Development of Parallel Coordinates Using D3 for Hybrid Reality Environments:

Visualising Acute Lymphoblastic Leukaemia Data

Author: Samantha Crome

Date of Submission: 11 December 2015

Keywords: Parallel Coordinates, Cancer, Biomedical Data

Abstract

Visualisation is the process of creating a visual representation of data. Visualisation is a tool that assists the analysis of data, the finding of patterns and meaning, the exploration of relationships and the development of hypotheses. Increasing amounts of medical data are being captured systematically as part of routine clinical care. In addition, medical research is identifying new biomarkers of disease diagnosis, prognosis and response to treatment such as gene expression profiles. Visualisation of biomedical data can assist in finding patterns in a particular disease to optimise treatment and prognosis. This is important, for example, for cancer patients, in which the opportunity of personalised treatment can make a large difference in the probability of a cure or remission. The primary objective of this research is to develop code to extend the 2D visualisation functionality of a newly built Data Arena (a CAVE-like Hybrid Reality Environment) by enabling a parallel co-ordinates data visualisation tool. This data visualisation tool will be used to evaluate multivariate high dimensional biomedical data of children with acute lymphoblastic leukaemia, with the aim of developing better predictive risk stratification models to guide therapeutic decisions, which in turn, may reduce comorbidity and improve survival.

Visualisation is an established exploratory tool for use with multivariate high dimensional data. It could be used in computer generated virtual environments. Virtual environments are three-dimensional computer generated environments that simulate sensory information. A recent technology is Hybrid Reality Environments (HREs) which combine the immersion of CAVE with the high resolution of large ultra-resolution displays. HREs are promising for the analysis of complex data. The University of Technology Sydney has built a CAVE-like HRE called the Data Arena. There are many visualisation libraries that could be integrated into the Data Arena for data visualisation tools. D3, a JavaScript library, has several components which improve interoperability, documentation, expressiveness, compatibility, and performance. D3 can also create Scalable Vector Graphics (SVG) images, which is useful when performing interactions of the images and rendering. Of the many multivariate data visualisations tools that could be used to explore the relationships within the acute lymphoblastic leukaemia data using D3 in the Data Arena, parallel coordinates has been shown to be useful because of its ability to find patterns and exceptions.

There are two aims for this thesis. The first aim is to demonstrate that parallel coordinates visualisation of a paediatric acute lymphoblastic leukaemia dataset can be implemented in the Data Arena. A secondary, minor aim is to be able to implement other 2D multidimensional

visualisations in the Data Arena. Through iterative development and evaluation two potential solutions emerged. The first solution is to use the WebView Omegalib module and use it to view a webpage containing a D3 parallel coordinates visualisation. Omegalib is the application framework used in the Data Arena. This solution has the benefit of having easy interaction, but is heavily dependent on the progress of the module for increase in quality. The second proposed solution is a work in progress proof of concept. It uses an XML parser to parse an SVG created using the D3 library that is first extracted by the user for a webpage using D3. By analysing the parsed elements, the elements are translated into an Omegalib Cyclops module objects.

An evaluation of these two solutions demonstrates that the first aim of the thesis is successful, however the second aim is not fulfilled. The parallel coordinates visualisations of acute lymphoblastic leukaemia data perform well in the Data Arena passing all test cases. The WebView solution succeeds in adapting some other types of 2D visualisations from external sources, but fails to visualise others. Additionally, as the SVG Parser solution is in the proof of concept stage there is still development needed to make it truly adaptable, but the potential is there. Overall, the thesis benefited the acute lymphoblastic leukaemia project by demonstrating that acute lymphoblastic leukaemia data can be visualised using parallel coordinates in the Data Arena with high quality and interactivity. Additionally, it assisted Data Arena developers because the process of using Omegalib and communicating with the developers gave feedback and identified some difficulties with Omegalib.

TABLE OF CONTENTS

Table of Figures

Table of Tables

1. Introduction

1.1 Background

In 1974 John Tukey wrote “The main tasks of pictures are … to reveal the unexpected, to make the complex easier to perceive.” (Tukey 1974, p. 525). The Merriam-Webster (n.d.) gives three definitions for visualisation: the creation of mental images, the process of interpreting in visual terms or putting into visible form, and a medical process that makes an internal organ or part visible by the introduction of radioactive isotopes followed by radiography. The definition of visualisation in information technology mainly follows the second definition. For this thesis the definition of visualisation is the second definition applied to the visualisation of data. Data visualisation supports exploration, communication, and examination of data (Few 2009; Tufte 2002). Visualisations do this by representing relationships as trends, patterns, and exceptions. It is often easier for people to understand or communicate complex data relationships through pictures or graphical representations than being provided with large amounts of numerical data (Williams 2011). As an example, it is easier to find an outlier on a scatterplot graph than it is to find in a table of numbers.

The concept of virtual environments, three-dimensional computer generated environments that simulate sensory information, was first described in 1965 (Cruz-Neira, Sandin & DeFanti 1993). One virtual environment system developed in 1991 by the Electronic Visualization Laboratory at the University of Illinios Chicago was the CAVE Automatic Virtual Environment (CAVE) (Cruz-Neira, Sandin & DeFanti 1993). CAVE (the term CAVE is a recursive acronym) was first developed as a scientific data visualisation system. CAVE comprises a room or space that projects the visuals to the walls of that space. CAVE systems use screens, surround audio, tracking systems, and 3D capabilities to create an immersive virtual space (Manjrekar et al. 2014).

There have been a number of CAVE-like systems built over the last two decades as well as associated application programming interfaces (APIs), libraries, and software. Although CAVE systems are beneficial in areas that require immersion or presence, there is little evidence that CAVE provides an advantage in complex data analysis over other graphical applications (Bowman & McMahan 2007). Hybrid Reality Environments (HRE) were developed by the Electronic Visualisation Laboratory in 2013. HREs combine virtual environments like CAVE with large ultra-high resolution display environments (Manjrekar et al. 2014). Ultra-high resolution display environments are large high-resolutions displays that are usually made from tiled LCD monitors to form a single adjoining surface. HREs provide both interactive and immersive 2D and 3D data visualisation capabilities (Reda et al. 2013). HREs have only been recently developed. One software framework that allows HREs ability to have multiple applications running on a cluster-controlled display system is Omegalib (Febretti et al. 2014). Although on face value HREs are a promising advance in data visualisation tools, their benefits and limitations require evaluation given the resources required for their construction and use.

Coding libraries have been developed by programmers to assist the development of applications. There are a number of different libraries that one could use in HRE to do 2D visualisation. One such library is Data Driven Documents, better known by its acronym D3. D3 is an open source JavaScript library that inspects and transforms the standard document object model (DOM) to make visualisations (Bostock, Ogievestky & Heer 2011). D3 uses native representation, which improves interoperability, documentation, expressiveness, and compatibility with other web languages such as CSS. D3 avoids unnecessary computation, thus improving performance, by using explicit transformations of a native representation. D3 can create Scalable Vector Graphics (SVG) images, which is useful when performing interactions of the images and rendering. D3 is able to be inspected and analysed using browser developer tools, such as Google Chrome’s developer tools. This assists in the debugging, analysis, and experimental modification of D3 visualisations and code. D3 does not generate predefined visualisations; functionality is extended by available plugins (Wang et al. 2015). This is useful as once base integration is achieved plugins easily provide additional functionality. Additionally, other libraries have been built on D3, which also add functionality (Wang et al. 2015). D3 has already been used in many bioinformatics applications (Gomez et al. 2013; Wang et al. 2015).

Parallel coordinates are a series of parallel axes where each observation in the data is represented by a line (Inselberg & Dimsdale 1990; Siirtola et al. 2009). Each axis in the parallel coordinates visualisation represents a feature. As parallel coordinates can view multiple dimensions simultaneously, it is a useful tool for multidimensional data sets (Few 2009). Parallel coordinates are used to their greatest advantage when they are interactive (Few 2009). Interactive elements such as filtering, brushing, similarity ranking, and similarity clustering provide greater analysis with parallel coordinates. Each of these methods may reveal different insights about the data, and most of these methods can be used in other visualisations in addition to parallel coordinates. There have been many applications for parallel coordinates, including has been use in air traffic control, robotics, statistics, computational geometry, instrumentation, computer vision, digital forensics, geographic information systems, social networking, earth sciences data, and bioinformatics (Cheng et al. 2008; Inselberg & Dimsdale 1990; Moustafa 2011; Tricaud, Nance & Saadé 2011).

Acute lymphoblastic leukaemia is the most common form of childhood cancer (Cancer Australia 2014; Leukaemia Foundation 2015). Intensive treatment with chemotherapy and radiotherapy achieves a cure rate of approximately 90% (Inaba, Greaves & Mullighan 2013). However, because treatment is associated with toxicity, accurate risk stratification is needed to optimise therapy for individuals (Pui et al. 2012). A number of risk profiles have been identified and these profiles can be augmented with gene expression data which may distinguish heterogeneous acute lymphoblastic leukaemia subtypes and facilitate personalised treatment (Inaba, Greaves & Mullighan 2013; Pui et al. 2012). However, gene expression data are often of a very high-dimensional, and cause difficulty with data analysis and data interpretation using traditional methods (Catchpoole et al. 2010; Ioannidis 2005).

1.2 New Contribution

The University of Technology Sydney (2014) has built a CAVE-like HRE called the Data Arena. The hardware components of the Data Arena are more CAVE-like, whereas the purpose and software of the Data Arena are more aligned with HRE. Omegalib is the software used in the Data Arena. The main purpose of the Data Arena is to facilitate research and education by presenting and interacting with 3D spatial models and complex data. The University of Technology Sydney has also been a collaborative institution on the paediatric acute lymphoblastic leukaemia project for a number of years along with the University of Western Sydney, Queen's University Canada and the Children's Hospital at Westmead. The project includes the evaluation of high dimensional clinical and biological information, including genetic data, with the aim of improving therapeutic decisions and clinical outcomes of children with acute lymphoblastic leukaemia. Parallel coordinates, a useful way of visualising this high-dimensional information, has not been used to evaluate the acute lymphoblastic leukaemia dataset. However, parallel coordinates are limited by aspect ratio and screen resolution (Moustafa 2011). This restricts the number of dimensions that can be used to view the acute lymphoblastic leukaemia data using parallel coordinates on a personal computer small display. By increasing the screen resolutions and display size, as can be achieved in a Data Arena, more dimensions can be viewed, and this limitation is mitigated. Fortuitously, some members of the acute lymphoblastic leukaemia project already have experience in D3 (Tran, Nguyen & Simoff 2014). This experience will be an advantage when implementing the D3 library in the new Data Arena. In developing parallel coordinates with D3 for use in the Data Arena we are in a position to analyse the multiple dimensions of the acute lymphoblastic leukaemia data set in a way that has not been previously possible.

1.3 Stakeholders

The stakeholders of this paper include members of the acute lymphoblastic leukaemia project as well as other University of Technology Sydney research staff, research students and visiting researchers who may use the Data Arena to apply visual analytic tools to their research data.

1.4 Aims

There are two aims, one major aim and a secondary minor aim. The major aim is to show that parallel coordinates of complex data, such as is found in paediatric acute lymphoblastic leukaemia dataset, can be implemented in the Data Arena. The secondary minor aim is to be able to have other types of 2D multidimensional plots available for use in the Data Arena.

1.5 Objectives

The first objective of our research project is to develop code to extend the 2D visualisation functionality of the Data Arena by enabling parallel coordinates data visualisation methods. The second objective is to ensure that this new code is adaptable to other visualisation methods.

1.7 Significance

Visualisation can facilitate analysis by discerning meaning in data relationships which, in turn, can inform better decisions. By developing code to extend the 2D visualisation functionality of the Data Arena, we will be able to provide parallel coordinate visualisation of the paediatric acute lymphoblastic leukaemia data set. If we demonstrate that parallel coordinates viewed in the Data Arena is able to view parallel coordinates plots that were built for viewing on a personal computer, then the Data Arena constitutes a new method of viewing the plots. By viewing parallel coordinate plots in a large, ultra-high resolution display environment such as that provided by the Data Arena, there is an anticipation that this new immersive and interactive human-computer interface tool, will synergise human cognitive processes with human-human exchanges to provide insightful and useful discoveries. In the specific context of the multi-dimensional acute lymphoblastic leukaemia data, it is hoped that this new tool will benefit clinical and statistical researchers by providing insightful exploration, analysis and interpretation of the acute lymphoblastic leukaemia data for individuals and more importantly, for interactive groups of researchers. Plots may be more easily built and shared among researchers. If this new tool generates new discoveries in paediatric acute lymphoblastic leukaemia pathogenesis and refinement of paediatric acute lymphoblastic leukaemia risk profiles, this may contribute towards new treatments as well as personalised treatment.

Furthermore, developing code that implements parallel coordinates using existing libraries provides an opportunity for easy implementation of other 2D visualisations in the Data Arena. This expands the use of the use of the Data Arena to other visual analytic methods for other users. As the main barriers to HREs are cost and resource management, these barriers are acceptable if the Data Arena as a research tool provides multiple visual analytics functionality for the use of many clinical and statistical researchers with many different datasets. If the Data Arena is shown to have a significant impact on research innovation and progress, barriers of cost and resource management may be diminished through the generation of new revenue from successful project grant funding. Finally, visualisation within the Data Arena also provides a means of effectively communicating methods, discoveries and other information with administrators, funders and consumers.

1.8 Outline of Research Methods

Two methods are used to address the objectives outlined in Section 1.5 Objectives. The first method is the development methodology used during the development of the code. This method is based on iterative development and evaluation of features. This development will be conducted on a personal computer (PC) as the Omegalib environment can be installed on a PC to develop applications for an HRE running Omegalib (Febretti et al. 2014). The second method is the evaluation methodology. This method uses a test plan with test cases. Code that is considered to be successfully developed on the PC through the iterative development process outlined above will be evaluated with test cases on the Data Arena.

1.9 Organisation of Content

The dissertation is organised as follows.

Section 1, Introduction, begins with a brief background of the opportunities for knowledge acquisition with visualisation tools, virtual and hybrid reality environments and visualisation libraries. This is followed by a brief description of the advantages of a specific visualisation tool, parallel coordinate plots, in the context of high-dimensional data, such as that associated with paediatric acute lymphoblastic leukaemia, and the potential for these visualisation tools to provide insights which may lead to clinical benefit and facilitate personalised treatment. The introduction continues with the new contribution to research, identification of stakeholders, presentation of the aims and objectives, a discussion of the significance of the dissertation, and an outline of the research methods.

Section 2, Literature Review, presents a detailed description and discussion of the acute lymphoblastic leukaemia, high dimensional low observation data, data visualisation, CAVE automatic virtual environment, Hybrid Reality Environments, visualisation libraries and parallel coordinates. The literature review ends with identifying some of the research gaps in demonstrating the value of Hybrid Reality Environments as a mean of knowledge acquisition in general, and the University of Technology Sydney's CAVE-like HRE Data Arena using D3 libraries, specifically.

Section 3, Methods, provides a detailed description and rationale of the methods used to achieve the research objectives. As outlined in Section 1.8, two principal methodologies were applied: (i) development methodology (development phase) and (ii) evaluation methodology (evaluation phase). Only solutions that were implemented successfully on the PC environment in the development phase were carried forward for evaluation on the Data Arena test cases. The test plan is described in detail. This section includes information on Omegalib installation, the required libraries and all solutions that were considered in development phase. The section also provides information on solutions that were not carried forward to the evaluation phase because they were (a) either discontinued early because of feasibility concerns, or (b) discontinued after development because of early failure in the PC environment.

Section 4, Results, reports the results of two solutions that were carried through to the evaluation phase. Parallel coordinate plots were evaluated for both solutions. However, only one solution evaluated other types of 2D multidimensional plots. The other solution showed early promise but was not evaluated on other 2D multidimensional plots because of time constraints, therefore remains a work in progress proof of concept. This section also includes test case comments.

Section 5, Conclusion, summarises the background, aims and results of this research, provides concluding remarks and discusses potential future research.

Section 6, References, lists the references in first author alphabetic order.

Section 7, Appendix, contains (1) source codes, and (2) proposes an example of an empirical method to test whether parallel co-ordinates implemented in a Data Arena offers any advantage over parallel coordinates implemented on a PC. This proposal uses a factorial research design.

