.66 GHz Intel Core 2 Duo Processor, 8 GB of 1067 MHz DDR3 RAM. We ran the same experiments to evaluate the pathway visualization and inspect if newer hardware would speed up the load times.

Table 7.7 show the load times for the different pathways. Even with 5 year old hardware the performance is acceptable.

7.3 Usability

The development process of Kvik followed an iterative approach, with frequent meetings and continuous feedback about the usability and lacking features



(a) hsa4630 **(b)** hsa4915

(c) hsa4151 **(d)** hsa5200

Figure 7.5: CPU and memory utilization when loading different pathways

Table 7.7: Comparison of pathway load times

Id	Mac Mini	Macbook Pro
hsa04630	0.20s	0.48s
hsa04915	0.34s	0.97s
hsa04151	0.62s	1.91s
hsa05200	1.00s	4.00s

of Kvik. Through these meetings the following results emerged.

Generating pathway visualizations using the traditional layout of KEGG was a feature that was chosen early. We explored other techniques, e.g. using a force-directed layout, but these maps were unusable by the researchers who were used to browsing the hand-drawn pathway maps in KEGG. Providing pathway visualizations based only on the KGML representation was suggested, but since these quickly became too complex to navigate, our collaborators requested another alternative. Visualizing nodes on top of the static KEGG images was a game-changer for our collaborators. The researchers we're able to orient themselves in the traditional pathway maps, allowing them to start data exploration faster. After the pathway visualization was completed, researchers could browse through pathway maps of their choice, investigating

genes and their relationships interactively. The next step in the development process, was the design and implementation of the gene inspection panel. Also with this component it took several iterations before the results were satisfactory to our collaborators. Throughout the project, adding features to Kvik was done using this iterative approach, by implementing and presenting one feature at a time.

From the iterative development process, adding components to Kvik became simpler with every feature. It is evident that collaborating in a team with frequent meetings is helpful both for biologists to come up with improvements and new features, but also for computer scientists to understand the problems from the end users' perspective.

With the development process that was used in Kvik, we believe that it builds a solid foundation for developing a more advanced exploration tool for the entire NOWAC cohort study.

We have identified three use cases discussed in Chapter 3.

8

Conclusion

This thesis presents Kvik, an interactive system for exploring the dynamics of carcinogenesis through integrated studies of biological pathways and genomic data. It provides researchers with a lightweight web application for navigating through biological pathways from the KEGG database and genomic data from the NOWAC postgenome biobank.

While state of the art data explorations systems for biological pathways and genomic data exploration run as stand-alone desktop applications, we have demonstrated that it is possible to design and implement such a system as a lightweight web application. Kvik consists of three separate components: i) The Kvik Browser, an interactive system for visual exploration of biological pathways and genomic data; ii) a Frontend that translates user interactions in the Kvik Browser into queries that retrieves data and visualizations; and iii) a Backend that provides the Kvik Browser with biological data and statistical analyses. We describe and demonstrate the advantages of separating components, allowing the researchers to explore large datasets using lightweight clients. Kvik is designed to handle both new data sources, analysis methods and visualization techniques. We have identified three new important use cases for Kvik, that can easily be added.

In collaboration with researchers from the NOWAC research group, we have performed a requirement analysis that targets the challenges of performing exploratory analysis of biological pathways and genomic data. From these requirements we have designed and implemented the first version. Using an

iterative approach we have improved the system through small development cycles by involving the end-users in the development process. Throughout the project we have gained valuable interdisciplinary experience in developing data exploration systems for use in cancer research.

Through an evaluation of the exploration tasks and workflow of an end-user, we demonstrate that Kvik has the capability of interactive exploration of genomic data and biological pathways. We measured load times in the Kvik Browser to be less than 1 second, demonstrating that it is an interactive system. We demonstrate that the NOWAC data engine scales to the full size of the NOWAC biobank.

We believe Kvik is important to enable novel discoveries from the data produced in the NOWAC project. It provides access to powerful compute and storage resources enabling the use of advanced statistical methods for the analysis. Finally, we use our experiences from developing Kvik to provide use cases and requirements for analysis, compute and data management infrastructure developed in our group and by others.