2. Literature Review

The literature review describes a detailed description and evaluation of the literature. It provides background on acute lymphoblastic leukaemia, high dimensional low observation data, data visualisation, CAVE automatic virtual environment, Hybrid Reality Environments, visualisation libraries and parallel coordinates. The section concludes by identifying the gaps in the literature.

2.1 Acute Lymphoblastic Leukaemia

Acute lymphoblastic leukaemia is a cancer of the blood. Immature white blood cancer cells (lymphoblast cancer cells) are overproduced, and because they are immature these white cells cannot fight bacterial and other infections. Furthermore, the overproduction of lymphoblasts in the bone marrow prevents the production of normal red blood cells and platelets which causes anaemia and bleeding (Leukaemia Foundation 2015). Acute lymphoblastic leukaemia is the most common form of childhood cancer (Cancer Australia 2014; Leukaemia Foundation 2015), although overall it is a rare cancer, accounting for only 0.3% of new cases in Australia in 2010 (approximately 300 new cases per year). Approximately 60% of acute lymphoblastic leukaemia cases are children under 14 years of age (Leukaemia Foundation 2015). Survival of childhood acute lymphoblastic leukaemia is approximately 90% (Inaba, Greaves & Mullighan 2013). Acute lymphoblastic leukaemia is treatment is based on the risk of relapse. Intensive treatment with chemotherapy and radiotherapy should be limited to mainly high-risk individuals because treatment toxicity causes significant morbidity and death (Pui et al. 2012). Therefore low and high risk profiles have been developed informed by clinical and biological data such as age of onset, initial response to induction treatment, number of cancer lymphoblasts in the peripheral blood, immunophenotyping of the lymphoblast to determine whether it is a B or T cell lineage lymphoblast, cytogenetics of the lymphoblast (such as the number of chromosomes and whether there are chromosomal translocations), as well as the level of minimal residual disease (MRD) after remission induction (Inaba, Greaves & Mullighan 2013; Pui et al. 2012). However, there is considerable heterogeneity within subtypes of acute lymphoblastic leukaemia and prognostic significance of even established indicators, such as MRD, have not always been demonstrated in new clinical trials (Pui et al. 2012). The hope is that the cure rate in paediatric acute lymphoblastic leukaemia can be increased with research to discover new cancer lymphoblast biological abnormalities to reveal this heterogeneity, which can be used to develop novel targeted treatments and novel prognostic tools to achieve the goal of personalized medicine.

There are many clinical and biological research groups working towards improving the diagnosis and classification of risk strata for treatment planning in the clinical setting. Genome profiling has identified new acute lymphoblastic leukaemia subtypes through identifying sequence mutations and sub-microscopic structural genetic changes (Inaba, Greaves & Mullighan 2013). Morphological identification of lymphoblast, immunophenotypic assessment of lineage commitment and developmental stage, chromosomal analysis, flow cytometry, reverse transcriptase polymerase chain reaction (PCR), and either multiplex ligation-dependant probe amplification or flurescence in-situ hybridisation are used for diagnosis and prognosis (Inaba, Greaves & Mullighan 2013). Demographic and clinical factors such as age, race, gender and comorbidity have an impact on the acute lymphoblastic leukaemia risk strata (Inaba, Greaves & Mullighan 2013). Early response to treatment is also predictive of risk of relapse and assignment to a risk-adapted treatment based on initial response is included in treatment protocols.

High-throughput gene expression profiling is one means of identifying and exploring diagnostic, prognostic and therapeutic heterogeneous subtypes of acute lymphoblastic leukaemia (Pui et al. 2012). Early innovators such as Golub et al. (1999) developed a general systematic framework for a class discovery and prediction using statistical methods of classification and clustering. They applied their approach to gene expression DNA microarray data and their model was able classify new cases of acute lymphoblastic leukaemia and acute myeloid leukaemia with some degree of success. Yeoh et al. (2002) used gene expression and statistical methods such as support vector machines, correlation-based feature selection, and decision trees to classify sub-groups of acute lymphoblastic leukaemia. Ross et al. (2003) continued this work by increasing the number of genes analysed to find novel sub-types and to work towards a standardised diagnostic platform. Díaz-Uríarte & de Andrés (2006) used random forests for gene selection and classification of microarray data. They found that this method was equivalent with other methods of the time and recommended it should be used as part of a method tool set. However, they did not address the problem of imbalanced classes (Anaissi et al. 2013). Anaissi et al. (2013), working with data from the Children's Hospital at Westmead as well as two NCI60 data sets, proposed balanced iterative random forests (BIRF) to address this problem. This method performed well against random forests and Multiple Machine-Recursive Feature Elimination. It was comparable or superior to other methods used on the NCI60 sets. However there were a few limitations. BIRF is not able to produce global correlation when splitting the data set, but this can be avoided by not splitting the data set. Also tuning of the parameter that handles the balanced classes is a limitation.

One aim of the UTS acute lymphoblastic leukaemia project is to develop better data visualisation (see Section 2.3) of the acute lymphoblastic leukaemia data and better presentation of the data to clinicians and researchers. Nguyen et al. (2011) developed a 3D visualisation and analysis approach and tool for clinical and genetic data sets of acute lymphoblastic leukaemia. Expression and single nucleotide polymorphisms were used to create a 3D similarity of patients, which was then displayed as an interactive visualisation. However, this development is based on Java3D, which is considered outmoded compared to new platforms such as the Unity3D platform (Khalifa et al. 2015). Khalifa et al. (2015) created a 3D scatterplot visualisation system for analysing genomic and biomedical acute lymphoblastic leukaemia data using the Unity3D platform. This was not developed for use in the UTS Data Arena. The new Data Arena provides the opportunity to explore many data visualisation methods using genomic and biomedical acute lymphoblastic leukaemia data.

2.2 High-Dimensional Low Observation Data

Dimensions in data refer to the number of variables and the number of records in a data set (Williams 2011). The acute lymphoblastic leukaemia data includes data from gene expression which has been shown in Section 2.2 to be an important tool in identifying and predicting the outcome of heterogeneous subtypes of acute lymphoblastic leukaemia. With the use of modern technology gene expression microarray data is high-throughput data (Kallioniemi, Kononen & Sauter 2012; Kallioniemi, Wagner, Koneneon & Sauter 2001). However, one of the limits of this new technology is how to analyse the vast amounts of data that is generated. Data analysis and interpretation remains one of the bottle-necks (Hoheisel 2006). It has been described as the ‘curse of dimensionality’ because there are many more measureable attributes in microarray data than there are samples (Catchpoole et al. 2010). This high-dimensional data low observations data means there are many more variables (e.g. gene expression products) than there are records (e.g. patients, observations), and this causes problems using traditional research analytical methods as traditional statistical inference generally requires more observations than variables in order to reduce false positive results (Ioannidis 2005). Paediatric acute lymphoblastic leukaemia is relatively rare so access to more observations remains difficult (Biondi & Cazzaniga 2013). The statistical aspects of prognostic factor studiers in cancer research were outlined by Simon and Altman in 1994.

They raised issues including the challenging problem of sample size. The problem of high-dimensional data low observations data requires new methods of analysis and new approaches which include data visualisation and data mining techniques (Kennedy et al. 2008).

2.3 Data Visualisation

In 1974 John Tukey wrote "The main tasks of pictures are … to reveal the unexpected, to make the complex easier to perceive." (Tukey 1974, p. 525). Data visualisation is visual representation that supports exploration, communication, and examination of data (Few 2009). Data visualisation can be split into two broad categories: information visualisation and scientific visualisation. Information visualisation, which is the category that this thesis fits under, is the visual representation of abstract data. Scientific visualisation tends to be the visual representation of the physical, such as a scan of a body part (Few 2009). Visualisations such as images and graphs, should support thinking and assist in the analysis of data. Visualisations are an effective way of describing, exploring and summarising quantitative data (Tufte 2002). The goal of visual analysis is to discern the meaning of the relationships in the data which can inform better decisions (Few 2009). Visualisations do this by representing relationships as trends, patterns, and exceptions. Tufte (2002) explains that graphics are supposed to show the data, encourage comparison of data, reveal several levels of detail from overview to fine, serve a clear purpose, focus the viewer on the substance rather than how it was made, make large data sets coherent, and be integrated with verbal description and statistical analysis. Well-designed visualisations are both powerful and simple. Visual analysis can reach a broader audience than advanced statistical analysis, which requires specialist training (Few 2009). However, visual analysis is only as good as the data, model, or theory from which it is created (Few 2009; Tufte 2002).

High-dimensionality is not inherently an issue. Multivariate data has detail and complexity that may add richer meaning (Few 2009). It is how this data is used and analysed that is an issue because it is often difficult for traditional tools. Visual analysis can help find insight even if that insight raises more questions (Few 2009). Visual analysis in virtual environments and HRE may assist in visual analysis of high-dimensional data sets.

2.4 CAVE Automatic Virtual Environment

Virtual environments are computer-generated 3D environments, which allow users to interact with the generated environment by using synthetic stimuli encompassing vision, sound, touch, and smell (Bowman & McMahan 2007; Cruz-Neira, Sandin & DeFanti 1993; Nan et al. 2014). The main focus of development of virtual environments has been vision. The mainstream term for virtual environments is virtual reality, with this term being common in the literature (Cruz-Neira, Sandin & DeFanti 1993). The term virtual environment is more accurate and is part of the CAVE acronym and will be used throughout the paper. The components to create a virtual environment include field of view, interactivity, stereoscopy, head motion and rendering, perspective projection, occlusion, convergence, accommodation, atmospheric, and lighting and shadows, frame rate, refresh rate, and display resolution (Bowman & McMahan 2007; Cruz-Neira, Sandin & DeFanti 1993).

During and prior to the inception of CAVE there were existing systems that were within the realm of virtual environments at least to an extent. Cruz-Neira, Sandin and DeFanti (1993) describe many of the other technologies. Omnimax theatres could provide a large field of view with stereo. Head-tracked monitors provided many of the components except field of view. Head-mounted displays and BOOMs implemented virtual realities by using the motion of the display to mimic head motion and rendering. Both CAVE and another projection based virtual environment called Virtual Portal, premiered at SIGGRAPH ’92, however used a different implementation. The Virtual Portal was smaller than the CAVE and had a 3 wall projection system, whereas the CAVE was larger and had a 3 wall and floor projection system.

According to Nan et al. (2014) virtual environment systems can be split into three main categories based on the level of immersion. Non-immersive environments use 2D monitors to display a 3D environment. Semi-immersive environments, such as head mounted displays, support looking at the environment. Fully immersive environments, such as CAVE, support the feeling of presence.

CAVE was initially developed as a virtual environment. Electronic Visualization Laboratory at University of Illinois Chicago designed and developed the first CAVE in 1991 (Cruz-Neira, Sandin & DeFanti 1993). The two main characteristics of CAVE are immersion and presence. Immersion is an objective and quantifiable variable, which measures sensory fidelity. As an example, visual immersion consists of, but is not limited to, field of view, field of regard, display size, display resolution, realism of lighting, frame rate, refresh rate, stereoscopy, and head-based rendering (Bowman & McMahan 2007). Presence is the psychological response of users, the sense of being in reality replicated by a virtual environment (Bowman & McMahan 2007). Different users can feel different levels of presence for the same system.

The CAVE systems can generally be split into the following components: audio system, video system, CAVE calibration, stereoscopic rendering, tracking systems, graphics engine, and user input devices (Manjrekar et al. 2014; Nan et al. 2014). Audio systems attempt to provide spatial audio for the provided space (Manjrekar et al. 2014). Video systems provide projected or LCD/LED screen 3D visuals. CAVE calibration software ensures that the multiple projectors or screens present a seamless visual interface. Using shutter glasses provides stereoscopic rendering, which adds depth information. There are three types of tracking systems: camera/vision tracking algorithm that track markers on the user's head, magnetic tracking system, and optical tracking from multiple viewpoints with markers (Manjrekar et al. 2014). Specialised graphics engines allow the rendering of 3D and 2D graphics over the CAVE platform (Febretti et al. 2014; Manjrekar et al. 2014). User input devices for CAVE systems are varied. Some user input devices include wands, hand gestures and markers, touch input on 2D devices such as tablets, keyboard, and mouse (Febretti et al. 2014; Nan et al. 2014). As technology progresses more solutions become available for each of the components of CAVE, improving quality, providing flexibility and opportunities, or decreasing cost (Manjrekar et al. 2014).

Bowman and McMahan (2007) describe two experiments from their own laboratory that investigated specific components of virtual environments. They found that components of wide field of view and high resolution decreased search time and facilitated navigation when searching for specific spatial or abstract information or when comparing between pieces of information. In contrast they report that components of stereoscopy, field of regard, and head-based rendering did not improve many tasks involving surface plots and 3D scatterplots.

As the technology of CAVE improves and develops, the number of applications have increased, often providing a unique approach to the task. The original motivation of the development of CAVE was to provide a tool for scientific visualisation (Cruz-Neira, Sandin & DeFanti 1993). Currently, the purpose of CAVE has expanded to entertainment, simulation, design, planning, treatment, art, presentation, and education (Bowman & McMahan 2007; Creagh 2003; Cruz-Neira et al. 1993; Dell'Unto et al. 2013; Manjrekar et al. 2014; Nan et al. 2014; Yang et al. 2011). Due to CAVE allowing multiple participants to be present in the space, it allows people to work together. This allows benefits such as team analysis (Yang et al. 2011) or simultaneous evaluation by differing professions (Creagh 2003).

The use of CAVE is multidisciplinary including engineering, archaeology, architecture, manufacturing, medicine, military, chemistry, and biology (Bowman & McMahan 2007; Creagh 2003; Cruz-Neira, et al.1993; Dell'Unto et al 2013; Manjrejar et al. 2014; Nan et al. 2014; Yang et al. 2011). The literature seems to comprise models, spatial uses, and simulation. There is little evidence available that demonstrates the advantages of analysing complex data in CAVE over other graphical applications.

Two of the main identifiable benefits of CAVE are its two main characteristics: immersion and presence. High levels of immersion benefited a project by Akkiraju et al. (1996) in viewing and analysing genomic structures created by different models. By using the CAVE system they were able to recognise the seriousness of a shortcoming in one of the models. Virtual reality therapy uses presence to help manage and treat phobias by allowing users to experience and manage the phobia in a contained and controlled environment (Bowman & McMahan 2007; Creagh 2003). Presence has also been a benefit in archaeological research when using and comparing different models and simulations of buildings in Pompeii. It allowed researchers to gain a new perspective on the probable architecture of a ruined ancient building (Dell'Unto et al. 2013).

The main barriers to CAVE are cost and time-allocation. High cost of complete immersion CAVE has been one of the main contentions when compared to lower cost, high-end graphics solutions (Akkiraju et al. 1996). High cost has been a shortcoming of CAVE since its initiation (Akkiraju et al. 1996; Cruz-Neira, Sandin & DeFanti 1993; Manjrekar et al. 2014). However, CAVE allows researchers to view information and incite discussion that is a necessity or a benefit to some projects (Akkiraju et al. 1996; Dell'Unto et al. 2013). Evaluating the needs of the project against the characteristics of immersion that CAVE provides means that the solution can be tailored to the users (Bowman & McMahan 2007). If head tracking does not provide an improvement in performance or efficiency of research then that cost can be moved to a characteristic the project needs, such as field of view.

Manjrekar et al. (2014) states that CAVE can be separated into three categories based on who makes the CAVE, its quality, and its cost. First are high quality and costly professional solutions that are sold by specialists in virtual environment equipment. Second, homemade solutions that are lower in quality but are very inexpensive and made with common materials. Third, low cost solutions that are intermediary architectures that aim for the quality of professional solutions while being lower in cost. Low-cost solutions are usually lower in quality than professional solutions. The various solutions mean that virtual environments can be tailored to the needs of the team or workplace.

CAVE is a highly contended resource as there is limited availability (Frebetti et al. 2014). This causes difficulty in scheduling and making the resource available for research. This is more difficult to solve because the majority of CAVE software and applications make it difficult to share the resource simultaneously (Frebetti et al. 2014). However, as technology improves and prices decrease, there could be more CAVEs available, lessening the resourcing issue.