Kvik is released as an open-source project hosted at github.com/fjukstad/kvik.

9

Future Work

Kvik fulfills our initial requirements and have proven useful. During development and deployment we have identified areas of improvement. From section 3.2 our collaborators identified the worklow in a system for exploration of the multi-level NOWAC databank. While Kvik implements a simple exploration tool with a pathway browser and a gene viewer, our researchers require the addition of more advanced analyses like GSEA to perform more complex exploration of the NOWAC databank. The addition of such analyses is possible through the extensive Backend, but was not done due to time constraints.

The first version of Kvik supports a single user, and does not store history about the researcher's exploration. This is common in exploration tools for genomic data, but with future versions of Kvik we plan to allow multiple users and individual storage of results within the Kvik Browser. When different researchers start to use Kvik it is necessary to introduce user accounts, both to enforce restriction on the content the users can view, but also to allow users to store results in Kvik for later inspection. Currently there are no restrictions on what data the user can view in the Kvik Browser. Since the NOWAC databank cannot be published yet, access restriction is a feature that is required in every component of Kvik. With user accounts the researchers can also store their results or save pathways to view later. Introducing a web framework like Revel[1] makes it possible to add multi-user functionality with little developer

1. revel.github.io

effort.

Another feature in Kvik our collaborators requested was the ability to export a complete workflow, or session, much like the work done in Wrangler [49], Galaxy[2] or the syntax files in SPSS[3]. Users want to record the data exploration steps, e.g. the navigation through pathways and statistical analyses done, and share it with other researchers.

Currently, Kvik has only access to gene expression data at time of diagnosis. The NOWAC databank consists of large quantities of other research data, such as gene expression and exposure through questionnaires that have been collected over decades. Ultimately researchers want to develop new methods for diagnosis which relies on additional statistical analyses, data from multiple levels and integrated analyses. With the extensibility of Kvik this is possible with minor overhead.

Since Kvik is an interactive system, these systems must provide fast, subsecond response times. See [14] for a discussion of modern data-intensive computing systems and their application in processing of biological data.

The Kvik Browser is currently limited to visualizing one pathway at a time. We plan to incorporate visualization of multiple pathways in the same view, much like the functionality in Entourage [25]. Navigating multiple pathways is often necessary to understand complex diseases such as cancer, and keeping them in the same view can speed up the exploration.

2. galaxyproject.org
3. ibm.com/software/analytics/spss

Bibliography

[1] R. Siegel, D. Naishadham, and A. Jemal, "Cancer statistics, 2013.," *CA: a cancer journal for clinicians*, vol. 63, pp. 11–30, Jan. 2013.

[2] J. Couzin-Frankel, "Cancer Immunotherapy," *Science*, vol. 342, pp. 1432–1433, 2013.

[3] J. D. Watson and F. H. C. Crick, "Molecular structure of nucleic acids," *Nature*, vol. 171, pp. 737–738, 1953.

[4] J. C. Venter *et. al.*, "The sequence of the human genome.," *Science (New York, N.Y.)*, vol. 291, pp. 1304–51, Feb. 2001.

[5] E. S. Lander *et. al.*, "Initial sequencing and analysis of the human genome.," *Nature*, vol. 409, pp. 860–921, 2001.

[6] M. L. Metzker, "Sequencing technologies - the next generation.," *Nature reviews. Genetics*, vol. 11, pp. 31–46, Jan. 2010.

[7] S. D. Kahn, "On the Future of Genomic Data," *Science*, vol. 331, pp. 728–729, Feb. 2011.

[8] L. D. Stein, "The case for cloud computing in genome informatics.," *Genome biology*, vol. 11, p. 207, 2010.

[9] A. Sboner, X. Mu, D. Greenbaum, R. K. Auerbach, and M. B. Gerstein, "The real cost of sequencing: higher than you think!," *Genome Biology*, vol. 12, p. 125, 2011.

[10] F. Nansen, *Farthest north*. 1897.

[11] H. Johansen, *With Nansen in the North*. Giuseppe Castrovilli, 1899.