2.5 Hybrid Reality Environments

HRE is a relatively new immersive environment that merges virtual environment systems with large ultra-high resolution display environments (Manjrekar et al. 2014). The same laboratory that created the first CAVE has developed HRE's such as Cyber-Commons, not CAVE-like, and CAVE2 which is CAVE-like. The purpose of HRE is to assist in constructing scalable, interactive visualisations that combine a variety of data sources, including 2D, 3D, temporal and multivariate (Reda et al. 2013). Reda et al. (2013) describe HRE as having 5 characteristics: a large high-resolution display, support for stereoscopic depth, support for naturalistic interactions, collocated collaboration space, and software that can simultaneously display multiple datasets and utilise 2D/3D hybrid visualisation and interaction. HREs allow multiple forms of analysis within one technology, increasing applications and allowing comparison between data sets that would have to be displayed on two devices. There is little available literature on the success cases of HREs, possibly because it has only been developed in the last two years. More published literature is anticipated in the near future. CAVE-like HREs have been used for visualisation in nanoscale-materials science, and it has been used for a two day meeting of the ENDURANCE project by NASA (Febretti et al. 2014; Reda et al. 2013). Similar to high quality virtual environments and large-scale display environments, HREs are expensive (Frebetti et al. 2014). Another issue is that HREs are a highly contended resource. However, there has been work into creating frameworks and libraries to allow users and developers to have multiple workspaces on the HRE simultaneously (Frebetti et al. 2014). Future work for HREs will include characterising the design space to provide guidelines and finding applications for HRE technology and software (Reda et al. 2013). As development in HREs are relatively recent compare to two decades of research into CAVEs, much work needs to be done to find the limits, empirical advantages, and empirical disadvantages compared to other large display systems.

Omegalib is a software framework that provides the ability to have multiple applications running on a cluster-controlled display system and assists application development on HREs (Febretti e al. 2014). It acts as an abstraction layer between the applications and the HRE hardware. It is used in the Data Arena and will be the software to which we will connect the visualisation library. The Electronic Visualisation Laboratory in the University of Chicago Illinois, the same laboratory that created the first CAVE, created Omegalib. The Omegalib API is exposed through a C++ and Python interface which can be used for the runtime console or for script applications (Febretti et al. 2014). Third party libraries can be simply integrated through Omegalib's support of pluggable front-ends. Additionally development can be done on PCs because script applications written for Omegalib can run on PCs or HRE without reconfiguration or recompilation. Because this is a new software framework, there is still a lot of opportunity to integrate libraries to increase functionality.

2.6 Visualisation Libraries

Coding libraries have been developed by programmers to assist the development of applications. There are a number of different libraries that one could use in the Data Arena to do 2D visualisation. One such library is Data Drive Documents, better known by its acronym D3. D3 is an open source JavaScript library that inspects and transforms the standard document object model (DOM) to make visualisations (Bostock, Ogievestky & Heer 2011). D3 has three main goals: compatibility, performance, and debugging. D3 uses native representation, which improves interoperability, documentation, expressiveness, and compatibility with other web languages such as CSS. D3 avoids unnecessary computation, thus improving performance, by using explicit transformations of a native representation. D3 can create SVG images, which is useful when performing interactions of the images and rendering. D3 is able to be inspected and analysed using browser developer tools, such as Google Chrome's developer tools. This assists in the debugging, analysis, and experimental modification of D3 visualisations and code. D3 does not generate predefined visualisations; however, functionality is extended by available plugins (Wang et al. 2015). This is useful because once base integration is achieved plugins easily provide additional functionality. Additionally, other libraries have been built on D3 which also add functionality (Wang et al. 2015). D3 has already been used in many bioinformatics applications. BioJS allows the use of the D3 library for its components (Gomez et al. 2013). This has been utilised by the components DNAContentViewer and wigExplorer (Wang et al. 2015). D3 has also been used in proteomics data analysis, genomic sequence annotation and classification, and a web-based genome browser (Wang et al. 2015).

These features of D3 provide a rationale for its use in developing 2D visualisations for the Data Arena. Moreover, some members of the acute lymphoblastic leukaemia project already have experience in D3 (Tran, Nguyen & Simoff 2014). This experience will be an advantage when implementing the library to the new Data Arena because the hardware of the Data Arena will be unfamiliar. D3 also has many tutorials and teaching resources available. Aside from the available free online resources there are also a number of books such as Mike Dewar's (2012) 'Getting Started with D3', Pablo Navarro Castillo's (2014) 'Mastering D3.js', and Scott Murray's (2013) 'Interactive Data Visualisation'. These online and the book resources are advantageous as they provide training material for the D3 library for future developers and users of the Data Arena.

An alternative to D3 is matplotlib, an open source Python library primarily focused on providing publication quality 2D visualisations (Hunter 2007; Wang et al. 2015). It is also an established library with books and online resources available for first time users, including Benjamin Root's (2015) 'Interactive Applications Using Matplotlib' and Shai Vaingast's (2009) 'Beginning Python Visualization'. Like D3 matplotlib provides interactivity (Bostock, Ogievestky & Heer 2011; Hunter 2007). Matplotlib used with the BioPython library provides tools for bioinformatics (Cock et al. 2013). Matplotlib has an advantage over D3 because Omegalib has a Python API which may allow easier integration (Febretti et al. 2014).

There are a number of libraries available that are very new, such as Bokeh or Vispy, which could also be used in the Data Arena. Both Bokeh and Vispy are Python libraries, which once again has the advantage of being used in conjunction with the Omegalib Python API (Febretti et al. 2014; Rossant & Harris 2013; Wang et al. 2015). Vispy has the additional benefit of being built on OpenGL, which is supported by Omegalib (Febretti et al. 2014; Rossant & Harris 2013). However, they are less established than matplotlib and D3, which is a risk. There is also comparatively less literature found. They are also newer libraries with core functionality that is likely to still be in development. While this means there are many opportunities for future work in these libraries, they currently do not have the same amount of tested functionality and reliability to use for this project.

Despite the advantage of the Python library of matplotlib, the prior experience of members of the acute lymphoblastic leukaemia project with the D3 library should encourage use and uptake of the Data Arena. Only open source libraries were considered for the project, as the code developed for this project would also be open source to allow for modification by future developers in the Data Arena.

2.7 Parallel Coordinates

Parallel coordinates were first developed in 1885 by Maurice d'Ocagne and were then rediscovered in the 1959 by Alfred Inselberg (Tricaud, Nance & Saadé 2011). Parallel coordinates are a series of parallel axes where each data point is represented by a line (Inselberg & Dimsdale 1990; Siirtola et al. 2009). A data point is made up of a series of coordinates for the parallel axes. A line represents an observation in the data with an axis representing a dimension. Thus, N-dimensions can be viewed on a 2D display (Johansson et al. 2005). Figure A demonstrates an interactive parallel coordinates graph for a car data set. Each car is a line in the graph. Parallel coordinates can view multiple dimensions simultaneously, which makes them a useful tool for multidimensional data sets. Comparing the shape of individual lines and groups of lines assists in analysis of the data. However, the shape of individual lines do not have inherent meaning, such as the slope of the line, as it is dependent on the order of the axes, which can be changed (Few 2009). There have been many applications for parallel coordinates. It has been used in air traffic control, robotics, statistics, computational geometry, instrumentation, computer vision, digital forensics, geographic information systems, social networking, earth sciences data, and bioinformatics (Cheng et al. 2008; Inselberg & Dimsdale 1990; Moustafa 2011; Tricaud, Nance & Saadé 2011).

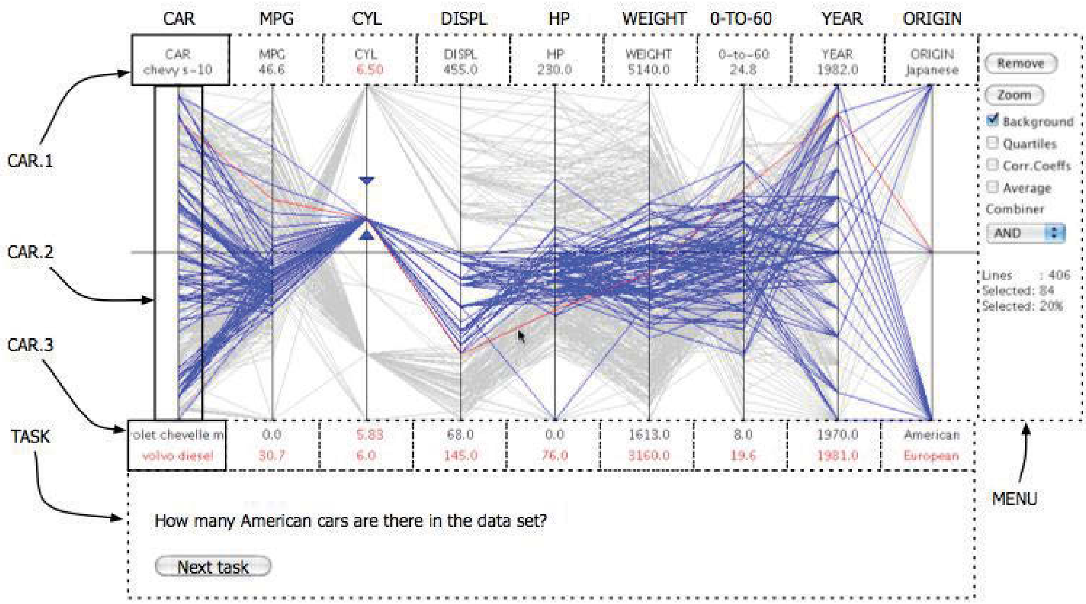


Figure A – A parallel coordinate graph surrounded by interactive software and diagrammatic cues (Siirtola et al. 2009).

Parallel coordinates are used to their greatest advantage when they are interactive. Without interactivity parallel coordinates can only give an overview of exceptions and prevalent patterns (Few 2009). Interactive elements such as filtering, brushing, similarity ranking, and similarity clustering provide greater analysis with parallel coordinates. Brushing is the selection of a subset of items in the graph and highlighting them. Filtering is the selection of a subset of items in the graph and removing all items from the graph except the subset. Similarity ranking and clustering use algorithms that group the items based on the similarity of the line. Ranking highlights all similar lines to the line selected. Clustering algorithms colour the lines based on similarity groups (Few 2009). Each of these methods may reveal different insights about the data, and most of these methods can be used in other visualisations in addition to parallel coordinates.

There are a couple of limitations with parallel coordinates. One limitation is that hidden patterns in a hypersphere and inliers are difficult to detect with basic parallel coordinates (Moustafa 2011). Parallel coordinates are also limited by aspect ratio and screen resolution. This limits the number of dimensions that can be viewed using parallel coordinates (Moustafa 2011). However, by increasing the screen resolutions and display size, such as using the Data Arena to view parallel coordinates, more dimensions can be viewed. Another limitation of parallel coordinates is that they can suffer from over-plotting for large data sets causing cluttering of the visualisation. However, there have been many developments to overcome these issues including filtering, clustering, and data transformation (Johansson et al. 2005). Splitting one parallel coordinates graph into many parallel coordinate graphs arranged as a matrix can assist in overcoming over-plotting. There may be additional advantages to analysing in this way because separating the parallel coordinates based on similarity of lines could provide additional insight (Few 2009). It could also provide the ability to find exceptions or patterns based on similarity of shape and then to compare those shapes for multivariate insight.

Alternatives to parallel coordinates for multivariate and multidimensional analysis include heatmaps, small multiples, visual crosstabs, table lens, glyphs, and concurrent views with brushing abilities. Heatmaps use a matrix similar to a spread sheet, with records being on one axis and variables being the other, but they encode meaning in colour rather than using text. Although they provide a good overview and can reveal exceptions, it is difficult to search for patterns (Few 2009). Also, because it relies on colour there are many constraints in ensuring that the contrast and colour are the best suited for optimal analysis. Small multiples, otherwise known as trellis display, are a series of graphs that are the same type, shape, and size (Few 2009; Tufte 2002). They can be arranged vertically, horizontally, or as a matrix. Small multiples encourage comparative, multivariate analysis which is efficient in interpretations (Tufte 2002). All graphs must be viewable simultaneously. They are most effective when arranged in meaningful order (Few 2009). Each graph in the sequence differs according to one variable. Visual crosstabs are similar to small multiples; however, each graph differs in more than one variable. The scales can differ in this visualisation because of the multiple differing variables. These are powerful methods to visualise multiple dimensions simultaneously (Few 2009). Table lens is a visualisation that is arranged like a table where rows encode items that were measured and the columns encode the variables (Few 2009; Rao & Card 1994). Horizontal bars are then used to encode the measurements (Few 2009). It is simple to view correlations between variables; however, it is better when interactive. Glyphs are graphical objects which are composed of visual attributes that encode particular variable measurements. There is limited real-world use for Glyphs (Few 2009). Despite this there may be specialised analyses where this may be useful. Concurrent views are similar to small multiples and concurrent tabs; however, they have multiple kinds of graphs in the one view. They are able to display multiple perspectives which is difficult for traditional visualisation (Few 2009). However they are more useful when brushing is incorporated because the user could highlight information in one graph and have the same subset highlighted in a different graph. Although each of these alternatives have certain advantages, the strength of parallel coordinates data visualisation in the Data Arena is that many more dimensions are viewable because of the Data Arena's increased display size and resolution. Parallel coordinates ability to find exceptions and patterns is also useful for the acute lymphoblastic leukaemia project. Nevertheless, these alternative data visualisations can also be created in D3 for the Data Arena in the future for confirming or exploring patterns and relationships discovered with parallel coordinates, as well as for the discovery of new patterns and relationships in the data.

2.8 Research Gaps

HRE is a recently developed display platform and not many have been built. Therefore there is less research on HREs compared with other virtual environments. Additionally, as HREs are relatively novel substantial research is needed to evaluate which methods, libraries, and plots will work well with this medium. There are no peer-reviewed research papers available that focuses on using D3 in the Data Arena and how to build and view plots that use libraries like D3. Additionally there are no research papers available that attempt to use parallel coordinates in an HRE. This thesis will attempt bridge these gaps by providing a method to use D3 to view and interact with parallel coordinates plots and other multidimensional visualisations, as well as providing clear instructions for other researchers using 2D visualisation in the Omegalib environment.

3. Methods

There are two phases that correspond to the outline of research methods described in Section 1.8. The development method is based on iterative development and evaluation of features. The evaluation method examines the final solutions in the Data Arena.

This section explains the methods and tools used, solutions developed and the test plan that will be used to evaluate the solutions. The section also provides information on solutions that were not carried forward to the evaluation phase because they were (a) either discontinued early because of feasibility concerns (Section 3.3.1 Qt and Omegalib Solution), or (b) discontinued after development because of early failure in the PC environment (Section 3.3.2 OSG SVG Reader Solution). These failed solutions are briefly described, so that the rationale of the final solutions (Section 3.4 WebView Solution and Section 3.5 SVG Parser Solution) is understood. Throughout the development process discussions with the Data Arena developers were held to ensure that the project stayed close to what would be needed in the Data Arena.

3.1 Development and Evaluation Methods

3.1.1 Development Method: Feature and Time Driven Methodology

There are two approaches to the development of parallel coordinates using D3 in a HRE. First one can integrate the D3 library into the HRE then build the parallel coordinates or one can develop the parallel coordinate in D3 first then integrate that part effectively into the HRE. The first option is preferable because by working from core functionality outwards it is easier to make cleaner, modular and testable code. Omegalib is a software framework for HRE that is used in the Data Arena. This first approach was the approach used for three solutions: (i) Qt and Omegalib and (ii) Open Scene Graph (OSG) SVG Reader solution, both of which are described in Section 3.3 Failed Solutions and (iii) WebView solution described in Section 3.4 WebView Solution. The second approach, develop the parallel coordinate in D3 then integrate that part effectively into Omegalib was used in for the SVG Parser Solution described in Section 3.5 SVG Parser Solution.

The task of developing parallel coordinates in the Data Arena using D3 was split into multiple features. Each feature was developed then examined in the PC environment in an iterative process. If successful, this feature was further examined by the evaluator (see Figure B). If the feature was accepted then the next feature was developed for examination.

Otherwise the original feature was modified based on feedback and then re-examined. This iterative process continued until all necessary features had been coded. Features of the task were ranked on the necessity of the feature. Necessity was determined by whether the feature was needed for the success of the project, whether another necessary feature was dependent on the delivery of this feature, and discussion with the stakeholders.