[12] S. I. O'Donoghue, A.-C. Gavin, N. Gehlenborg, D. S. Goodsell, J.-K. Hériché, C. B. Nielsen, C. North, A. J. Olson, J. B. Procter, D. W. Shat-

tuck, T. Walter, and B. Wong, "Visualizing biological data-now and in the future.," *Nature methods*, vol. 7, pp. S2–4, Mar. 2010.

[13] B. Fjukstad, "NOWAC Data Exploration." http://bdps.cs.uit.no/papers/capstone-bjorn.pdf, 2013.

[14] M. Ernstsen, "Mario. A system for iterative and interactive processing of biological data," 2013.

[15] N. Tractinsky, "Aesthetics and apparent usability: empirically assessing cultural and methodological issues," in *CHI'97 Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pp. 115–122, 1997.

[16] E. Lund, V. Dumeaux, T. Braaten, A. Hjartå ker, D. Engeset, G. Skeie, and M. Kumle, "Cohort profile: the Norwegian women and cancer study—NOWAC—Kvinner og kreft," *International journal of epidemiology*, vol. 37, no. 1, pp. 36–41, 2008.

[17] E. Lund, "An exposure driven functional model of carcinogenesis," *Medical hypotheses*, vol. 77, no. 2, pp. 195–198, 2011.

[18] R. Miller, "Response time in man-computer conversational transactions," *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pp. 267–277, 1968.

[19] C. Partl, D. Kalkofen, A. Lex, K. Kashofer, M. Streit, and D. Schmalstieg, "enRoute: Dynamic path extraction from biological pathway maps for in-depth experimental data analysis," in *2012 IEEE Symposium on Biological Data Visualization (BioVis)*, pp. 107–114, IEEE, Oct. 2012.

[20] Kanehisa Laboratories, "KEGG: Kyoto Encyclopedia of Genes and Genomes."

[21] D. Nishimura, "BioCarta," *Biotech Software & Internet Report*, vol. 2, pp. 117–120, 2001.

[22] Z. Hu, J. Mellor, J. Wu, and C. DeLisi, "VisANT: an online visualization and analysis tool for biological interaction data.," *BMC bioinformatics*, vol. 5, p. 17, 2004.

[23] B. H. Junker, C. Klukas, and F. Schreiber, "VANTED: a system for advanced data analysis and visualization in the context of biological networks.," *BMC bioinformatics*, vol. 7, p. 109, 2006.

[24] J. M. Villaveces, R. C. Jimenez, and B. H. Habermann, "KEGGViewer, a BioJS component to visualize KEGG Pathways," *F1000Research*, vol. 3, p. 43, 2014.

[25] A. Lex, C. Partl, D. Kalkofen, M. Streit, S. Gratzl, A. M. Wassermann, D. Schmalstieg, and H. Pfister, "Entourage: visualizing relationships between biological pathways using contextual subsets.," *IEEE transactions on visualization and computer graphics*, vol. 19, pp. 2536–45, 2013.

[26] M. Streit, M. Kalkusch, K. Kashofer, and D. Schmalstieg, "Navigation and Exploration of Interconnected Pathways," *Computer Graphics Forum (EuroVis '08)*, vol. 27, pp. 951–958, 2008.

[27] N. Kono, K. Arakawa, R. Ogawa, N. Kido, K. Oshita, K. Ikegami, S. Tamaki, and M. Tomita, "Pathway projector: web-based zoomable pathway browser using KEGG atlas and Google Maps API.," *PloS one*, vol. 4, p. e7710, 2009.

[28] F. H. CRICK, "On protein synthesis.," *Symposia of the Society for Experimental Biology*, vol. 12, pp. 138–163, 1958.

[29] E. Bianconi, A. Piovesan, F. Facchin, A. Beraudi, R. Casadei, F. Frabetti, L. Vitale, M. C. Pelleri, S. Tassani, F. Piva, S. Perez-Amodio, P. Strippoli, and S. Canaider, "An estimation of the number of cells in the human body.," *Annals of human biology*, vol. 40, pp. 463–71, 2013.

[30] G. Elgar and T. Vavouri, "Tuning in to the signals: noncoding sequence conservation in vertebrate genomes.," *Trends in genetics : TIG*, vol. 24, pp. 344–52, July 2008.