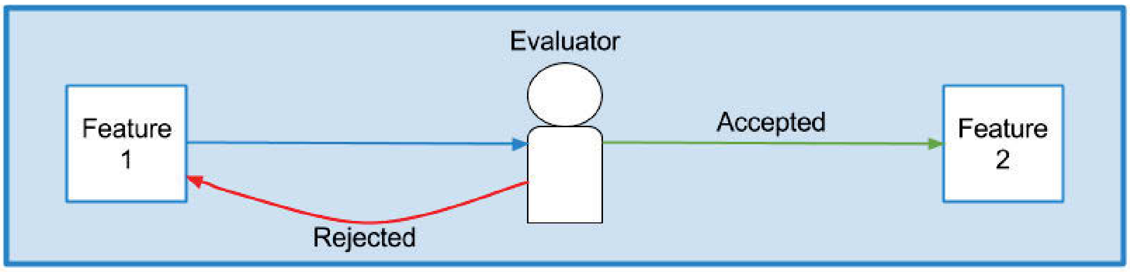


Figure B - Progression Through Feature Evaluation

The programming of the each feature used a simple subset of the acute lymphoblastic leukaemia dataset to test the components of each feature. These components were a simple pass or fail test. For example, did the visualisation appear on the screen?

Python and JavaScript coding standards and styles were used for programming. Coding standards and styles improved readability of code. High readability is important, as there was time difference between dependant features when developing for the Data Arena. For example, if Feature 8 was dependent on a method written in Feature 1 and that method was not used for three weeks there could be difficulties in understanding the code and purpose of that method. High readability improved the efficiency of understanding, using, and modifying methods from early in the project. Not only was high readability useful during the development of parallel coordinates in the Data Arena for this project, but also high readability will be useful to any project that wished to use or extend the code from this project. The coding style for Python used was the Python Enhancement Proposal style guide on the Python website (van Rossum, Warsaw & Coghlan 2013). The coding style for JavaScript used was the Google JavaScript Style Guide, which is used for Google's open-source projects (Whyte et al. n.d.).

A limitation of the project was lack of time available in the Data Arena. The Data Arena is a heavily contended resource, limiting the amount of time an individual programmer can spend code development and testing in the Data Arena. To overcome this issue, the majority of the development process occurred in the Omegalib environment on a PC.

3.1.2 Evaluation Method

Once a successful solution in the PC environment was developed, this solution was evaluated in the Data Arena using a pre-specified test plan. Note that this was not an empirical test. It was simply an evaluation of whether the solution that was developed in the PC environment worked in the Data Arena. However, one empirical test that was planned but was not implemented because of time constraints is provided in Appendix 7.2. This may be realised in future research.

The test plan contained test cases based on the features of the D3 visualisations. Each of the test cases attempted to use those features in the Data Arena. If one of the components of the feature did not work in the Data Arena the solution failed that test. Table 1 is an example of the layout of a test case. The full test plan can be seen in Section 3.6 Test Plan.

Table 1 - Test Case Template

Test Case Number	XX
Test Case Description	Description goes here
Pass Criteria	Pass criteria outlined in a list
Fail Case	Description of the circumstances that the test is considered to be failed

3.2 Tools

A variety of tools were used in the development and use of the solutions. The Omegalib environment, virtual machines, and JavaScript and Python libraries played a part in the development and implementation of solutions. Due to the novelty of Omegalib and the Data Arena, many of the tools and how they were used required ongoing refinement through a process of trial and error.

Although Omegalib had been shown to function in Windows environment on a PC, because of inconsistencies with the Windows environment, a Linux environment was finally implemented because Linux is used in the Data Arena. Initially a Linux Ubuntu operating system was implemented. Subsequently a beta version of the Data Arena's virtual machine running the Gentoo Linux operating system was received for development. Ongoing development occurred in both Ubuntu and Gentoo environments.

The steps needed to install Omegalib in Ubuntu are shown next. These include the settings needed for the development and running of the solutions in a Virtual Box virtual machine, and an outline of the JavaScript and Python libraries needed for the solutions. The settings for the Gentoo environment are provided in Section 3.2.2. However, the Gentoo virtual machine can be obtained from the Data Arena developers.

3.2.1 Installation of Omegalib

As described above solutions developed away from the Data Arena required a local version of Omegalib initially installed on an Ubuntu virtual machine. Later in the development process this was replaced by a beta Gentoo virtual machine built by the Data Arena developers. Basic instructions for installation are outlined below. These instructions are an amalgamation of the instructions on the website (https://github.com/uic-evl/omegalib) as well as minor steps used to make it work on the virtual machine (Febretti et al. 2014).

To do a basic installation of Omegalib with the WebView module on Ubuntu, open the terminal and install the following:

- Install g++

```
sudo apt-get install build-essential
```

- Install OpenGL

```
sudo apt-get install mesa-common-dev

sudo apt-get install libglu1-mesa-dev –y
```

- Install CMake

```
sudo apt-get install cmake
```

- Install Git

```
sudo apt-get update

sudo apt-get install git
```

- Install dependencies

```
sudo apt-get install freeglut3 freeglut3-dev python-dev flex bison

sudo apt-get install xorg-dev
```

- Install Omegalib (instructions from the Omegalib GitHub page)

```
mkdir omegalib
```

```
cd omegalib

wget http://omegalib.s3.amazonaws.com/maintenance-utils/omega

chmod +x omega

./omega get master common-modules
```

- If './omega get master common-modules' does not get and build Omegalib go to the omegalib directory and run the command ‘`rm -r cmake`’ then run ‘`./omega get master common-modules`’ again
- Install WebView

```
./omega get master webView
```

- Install Awesomium (go to their website and follow instructions)

If the version of Omegalib is not detecting all the OSG and other libraries on the virtual machine, the following may need to be executed ‘`make –B <particular module/directory`’ in the Omegalib build directory. If more control is needed over the Omegalib build and installed modules, obtain the CMake GUI by executing the command ‘`sudo apt-get install cmake-gui`’. Then use the CMake GUI and the ‘`make`’ command to customise and build the Omegalib version. The Omegalib website contains more information about custom builds (Febretti et al 2014).

3.2.2 Virtual Box

As mentioned previously, a virtual machine was used as a development environment for this project. The virtual machine software Virtual Box was used for this process. It is important that 3D acceleration is enabled for the virtual machine as otherwise the WebView module and other components will not be able to work to their fullest potential. In particular, canvas objects in a webpage are extremely slow to load and update when displaying in Omegalib using WebView when 3D acceleration is disabled.

The settings for each of the virtual machines, first the Ubuntu virtual machine then the Gentoo virtual machine (the latter built by the Data Arena developers) are shown as screen shots in Figures C and D respectively.

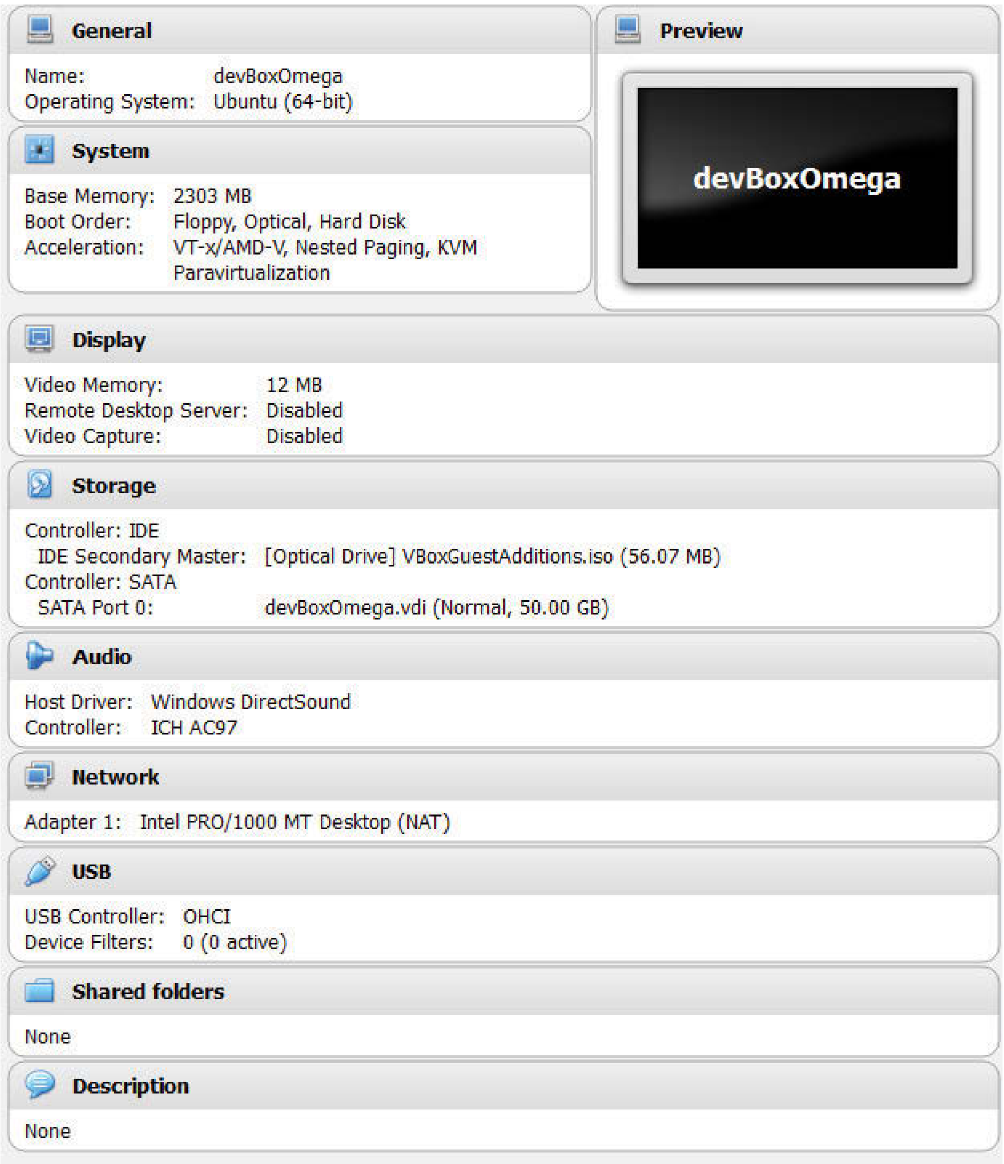


Figure C - Ubuntu Virtual Box Settings

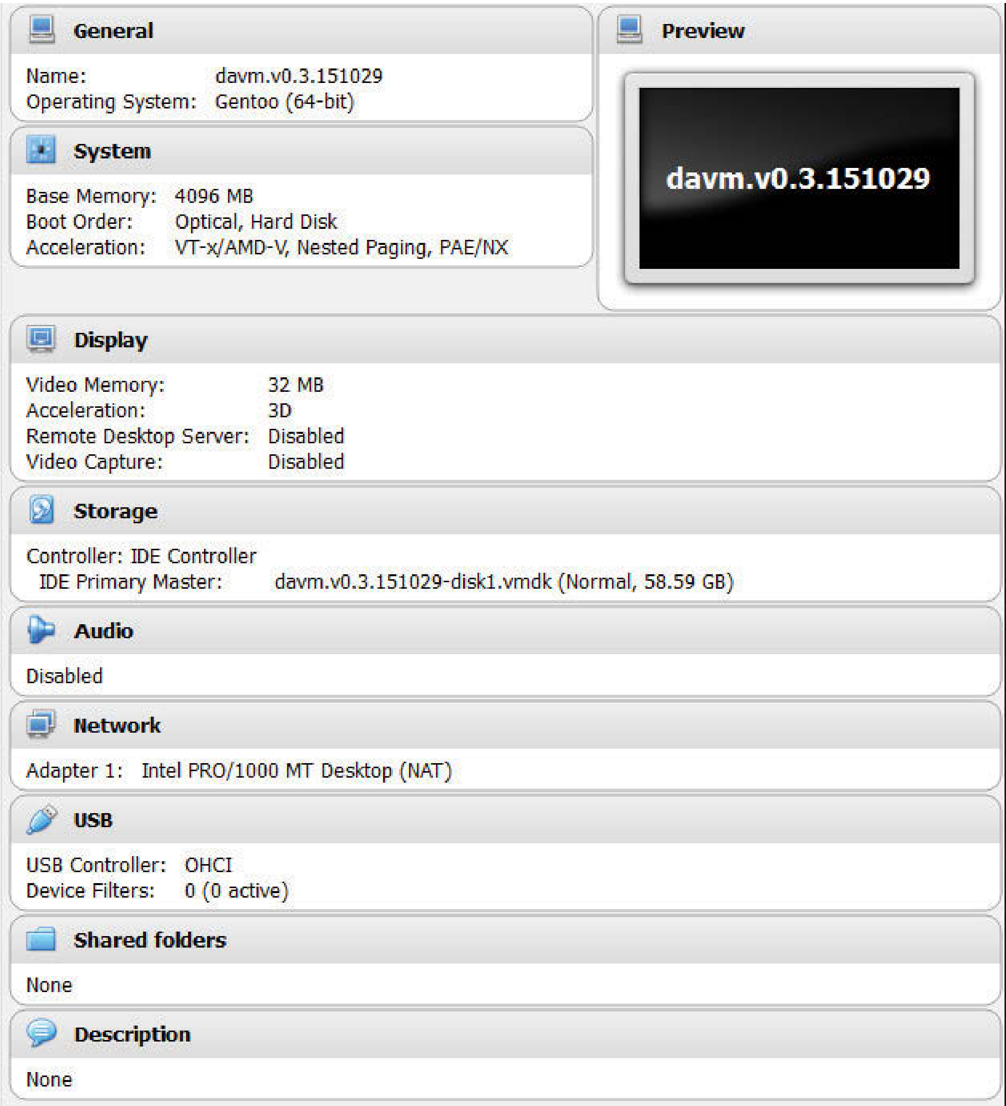


Figure D - Gentoo Virtual Box Settings

3.2.3 Required Libraries

In addition to the libraries mentioned in Section 3.2.1 Installation of Omegalib, there are a libraries that are required for the solutions outlined in Section 3.4 WebView Solution and Section 3.5 SVG Parser Solution: a Work in Progress.

The WebView solution requires two primary JavaScript libraries for the development and implementation of parallel coordinates. These include the d3.js library by Bostock, Ogievestky and Heer (2011) and the parallel-coordinates library by Chang (2012). D3.js library was outlined in Section 2.6 Visualisation Libraries. The parallel-coordinates library is a plugin for d3.js to more easily build the parallel-coordinates plots. The parallel-coordinates library is linked on the D3.js plugins webpage. The parallel-coordinates library also has a dependency on the sylvester.js library by James Coglan. This library is needed for vector and matrix calculations and is included in the examples/lib directory when obtaining the parallel-coordinates library.

The SVG Parser solution requires one JavaScript library and one Python library. The webpage used to generate the simple parallel coordinates SVG for the SVG Parser solution only requires the d3.js library. The Python script executed in Omegalib's Python script reader orun requires the xml reader library xml.etree.ElementTree. This library is needed to parse the SVG file into a Python object. The Python object is used to read each of the elements in the SVG. These elements are then translated into Omegalib's Cyclops module elements.

3.2.4 Acute Lymphoblastic Leukaemia Data and Other Data Used in the Development Process

Two main datasets were used in the development process of the parallel coordinates. One was a simple example dataset from the examples page of the parallel coordinates plugin. The dataset was the matrix used for the minimal example (Chang 2015). This simple dataset was used for the initial development of the parallel coordinates. For later development and testing a subset of the acute lymphoblastic leukaemia data was used. This subset had only 16 features (from a set of approximately 36,000 features) so that development and testing could focus on problems not caused by storage and handling of extremely high dimensional datasets.

3.3 Failed Solutions

In the process of developing an approach to the problem of parallel coordinates and d3.js in the Data Arena two methods, Qt and Open Scene Graph (OSG) SVG Reader, were considered but were subsequently discontinued because of either feasibility concerns or because of early failure during development.

The preferred solution was the WebView solution. WebView had interactive benefits because it is a browser. Furthermore, WebView already implemented as an Omegalib. However, previous experience with the WebView solution had aliasing of the lines of the parallel coordinates. Therefore, Qt and OSG SVG Reader methods were considered early in the development process. It was hoped that either Qt or OSG SVG Reader would solve the aliasing issue and interactivity would be built in at a later date. Unfortunately both the Qt and OSG SVG Reader methods were discontinued for reasons outlined below.

3.3.1 Qt and Omegalib Solution

Qt is a popular tool for cross-platform application development. One of the reasons it was considered for this project is its extensive API with many comprehensive and modularised library classes (The Qt Company 2015). One of these libraries includes the ability to render SVGs. Most d3.js plots are built and rendered as an SVG. As SVGs are scalable they should not have an aliasing issue when rendered using the appropriate methods. Testing the Qt SVGViewer application examples showed that the SVG image was rendered properly and did not have the aliasing problem that was present in the WebView solution.

However, testing basic Qt elements within Omegalib using Python scripts identified problems. The Qt application appeared to be slightly separate to the orun application that runs the Python scripts. The container was able to move outside the bounds of the orun application. This raised concerns about how the Data Arena would be able to handle the Qt elements. Additionally a Python/Qt interpreter was also required. During development Py-Qt was used in order to allow Qt to be accessed with a Python script. Writing the solution in C++ was an option to avoid having to use a Python/Qt interpreter. However, the developer had no experience with the C++ language. Therefore, this option was not feasible given the time constraints of the project. When discussing these concerns with the Data Arena developers, the developers proposed another possible solution. Consequently, for this specific project, the Qt method was discontinued.

3.3.2 OSG SVG Reader Solution

The Data Arena developers proposed using the SVG Reader plugin for OSG to read and display an SVG file as D3 can generate visualisations as SVGs. OSG is supported by Omegalib and therefore the OSG plugins. However, the development build that was being used to develop the code for this project did not get the OSG plugin initially. After installing the appropriate libraries and forcing a clean rebuild of the OSG module, 'make –B osg', the OSG SVG plugin could be used when reading SVG image files in the Python script. The SVG was implemented by using it as a texture of a plane that was created and displayed in orun. Unfortunately, this method had the same aliasing problem as the WebView method. The Omegalib OSG SVG reader appeared to be rendering the image using the dimensions of the image file without scaling the image to meet the dimensions of the plane. Furthermore, the image did not adapt when scaling the plane to larger sizes. Therefore the OSG SVG reader solution had the same aliasing problem as the WebView solution. Given, that

Webview already had built in interactivity, the OSG SVG reader solution was discontinued and the WebView solution was re-considered with renewed efforts to mitigate the aliasing.

3.4 WebView Solution

Initially, it was thought by doubling the resolution of the webpage object in Omegalib then resizing the frame that contains the webpage object, would mitigate the aliasing. There are two parts to this WebView solution. The first part was developing WebView Python script that allows webpages to be viewed in the Data Arena. The second part is a HTML file that contains the JavaScript code used to generate the parallel coordinates plot of acute lymphoblastic leukaemia data subset.

The first file, the WebView Python script, was developed and it worked in a PC environment. However, this WebView Python script, when tested in the Data Arena did not work. The WebView Python script needed additional code to work on a cluster system, but did not need this additional code to work in a PC environment.

Despite this failure, the developers proceeded to test the second file HTML file in the Data Arena using the original WebView Python script that had been developed by Data Arena developers, the same script that had aliasing issues when tested at the beginning of the project. During this testing of the HTML file that contained the JavaScript code used to generate the parallel coordinates, unexpectedly, the aliasing that was the original problem with the WebView solution, seemed to have disappeared. The lines in the parallel coordinates had lost their jagged effect, and now appeared as smooth continuous lines. One explanation for this improvement was that ongoing development of the Data Arena had somehow diminished the problem of aliasing, at least for the line thickness needed for the parallel coordinate plot.

Therefore, with the Data Arena developer's permission, a simplified version of the original WebView Python script was created. This version served as the companion Python script for the HTML file that contained the JavaScript code used to generate the parallel coordinates. These two files now comprised the WebView solution that was further tested in the evaluation phase.

3.5 SVG Parser Solution: a Work in Progress

Simultaneous with reconsideration of the WebView solution the SVG Parser solution was developed. The SVG Parser solution is a work in progress proof of concept. It is based on the idea that by translating SVG elements into supported objects in Omegalib, many of the rendering issues would be solved. This method should be able to take any d3.js plot that is saved as a SVG and parse it into Omegalib elements. It is important to note that to save a SVG image so that it has the same appearance as an SVG image in a webpage, the styles for each of the elements must be in line with the element, and not in a separate CSS section or file. This is because the SVG image must be self-contained when saved. Future work may want to examine how to extract the SVG and the styling from a webpage.

As it is a work in progress, the development of features has been focused on a simple parallel coordinates plot based on Bostock's (2011c) example of parallel coordinates with a dataset on cars. The code was adapted to have styling as an element attribute for the paths and axes. Additionally, the code was simplified to remove interaction features such as brushing. This was because only the generated SVG image was needed at this stage of development. To aid visualisation only numerical attributes of the acute lymphoblastic leukaemia data were used in the parallel coordinates. The webpage was opened in a browser to generate the SVG. By using the 'Inspect Element' tool that occurs on most browsers, the code for SVG was copied into a file. As SVGs are in the XML format an XML parser was used in the Python script to read the SVG file. This parser stored a tree of elements, which had a method that would iterate through all the elements. Looping through each of the elements, a series of if statements were used to determine the processing needed and the type of Omegalib element added to the scene. The if statement was based on the element tag, for example 'svg', 'line', and 'path'. At this stage, the styling of the elements was ignored. Instead development focused on generating the axes and paths. Due to the use of groups to position axes and scale ticks, translating the elements to the Omegalib elements was a difficult task, especially as the parent of the current element was unable to be retrieved. The built in class names for the groups and text assisted the positioning of elements by being able to indicate when the translation values should be used and updated. As this is very plot specific, it is recommended that the XML parser library be changed to another library where the parents of elements can be retrieved. The code for the SVG Parser Python script, HTML page, and SVG image can be found in Appendix 7.1.2. The SVG Parser solution for parallel coordinates plots only were further tested in the evaluation phase. Therefore this solution remains a work in progress.

3.6 Test Plan

A test plan outlining the criteria for evaluation of the WebView and SVG Parser solutions was developed. For the purposes of this project empirical tests, although considered, were not implemented because of time constraints. Rather, the test plan is an evaluation of the solution. However, a plan of empirical testing that was considered is provided in Appendix 7.2 and this empirical testing may be useful for future research. The test plan first describes the features that will be tested and then presents the test cases.

3.6.1 Features to Test

The features that were tested for the WebView solution can be broken down into two parts. First is the testing of the parallel coordinates plot features in the Data Arena, based on the features of the parallel coordinate webpage. The visibility of elements and interactivity features are the main features tested using the test cases. A commentary of how well the parallel coordinates plot is rendered in the Data Arena is provided in the results. Second is the testing of the WebView solution on other 2D plots. These plots were selected because they can be used to view multiple dimensions. The plots were obtained from the D3.js examples webpage (GitHub 2015).

The features to test for the SVG parser solution were the expected finished features by the day of testing. As this solution is a work in progress proof of concept, tests focused on the available elements attempting to render a parallel coordinates SVG image. The main tests were on how the lines are rendered and whether they were similar to the lines on the SVG image, and on the rendering and positioning of the axes.

3.6.2 Test Cases

WebView Solution Test Cases

Table 2 - Test Case 01

Test Case Number	01
Test Case Description	Load the local parallel coordinates webpage in the Data Arena using the python script
Pass Criteria	Loads the parallel coordinates plot Loads the legend Loads the table
Fail Case	At least one of the elements does not load within 30 seconds

Table 3 - Test Case 02

Test Case Number	02
Test Case Description	Interact with the parallel coordinates by selecting a small portion of one of the axes
Pass Criteria	All axes are selectable All selections are able to be deselected All lines under a selection are visible All lines not included in a selection are not visible
Fail Case	At least one of the pass criteria is not met

Table 4 - Test Case 03

Test Case Number	03
Test Case Description	Able to select two or more axes simultaneously
Pass Criteria	All selections are able to be deselected All lines under all selections are visible All lines under only one of the selections are not visible
Fail Case	At least one of the pass criteria is not met

Table 5 - Test Case 04

Test Case Number	04
Test Case Description	Interact with the parallel coordinates by reordering the axes
Pass Criteria	All axes are able to be dragged All axes can be reordered using drag and drop
Fail Case	At least one of the pass criteria is not met

Table 6 - Test Case 05

Test Case Number	05
Test Case Description	The parallel coordinates plot in the Data Arena is similar to the parallel coordinates plot in a web browser
Pass Criteria	Lines form the same patterns Lines have similar colours
Fail Case	At least one of the pass criteria is not met

Table 7 - Test Case 06

Test Case Number	06
Test Case Description	Load and view other types of d3.js plots using the WebView solution python script. The types of visualisations chosen are multidimensional plots that were briefly discussed in the literature review, as well as additional examples from the D3.js examples page. However, an available table lens example as described in the literature review could not be found for D3.js.
Pass Criteria	Loads and the user is able to view the following plots: Network Diagram • http://bl.ocks.org/mbostock/raw/4062045/ (Bostock 2012a) • http://www.whodotheyserve.com/ (Who Do They Serve 2015) Tree Diagram • http://mbostock.github.io/d3/talk/20111018/tree.html (Bostock 2011a) Scatterplot/Line Plot • http://charts.animateddata.co.uk/uktemp/ (Cook 2015) Visual crosstabs, scatterplot matrix with different scales • http://mbostock.github.io/d3/talk/20111116/iris-splom.html (Bostock 2011b) Heatmap • http://joshua-gould.github.io/public/examples/miserables/ (Bostock 2012b) • http://bl.ocks.org/oyyd/859fafc8122977a3afd6 (Bl.ocks.org 2014) Glyphs/Chernoff Faces • http://www.larsko.org/v/hpi/ (Kotthoff 2012) Small multiples • http://bl.ocks.org/mbostock/1157787 (Bostock 2011d) • http://bl.ocks.org/tylercraft/3630001 (Bl.ocks.org 2012a) Star Plots • http://annalyn-ng.com/starwarsb5/chart.html (Ng 2015) Concurrent Views • http://bl.ocks.org/diethardsteiner/raw/3287802/

	(Bl.ocks.org 2012b) • http://square.github.io/crossfilter/ (Square Inc. 2012)
Fail Case	At least one of the plots is unable to load and be viewed in the Data Arena

SVG Parser Solution Test Cases

Table 8 - Test Case 07

Test Case Number	07
Test Case Description	Be able to load and run the SVG Parser solution python script in the Data Arena
Pass Criteria	Loads the SVG Parser solution python script in the Data Arena Runs the SVG Parser solution python script in the Data Arena
Fail Case	At least one of the pass criteria is not met

Table 9 - Test Case 08

Test Case Number	08
Test Case Description	Render Paths
Pass Criteria	Renders the paths of the adapted Bostock SVG
Fail Case	The paths do not render The shapes the paths make do not match the SVG

Table 10 - Test Case 09

Test Case Number	09
Test Case Description	Render Axes
Pass Criteria	Renders the axes of the adapted Bostock SVG Renders the text for each of the axes The rendered axes match the adapted Bostock example
Fail Case	The axis do not render The scales are not correct on the axis The axis and path’s do not meet in the correct places

4. Results

The results section has been split into two parts: the results for the WebView solution and the results for the SVG Parser solution, respectively. Each part in turn has a section that comments on how parallel coordinates appear in the Data Arena and a test report. The test report contains the results of the test cases as well as general comments that highlight relevant issues and offers potential solutions. Each test case indicates whether it passed as well as specific comments on the results of the test. Some of the test cases included images of the Data Arena displaying the results.

4.1 WebView Solution

The source code for the WebView solution is in Appendix 7.1.1. The source code is of the script used to run the webpages and the parallel coordinates webpage. The user should ensure that when using the source code the correct libraries are in the appropriate directories. The libraries needed are explained in Section 3.2.3 Required Libraries.

4.1.1 Parallel Coordinates in the Data Arena

Parallel coordinates in the Data Arena using the WebView solution allowed for clear viewing of the coordinates and for interactive analysis. The dimensions and lines are easy to read and it is clear that a greater number of variables would also work within the Data Arena (see Figure E). This plot was about a third of the Data Arena in circumference, so if the dimensions look cramped to a user, or if more dimensions are incorporated, the width of the container could be increased. The text was a good size for reading even on the other side of the Data Arena. Figure F shows that although the lines are clear, the alpha and the colours could use some additional adjusting for better comparison and analysis. It is recommended that the lines go no thinner or the alpha is increased, as at lower levels it will be difficult to view the lines. It is also recommended that the colours are changed to darker blues, browns and greens for a clearer view of the lines.

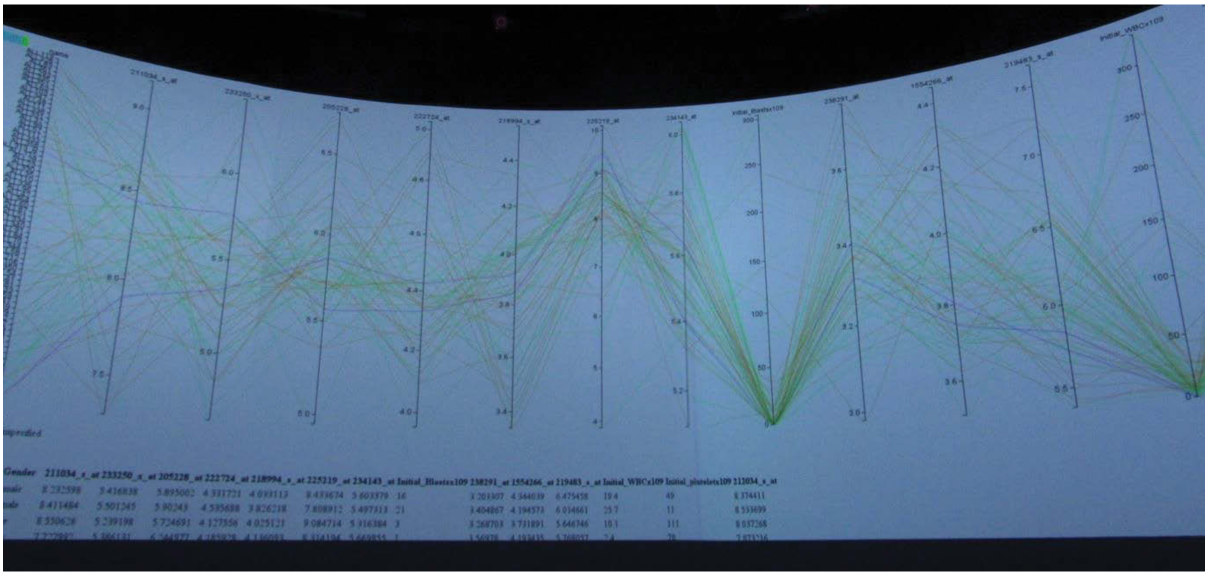


Figure E - Parallel Coordinates Webpage in the Data Arena

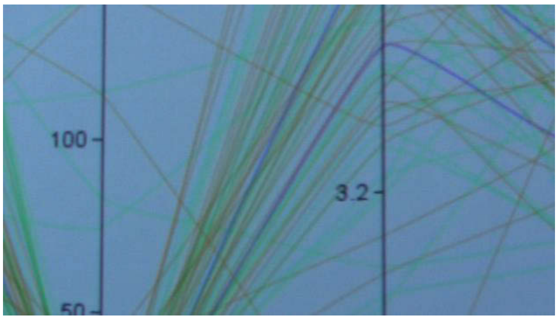


Figure F - Close-up of Parallel Coordinates Webpage in the Data Arena

4.1.2 Test Report

The WebView solution passed almost all the tests. The only failed test was Test Case 06, as not all of the example plots were similar to their browser version. Each of the test case's pass/fail status as well as a comment on the test are presented in the following Tables.

Table 11 - Test Case 01 Results

Test Case Number	01
Test Case Description	Load the local parallel coordinates webpage in the Data Arena using the python script
Pass/Fail	Passed
Comment	The parallel coordinates webpage loaded quickly in the Data Arena. All the elements loaded of good quality.