[31] G. Moore, "Cramming More Components Onto Integrated Circuits," *Proceedings of the IEEE*, vol. 86, 1998.

[32] K. S. Olsen, *Blood gene expression, lifestyle and diet - The Norwegian Women and Cancer Post-genome Cohort*. Doctoral thesis, University of Tromsø, 2013.

[33] National Human Genome Research Institute, "Biological Pathways," 2012. Accessed: 16/03/14.

[34] J. A. Fresno Vara, E. Casado, J. de Castro, P. Cejas, C. Belda-Iniesta, and M. González-Barón, "PI3K/Akt signalling pathway and cancer.," *Cancer treatment reviews*, vol. 30, pp. 193–204, 2004.

[35] Kanehisa Laboratories, "KEGG Markup Language." http://www.kegg.jp/kegg/xml/docs/. Accessed: 29.01.2014.

[36] M. Streit, *Metabolic Pathways Influencing Gene-Expression Analysis*. Master's thesis, Graz University of Technology, 2007.

[37] Donnelly Centre at the University of Toronto., "Cytoscape.js." http://cytoscape.github.io/cytoscape.js/. Accessed: 20.10.2013.

[38] P. Du, G. Feng, W. Kibbe, and S. Lin, *lumiHumanIDMapping: Illumina Identifier mapping for Human*.

[39] Kanehisa Laboratories, "KEGG - Current Statistics." http://www.kegg.jp/kegg/docs/statistics.html. Accessed: 29.01.2014.

[40] Institute for Computer Graphics and Vision TU Graz, "Caleydo - Visualization for Molecular Biology —." http://www.icg.tugraz.at/project/caleydo/. Accessed: 12/02/14.

[41] A. Lex, M. Streit, H.-J. Schulz, C. Partl, D. Schmalstieg, P. Park, and N. Gehlenborg, "StratomeX: Visual Analysis of Large-Scale Heterogeneous Genomics Data for Cancer Subtype Characterization," *Computer Graphics Forum*, vol. 31, pp. 1175–1184, 2012.

[42] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit, "LineUp: visual analysis of multi-attribute rankings.," *IEEE transactions on visualization and computer graphics*, vol. 19, pp. 2277–86, 2013.

[43] Institute for Computer Graphics and Vision TU Graz, "Caleydo 3.1.0 Beta Release Notes." http://www.icg.tugraz.at/project/caleydo/caleydo-3-1-0-release-notes. Accessed: 12/02/14.

[44] C. Collins, G. Penn, and S. Carpendale, "Bubble sets: revealing set relations with isocontours over existing visualizations.," *IEEE transactions on visualization and computer graphics*, vol. 15, pp. 1009–16, Jan. 2009.

[45] Z. Hu, E. S. Snitkin, and C. DeLisi, "VisANT: an integrative framework for networks in systems biology.," *Briefings in bioinformatics*, vol. 9, pp. 317–325, 2008.

[46] J. Gómez, L. J. García, G. A. Salazar, J. Villaveces, S. Gore, A. García, M. J. Martín, G. Launay, R. Alcántara, N. Del-Toro, M. Dumousseau, S. Orchard, S. Velankar, H. Hermjakob, C. Zong, P. Ping, M. Corpas, and R. C. Jiménez, "BioJS: an open source JavaScript framework for

biological data visualization.," *Bioinformatics (Oxford, England)*, vol. 29, pp. 1103–4, 2013.

[47] H. Rohn, A. Junker, A. Hartmann, E. Grafahrend-Belau, H. Treutler, M. Klapperstück, T. Czauderna, C. Klukas, and F. Schreiber, "VANTED v2: a framework for systems biology applications.," *BMC systems biology*, vol. 6, p. 139, 2012.

[48] E. LIMPERT, W. A. STAHEL, and M. ABBT, "Log-normal Distributions across the Sciences: Keys and Clues," 2001.

[49] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer, "Wrangler: interactive visual specification of data transformation scripts," in *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, (New York, New York, USA), p. 3363, ACM Press, May 2011.

A Source Code

The source code follows on a CD-ROM and can also be found at github.com/fjukstad/kvik.