Table 12 - Test Case 02 Results

Test Case Number	02
Test Case Description	Interact with the parallel coordinates by selecting a small portion of one of the axes
Pass Criteria	Passed
Comment	Each of the axes could be selected. This was expected as they all have the same classes and each axis was generated the same way. The lines were brushed and hidden as expected based on the selection of the axis. However, there was one small issue with the interaction method of using a wand. It is very difficult to double click in the same place with the wand, which is what is required by the parallel coordinates plot to deselect axes. It is recommended that a mouse be included as an interaction method when using webpages to ensure there is another way to interact with the page if the wand becomes too difficult or tiring.

Table 13 - Test Case 03 Results

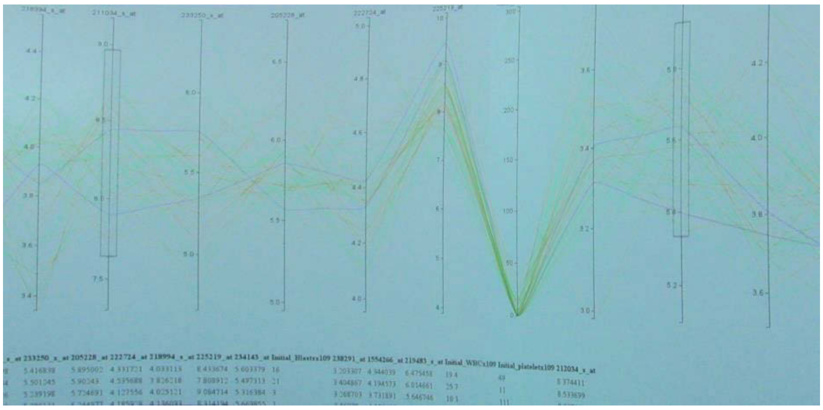
Test Case Number	03
Test Case Description	Able to select two or more axes simultaneously
Pass/Fail	Passed
Comment	Two or more axes are able to be selected simultaneously. The brushing and visibility of the lines are appropriate according to the selections. A similar issue occurred with deselecting the axes as was described in Test Case 02.  **Figure G - Multiple Selected Axes in the Data Arena**

Table 14 - Test Case 04 Results

Test Case Number	04
Test Case Description	Interact with the parallel coordinates by reordering the axes
Pass/Fail	Passed
Comment	All the axes are able to be dragged and can be reordered. The coordinates do not snap to position, in contrast to when viewing the plot in Chrome or Firefox. However, the axes do not snap when viewing the plot in Safari and Internet Explorer. Hence it is something that is common, thus will not cause a fail case in this test.

Table 15 - Test Case 05 Results

Test Case Number	05
Test Case Description	The parallel coordinates plot in the Data Arena is similar to the parallel coordinates plot in a web browser
Pass/Fail	Passed
Comment	The Data Arena presentation of the parallel coordinate plot is similar to the same page in a browser. The layout of elements is the same, with a larger page width causing a larger aspect ratio between dimensions in the plot, as could be seen if a monitor had a similar aspect ratio to the designated viewing area in the Data Arena. The colours are similar in the Data Arena, however red tones are diminished causing some colours to appear washed out. It is recommended that browns, blues and dark greens are used as line colours instead of the light green, orange and purple as used in this graph.

Table 16 - Test Case 06 Results

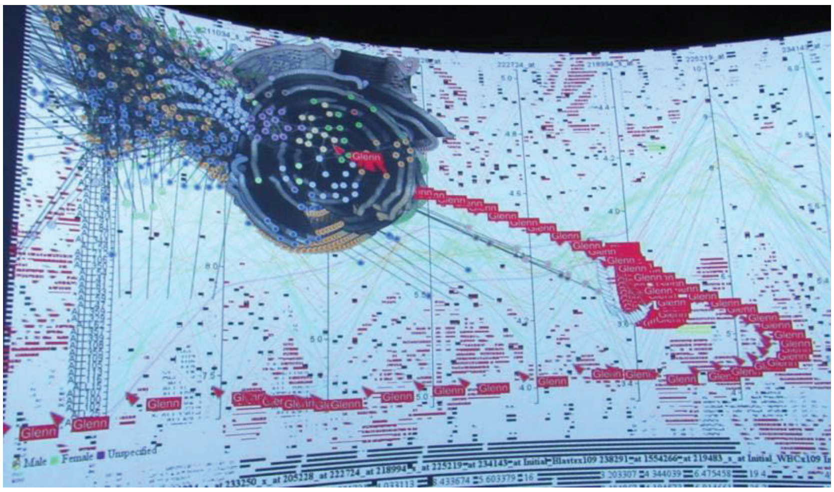
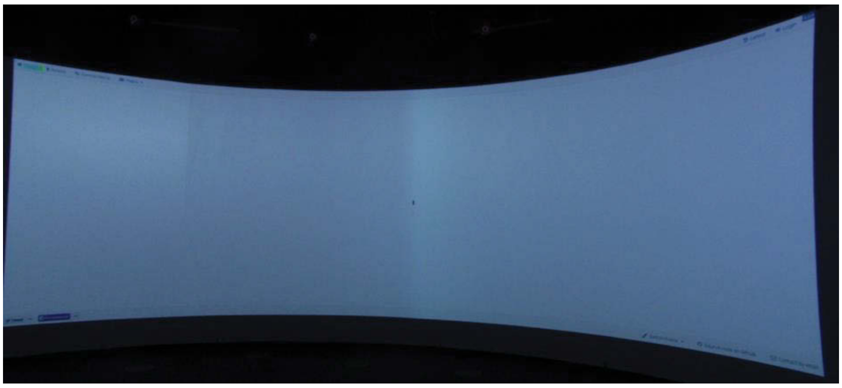
Test Case Number	06
Test Case Description	Load and view other types of d3.js plots using the WebView solution python script. The types of visualisations chosen are multidimensional plots discussed briefly in the literature review, as well as some additional examples from the D3.js examples page. However, an available table lens example could not be found for D3.js.
Pass/Fail	Failed
Comment	Loads and the user is able to view the following plots: **Network Diagram** Diagram: http://bl.ocks.org/mbostock/raw/4062045/ (Bostock 2012a) Result: The plot loaded but there was an issue with the previous image staying on the screen. In a browser the previous image does not stay on the screen, but this is not the case in the Data Arena. This issue will be further discussed in the General Comments section below. The plot was completely interactive, as was expected if it were loaded in a browser.  **Figure H - Force Directed Network Diagram in the Data Arena**

	Diagram: http://www.whodotheyserve.com/ (Who Do They Serve 2015) Result: The diagram did not load. Some aspects of the page loaded, including the header and footers. It is unclear why this occurred, however it could be to do with the WebView module not interacting correctly with an imported JavaScript file or other elements of the webpage.  **Figure I – ‘Who Do They Serve’ website in the Data Arena**

Tree Diagram

Diagram: http://mbostock.github.io/d3/talk/20111018/tree.html

(Bostock 2011a)

Result:

The collapsible tree diagram loads properly and is completely interactive. It is slightly difficult to click on the nodes to expand or collapse the tree because of small size combined with the wand mode of input.

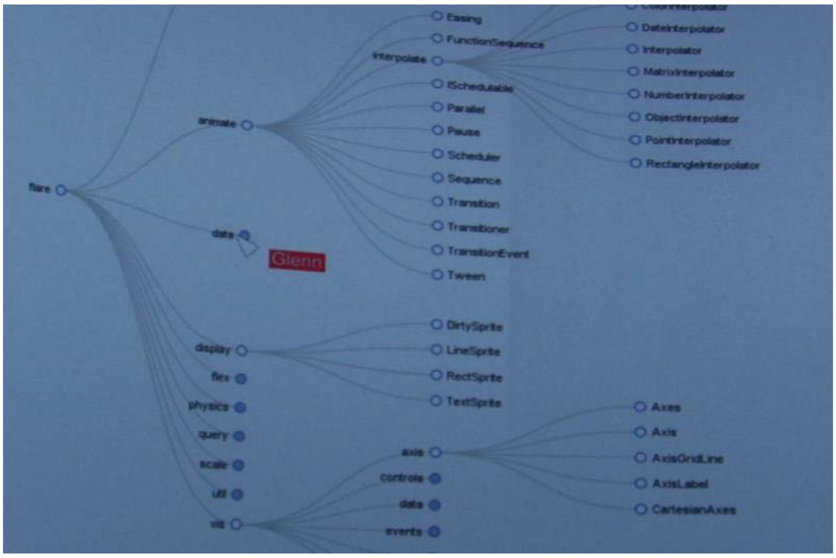


Figure J - Tree Diagram in the Data Arena with a Node Selected for Expansion

Scatterplot/Line Plot

Diagram: http://charts.animateddata.co.uk/uktemp/ (Cook 2015)

Result:

The plot is viewable and allows interactivity.

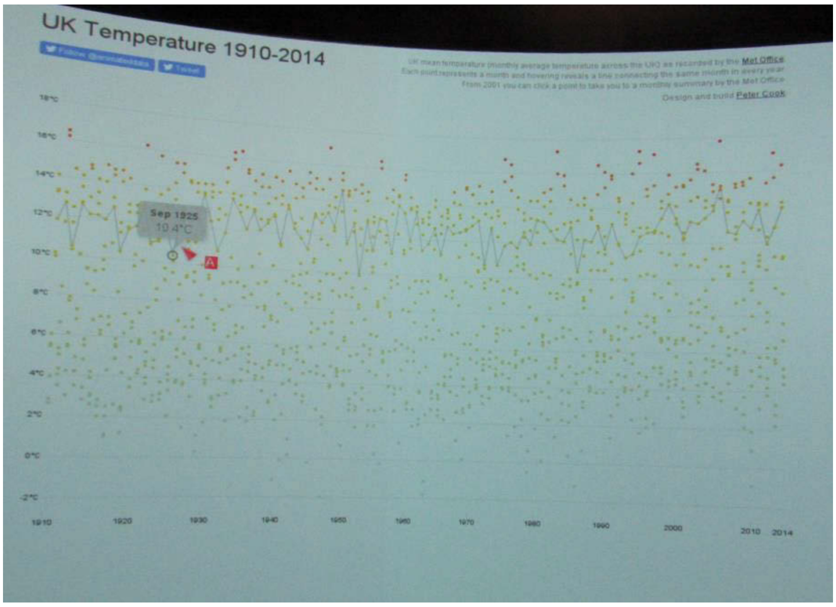


Figure K - UK Temperature 1910-2014 Scatterplot

Visual crosstabs, scatterplot matrix with different scales

Diagram: http://mbostock.github.io/d3/talk/20111116/iris-splom.html (Bostock 2011b)

Result:

The plot loads and is interactive however it has the same issue as the tree diagram with the previous images being left on the screen. Using the wand to select areas of the plot for brushing is simple, but requires learning.

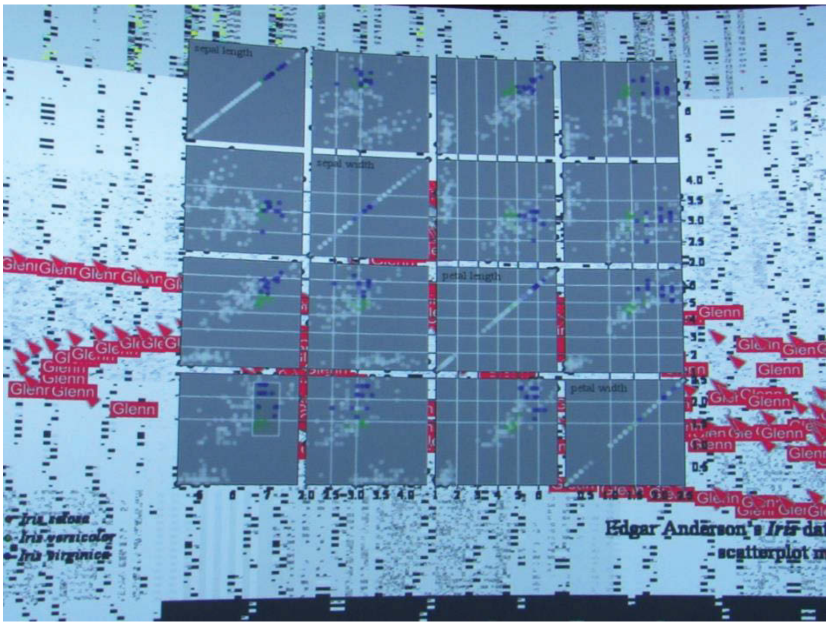


Figure L - Scatterplot Matrix in the Data Arena with Small Brushed Area

Heatmap

Diagram: http://joshua-gould.github.io/public/examples/miserables/ (Bostock 2012b)

Result:

The diagram loads correctly and is interactive. However it is difficult to use the drop down menu on the right hand side of the page. Once again this may be because of the use of the wand.

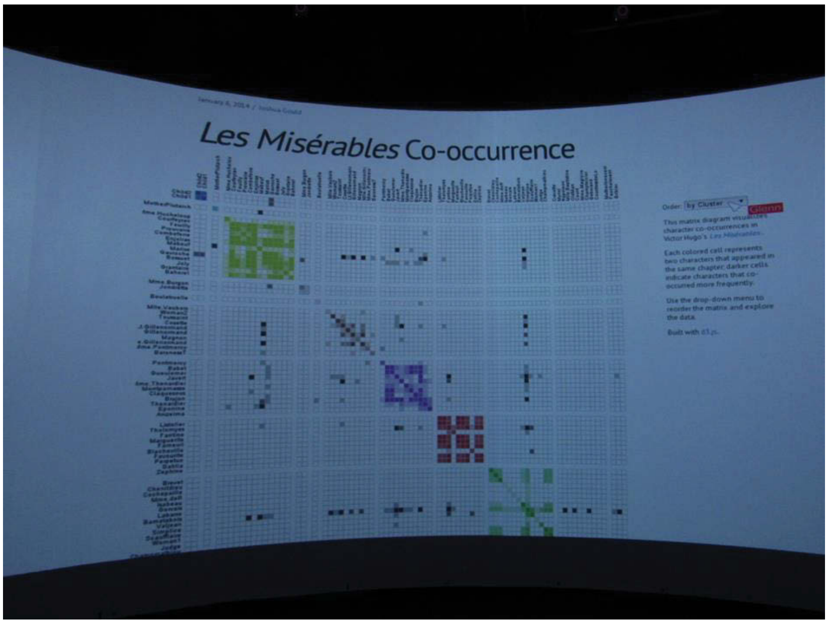


Figure M - Les Misérables Co-occurrence Heatmap in the Data Arena

Diagram: http://bl.ocks.org/oyyd/859fafc8122977a3afd6
(Bl.ocks.org 2014)

Result:

The page loads but the plot does not show and the same previous image problem occurs. The plot not loading may be because of a style-sheet loading incorrectly or running the JavaScript incorrectly.

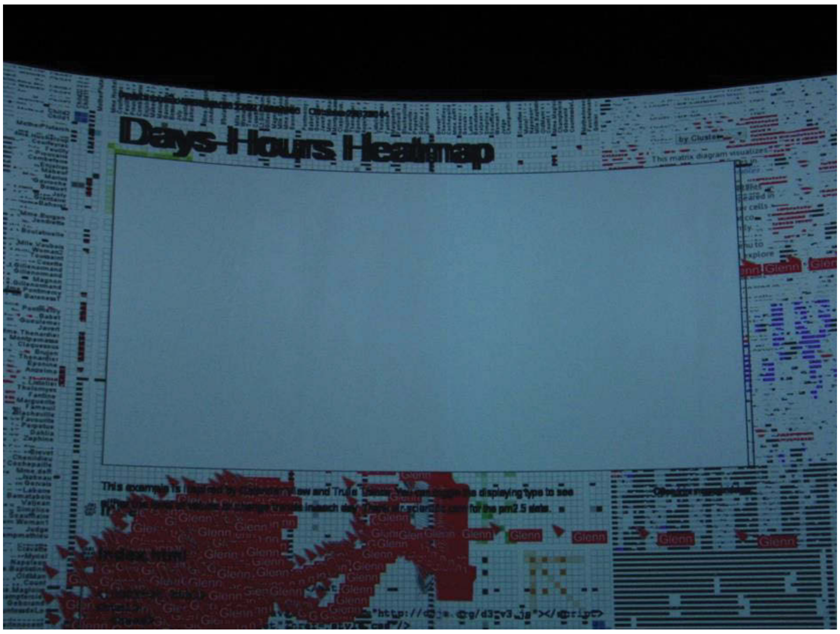


Figure N - Days-Hours Heatmap in the Data Arena

	Glyphs/Chernoff Faces Diagram: http://www.larsko.org/v/hpi/ (Kotthoff 2012) Result: The plot loads properly and is interactive. It is very similar to how it appears in the browser.  **Figure O - Chernoff Faces in the Data Arena**

Small multiples

Diagram: http://bl.ocks.org/mbostock/1157787 (Bostock 2011d)

Result:

The small multiples plot loads but has the same background issue that some of the other webpages have.

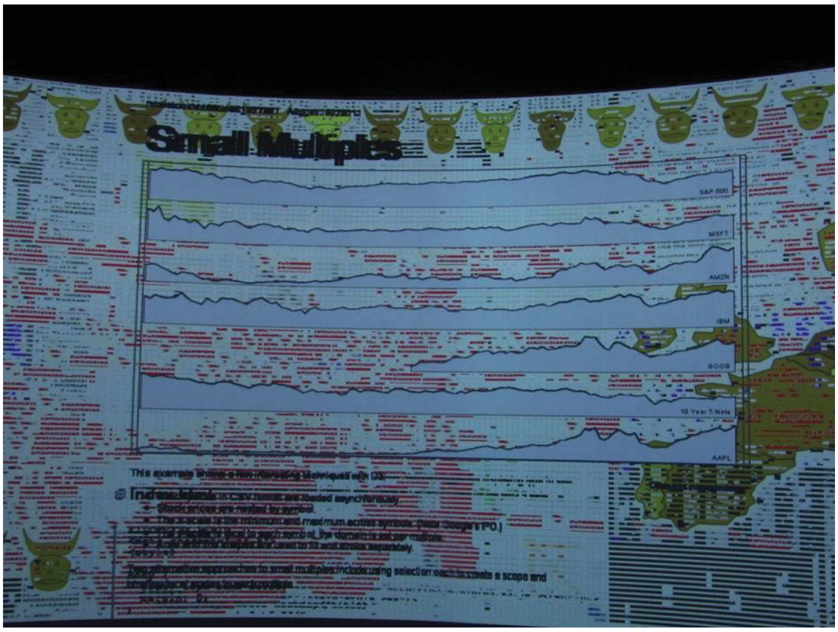


Figure P - Small Multiples in the Data Arena

Diagram: http://bl.ocks.org/tylercraft/3630001 (Bl.ocks.org 2012b)

Result:

The small multiples plot loads and is able to be easily interacted with using the wand. However, this page also has the same page background issue and the background area of the plot has been darkened. The plot background being partially transparent may have caused the darkened area of the plot.

Figure Q - Interactive Small Multiples in the Data Arena

Star Plots

Diagram: http://annalyn-ng.com/starwarsb5/chart.html (Ng 2015)

Result:

The plots load and can be viewed. However the background issue causes the text on the page to be almost unreadable. The star charts are viewable and the larger aspect ratio for the webpage means that more star plots can be next to each other than in a normal browser on a monitor.

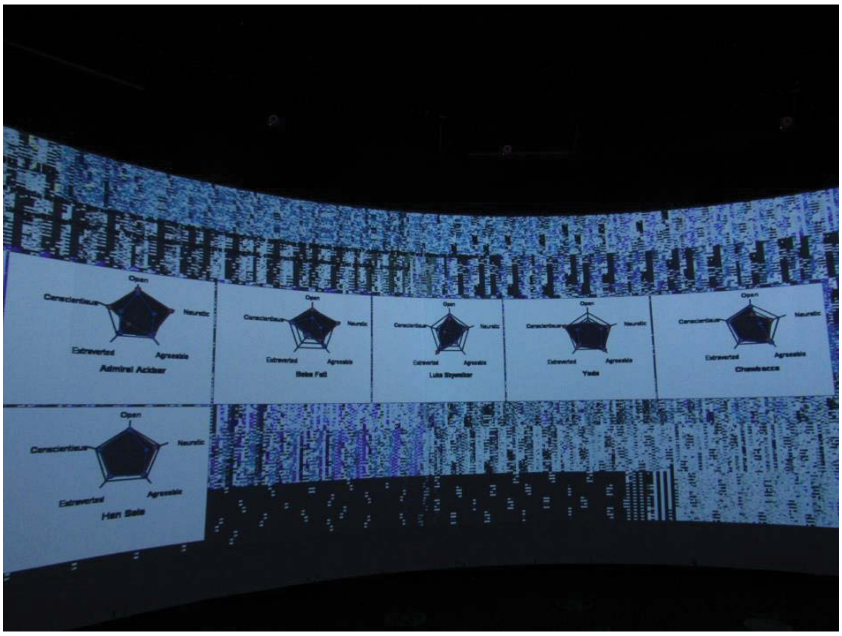


Figure R - Star Plots in the Data Arena

Concurrent Views

Diagram: http://bl.ocks.org/diethardsteiner/raw/3287802/

(Bl.ocks.org 2012)

Result:

Once again there is a background issue with the remains of previous images staying on the screen with only the plot elements with full alpha being unaffected. However, the plots still work and interacting with the pie chart appropriately updates the other elements.

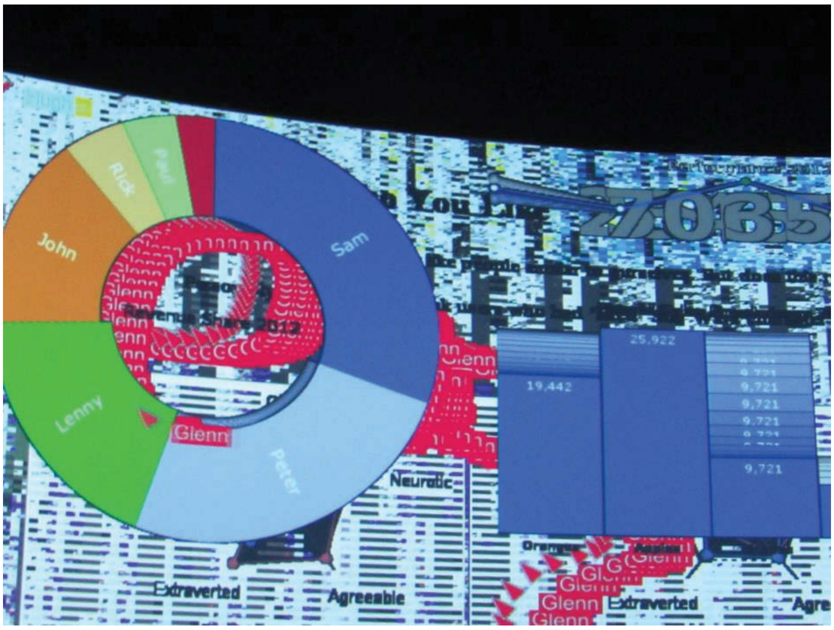
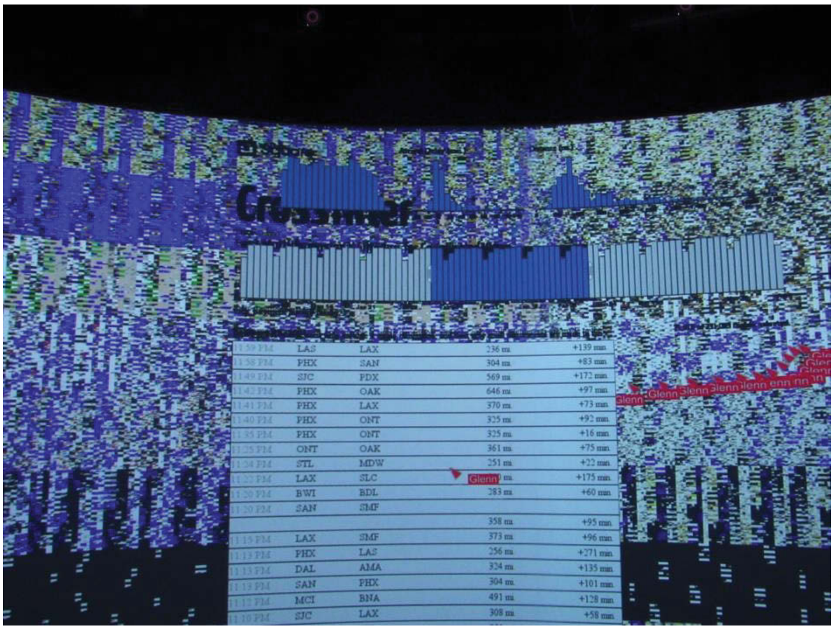


Figure S - Dashboard in the Data Arena

	Diagram: http://square.github.io/crossfilter/ (Square Inc. 2012) Result: The plot and the tables load, however the previous images remaining on the screen in transparent areas make some parts difficult to read, such as the text for the scales. The plot is still interactive. By brushing the long time-based plot the table and the other three plots are updated.  **Figure T - Crossfilter in the Data Arena**

General Comments:

The main issues with the WebView solution were twofold. The Omegalib system would not clear the background image with new iterations of the script and the default colour for the background is a dark colour. As some of the webpages had backgrounds that were transparent, it caused the effect seen in Figure U. This could be fixed by setting the 'background-color' in a style attribute on the body HTML tag to 'white'. This caused the webpage to behave as expected, as can be seen in Figure V. This effect was also duplicated for the second concurrent views visualisation (Square Inc. 2012). Once the body tag had a white background, the webpage appeared the same as when run in a browser.

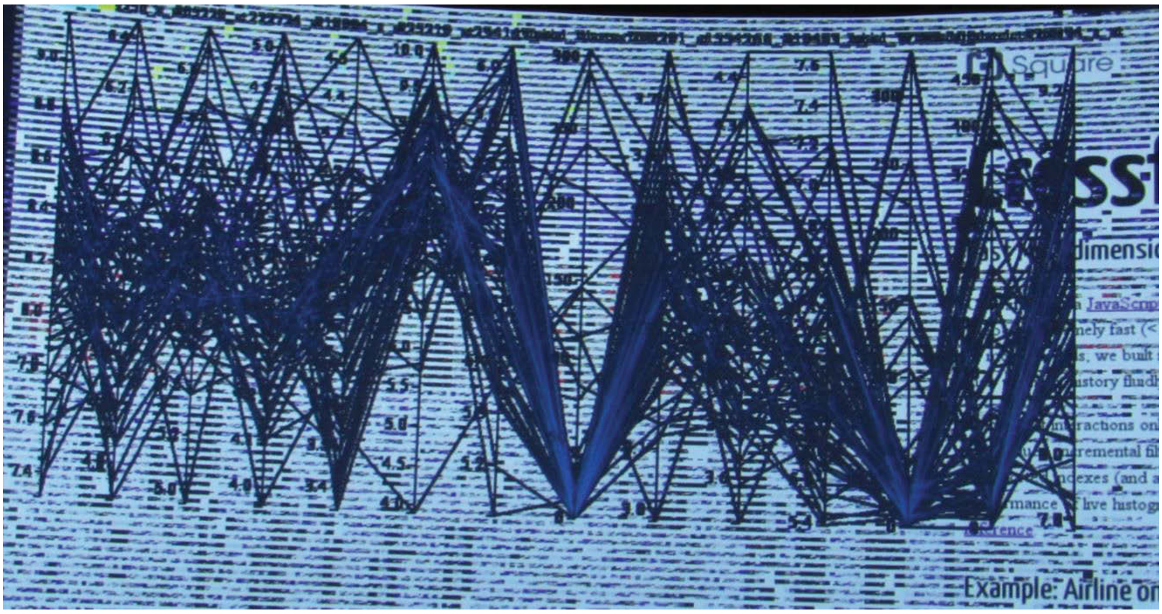


Figure U - Basic Parallel Coordinates with Transparent Background

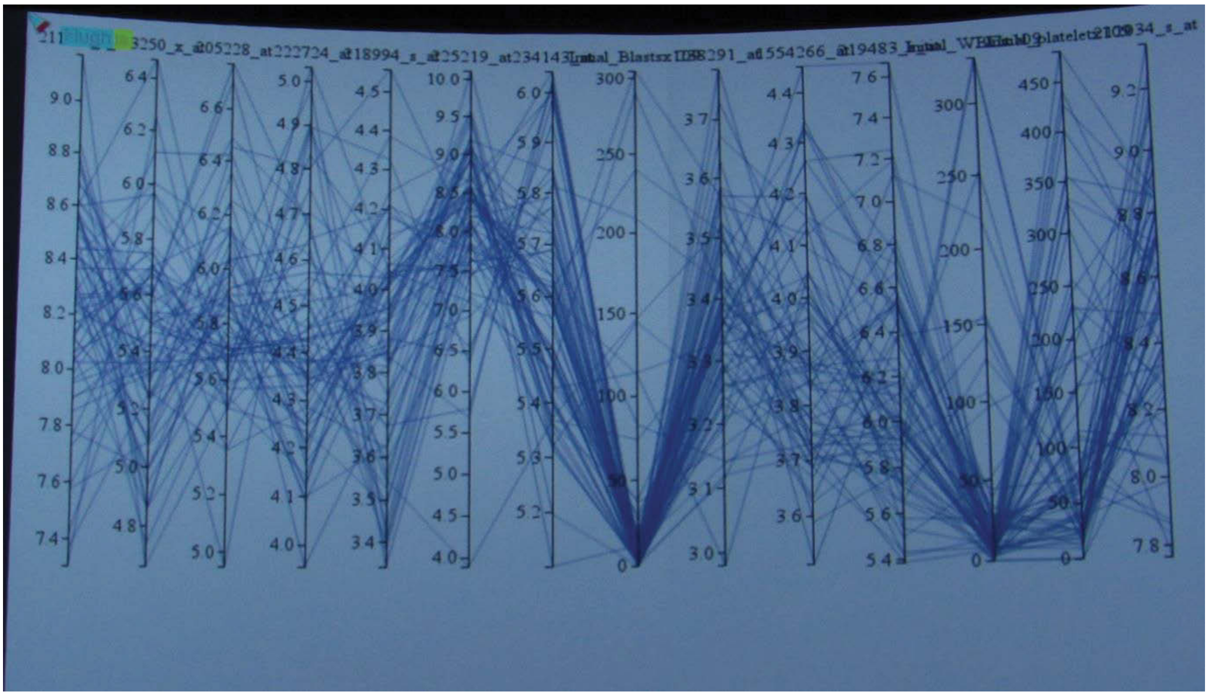


Figure V - Basic Parallel Coordinates with White Background

The cause for why the heatmap and network diagram plots not loading remains unknown. It may be that the JavaScript files are not being read correctly. It is possible that the WebView module in Omegalib is reading less than or greater than signs in the embedded JavaScript code as HTML tags. This would cause some aspects of the page to break, while retaining others. If the problem does occur within the WebView module, this might be fixed in a future update of Omegalib or Awesomium.

4.2 The SVG Parser Solution: a Work in Progress

4.2.1 Parallel Coordinates in the Data Arena

Parallel coordinates in the Data Arena worked extremely well using the SVG Parser (see Figure W). As the parser generates 3D elements in a 3D space, 3D glasses must be used to view the graph. If you do not use the 3D glasses the image will appear as in Figure X. Each of the lines are clear and the dimensions are spaced correctly (see Figures W and Y). The method has potential to be expanded in ways that are unavailable to the WebView solution. 3D elements could be interspersed with the 2D image. For example, legends and a zoom view could be placed on a closer plane to the viewer. Also, this graph could work well in combination with other graphs on the 3D plane.

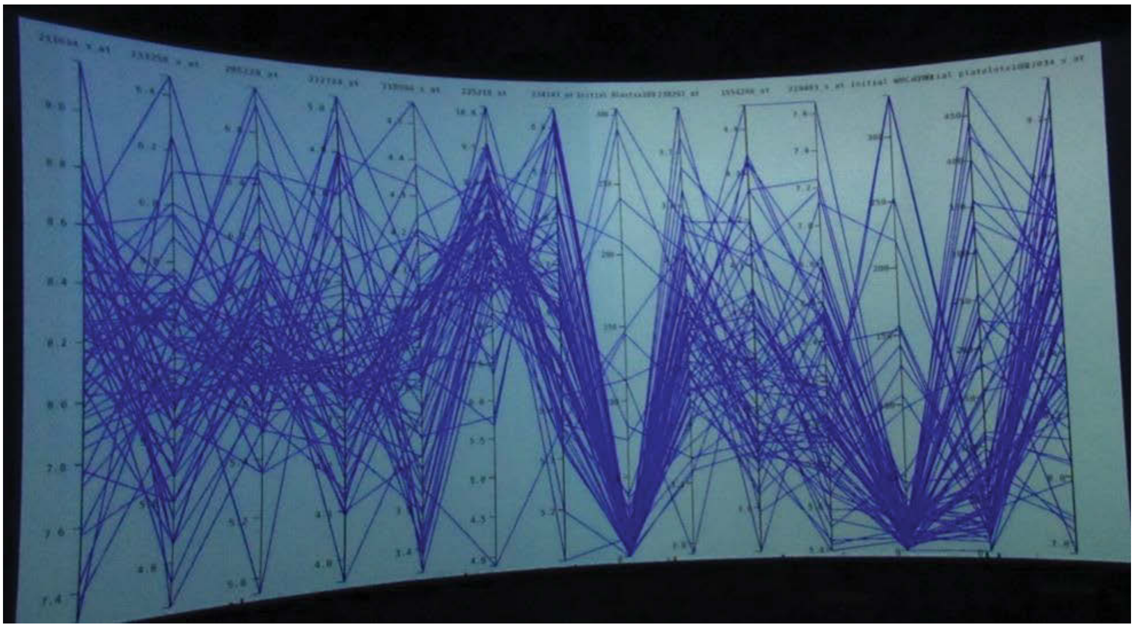


Figure W - SVG Parsed Parallel Coordinates with 3D Glasses

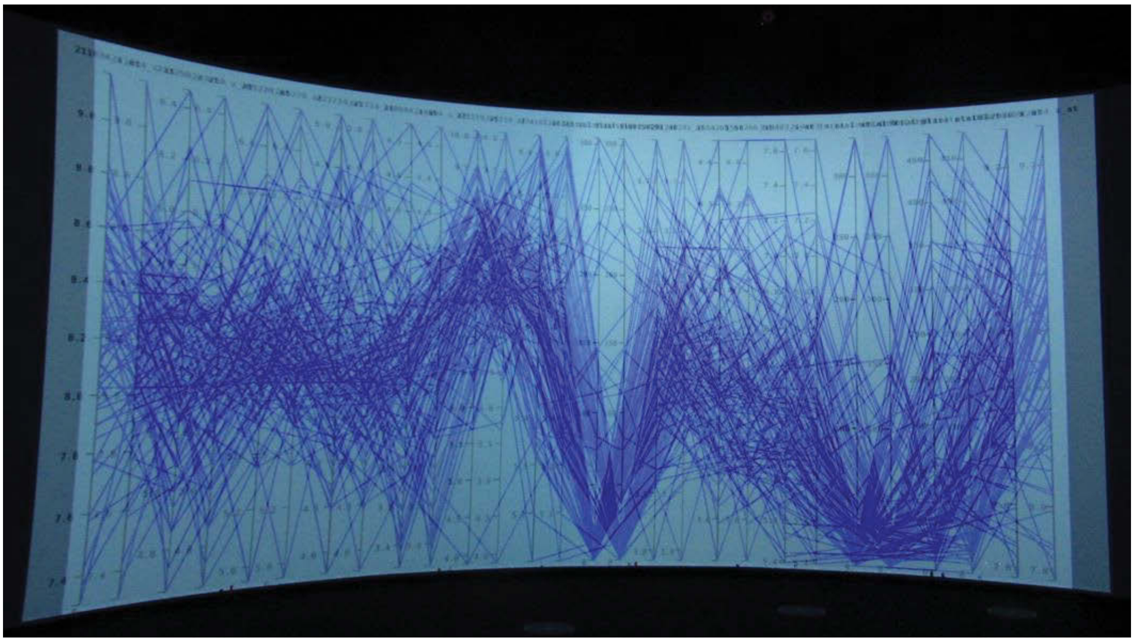


Figure X - SVG Parsed Parallel Coordinates without 3D Glasses

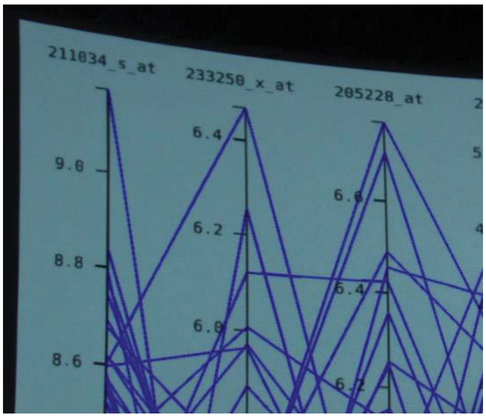


Figure Y - Close-up of SVG Parsed Parallel Coordinates

4.2.2 Test Report

The SVG Parser solution passed all the tests. Each of the test case's pass/fail status as well as a comment on the test are presented in the following Tables.

Table 17 - Test Case 07 Results

Test Case Number	07
Test Case Description	Be able to load and run the SVG Parser solution python script in the Data Arena
Pass/Fail	Passed
Comment	The SVG is able to load in the Data Arena and retrieve the local SVG image.

Table 18 - Test Case 08 Results

Test Case Number	08
Test Case Description	Render Paths
Pass/Fail	Passed
Comment	All paths are rendered with the correct grouping and positioning. The lines meet the axes in the correct spots.

Table 19 - Test Case 09 Results

Test Case Number	09
Test Case Description	Render Axes
Pass/Fail	Passed
Comment	All axes are rendered. Dimension headers and tick scale text is readable. The axes do not overwhelm the lines.

General Comments:

This solution seems to work well in the given circumstances. It may be advisable to move the camera slightly further back to get a better view of the plot. There is also a slight jagged edge to the diagonal lines. As this is using the Omegalib line and line set objects it is unclear how this can be solved using the Python scripts. As with the rendering of the WebPages this may be improved as the development of the Data Arena Progresses. The solution of parsing the SVG into Omegalib objects seems promising, even if the code will need to be updated often to adapt to new SVG plots. The high processing speeds of the Data Arena reads and renders the image quickly. Even if there are a multitude of lines it should be able to scale well when it comes to processing speed. It may be possible that there is a limit on the number of objects in a scene. However, this can be explored.

One concern is that the current solution is very dependent on the classes of elements. This was caused by not being able to access the parents of the current element that is being processed. If a different XML parser was used that had access to the parents it would make the code more adaptable to other SVG images.

The next steps would be including curves in the paths, reading the styles of an element and adapting the Omegalib object textures and attributes to that style. As the development of this method progresses so should its quality and adaptability.

5. Conclusion

Visualisation is a powerful tool in data analysis and may be useful in the analysis of multivariate high dimensional data. Virtual environments are 3D computer generated environments that simulate sensory information. One kind of virtual environment is CAVE. CAVE is a beneficial tool to applications that need immersion or presence. At this stage its application has been limited to 3D models, spatial studies, and simulation. There is little evidence to support its value in analysing complex data over other graphical applications. However, some aspects of immersion such as field of view and high resolution have been found to facilitate data-analytic tasks, and HREs, which incorporate features of ultra-high resolution display walls, may have increased value over CAVE for the analysis of complex data.

Acute lymphoblastic leukaemia is the most common form of childhood cancer. Medical management requires accurate risk stratification. Current risk profiles can be refined with the use of gene expression data to identify new subtypes of acute lymphoblastic leukaemia, further our understanding of the disease process and work towards personalised treatment. However, as the acute lymphoblastic leukaemia data is a high-dimensional/low observation, it is difficult to analyse using traditional methods.

The main barriers to both CAVEs and HREs have been cost and time-allocation. Visualising acute lymphoblastic leukaemia biomedical data in the Data Arena, a CAVE-like system with HRE qualities, could be useful for visualisation of high-dimensional data due to its display size and resolution. There are many visualisation libraries that could be integrated into the Data Arena for visualisation of the acute lymphoblastic leukaemia data set. D3 was chosen because of its compatibility with other web languages, performance, available learning resources, and previous research experience of acute lymphoblastic leukaemia investigators. There are various multivariate data visualisations that could be used to explore the relationships within the acute lymphoblastic leukaemia data.

The parallel coordinates method of presenting multiple dimensions has been shown to be useful because of its ability to find patterns and exceptions. Therefore it would seem important for the acute lymphoblastic leukaemia project clinical, biomedical and data analysis researchers to develop the integration of the D3 library for parallel coordinates in the Data Arena.

Omegalib is a software framework for HRE that is used in the Data Arena. Two approaches were used to develop the proposed solutions. The WebView solution developed script that could run pages using D3 library into the HRE, and then built the parallel coordinates. For the SVG Parser solution the parallel coordinates were first developed in D3 then integrated that part effectively into the HRE. The task of developing parallel coordinates in the Data Arena using D3 was split into multiple features. For example, for the WebView solution one feature was basic parallel coordinates and another feature was the ability to drag and reorder axes. The features that were developed in the development iteration were then presented for evaluation. If the features are accepted, then the next features can be developed for another evaluation. Otherwise, any features that were not accepted were modified based on feedback and re-presented for evaluation. Features of the task were ranked on necessity of the feature. Necessity was determined on whether the feature is needed for the success of the project, whether another necessary feature is dependent on the delivery of this feature, and discussion with the stakeholders. Validation of the solution included evaluation based test cases built to test the various features of each of the plots in the Data Arena and comparing them to the plots in a browser as the baseline.

The two solutions tested well in this evaluation, fulfilling the first objective of the thesis. Parallel coordinates plots were clear and able to load. The interactive elements that were part of the WebView solution were interactive in the Data Arena. Although the colours of the lines could be truer in hue to the original colours in the browser overall the parallel coordinates webpage in the Data Arena was very similar to the same parallel coordinate webpage in a browser.

However, neither the solutions are at a point where they are able to fulfil the second objective. The WebView solution was unable to view all the proposed visualisations when testing its ability to adapt to other visualisations. The majority of the failures were caused by the WebView solution not handling transparent backgrounds well. The previous images on the screen could be seen if the background was transparent. Work will need to be done to the WebView module, or only visualisation webpages with opaque backgrounds can be used in the Data Arena. There seemed to be another problem with two of the visualisations, likely to be caused by a misreading of JavaScript code by the WebView Omegalib module. It is out of the scope at this point in time to resolve this problem, however it could be considered for future projects. The SVG Parser solution has the potential to be adaptable with other visualisations in the SVG format; however, the code would have to be further developed.

Overall, the project demonstrated there is potential for 2D visualisations such as parallel coordinates in the Data Arena. As parallel coordinates have shown to be possible with high quality visualisation, the acute lymphoblastic leukaemia project has new avenues for visualisation of acute lymphoblastic leukaemia data. With the additional code and experience with the Data Arena that was explored in this thesis, other researchers have opportunities to present and analyse their data in the Data Arena. An unexpected benefit of the project was the feedback that was given to the Data Arena developers on their beta virtual machine and the process of developing code for the Data Arena.

5.1 Future work

There are several opportunities for future work in both the WebView solution and the SVG Parser Solution. The next step in the exploration of the WebView solution is to do an empirical study to test the usefulness of view parallel coordinates in the Data Arena compared to a personal computer. A top-level plan for an empirical study is presented in Appendix 7.2. Another area of future work is to work with Omegalib and the WebView module to make it more adaptable to other plots and webpages. By improving the underlying technology the uses and quality of visualisations will improve. This is evident by the development progress of the Data Arena mitigating the previously present aliasing issue when using WebView.

The SVG Parser has great potential for future work and development within the Data Arena. By expanding the code of the work in progress proof of concept, a fully fledged SVG Parser could be developed to translate the SVG into Omegalib objects. This has the added benefit of further advancement as the 3D areas of the Data Arena progress. Increasingly there is more focus on the 3D features of HREs, so by attaching the progress of the 2D visualisations to the 3D space speed of development should increase. Future researchers could also experiment with the SVG Parser solution by using a DOM parser rather than an XML parser. It may be more versatile as the SVG and styles could be extracted from an HTML document with other elements. Also, the parents of elements should be able to be retrieved, which was an issue in the XML parser that was used. Much of the current code could be moved across for this method and the script in general may become cleaner and more readable. The if statements could also be changed to a dictionary reference to functions, making a more readable and probably less computationally expensive method of parsing the SVG elements to Omegalib objects. The SVG Parser solution could also undertake an empirical study regarding the perceived benefits of the Data Arena over personal computers. The same plan that was proposed for the WebView solution is also suggested for the SVG Parser solution (see Appendix 7.2).

Although the SVG Parser has more opportunities and may result in a more adaptable solution than the WebView solution, its current lack of interactivity is a disadvantage. It is possible that the SVG Parser may progress to have interactive features. However, the role and relative importance of interactivity could also be evaluated in a head-to-head empirical study that compares the WebView solution to the SVG Parser solution, if both solutions are shown in studies (as proposed in Appendix 7.2) to perform better in a Data Arena than on a personal computer. The design of a head-to-head WebView solution versus SVG Parser solution empirical study would also be similar to that proposed in Appendix 7.2. However, it is recommended that static visualisations be used initially before moving to interactive visualisations.

6. References

Akkiraju, N., Edelsbrunner, H., Fu, P. & Qian, J. 1996, 'Viewing geometric protein structures from inside a CAVE', *IEEE Computer Graphics and Applications*, vol. 16, no. 4, pp. 58-61.

Anaissi, A., Kennedy, P.J., Goyal, M. & Catchpoole, R.D. 2013, 'A balanced iterative random forest for gene selection from microarray data', *BMC Bioinformatics*, vol. 14, viewed 30 May 2015, <http://www.biomedcentral.com/1471-2105/14/261>.

Beyer, K., Goldstein, J., Ramakrishnan, R. & Shaft, U. 1999, 'When is "Nearest Neighbor" meaningful?', in Beeri, C. & Buneman, P. (eds), *Database theory -ICDT'99*, Springer, Berlin, pp. 217-35.

Biondi, A. & Cazzaniga, G. 2013, 'Novel clinical trials for pediatric leukemias: lessons learned from genomic analyses', *ASH Education Program Book*, vol. 2013, pp. 612-9.

Bl.ocks.org 2012a, *Multiple Area Charts*, viewed 18 November 2015, <http://bl.ocks.org/tylercraft/3630001>.

Bl.ocks.org 2012b, *Testing Pie Chart*, viewed 18 November 2015, <http://bl.ocks.org/diethardsteiner/raw/3287802/>.

Bl.ocks.org 2014, *Days-Hours Heatmap*, viewed 18 November 2015, <http://bl.ocks.org/oyyd/859fafc8122977a3afd6>.

Bostock, M. 2011a, *d3.layout.tree*, mbostock.github.io, viewed 18 November 2015, <https://mbostock.github.io/d3/talk/20111018/tree.html>.

Bostock, M. 2011b, *Edgar Anderson's Iris dataset scatterplot matrix*, mbostock.github.io, viewed 18 November 2015, <http://mbostock.github.io/d3/talk/20111116/iris-splom.html>.

Bostock, M. 2011c, *Parallel Coordinates*, Bl.ocks.org, viewed 20 November 2015, <http://bl.ocks.org/mbostock/1341021>.

Bostock, M. 2011d, *Small Multiples*, Bl.ocks.org, viewed 18 November 2015, <http://bl.ocks.org/mbostock/1157787>.

Bostock, M. 2012a, *Force-Directed Graph*, Bl.ocks.org, viewed 18 November 2015, <http://bl.ocks.org/mbostock/raw/4062045/>.

Bostock, M. 2012b, *Les Misérables Co-occurrence*, Bost.ocks.org, viewed 18 November 2015, <http://bost.ocks.org/mike/miserables/>.

Bostock, M., Ogievestky, V. & Heer, J. 2011, 'D3: Data-Driven Documents', *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301-9.

Bowman, D.A. & McMahan, R.P. 2007, 'Virtual reality: how much immersion is enough?', *Computer*, vol. 40, no. 7, pp. 36-46.

Cancer Australia 2014, *Acute lymphoblastic leukaemia statistics*, viewed 26 April 2015, <http://canceraustralia.gov.au/affected-cancer/cancer-types/leukaemia/acute-lymphoblastic-leukaemia-statistics>.

Castillo, P. 2014, *Mastering D3.js*, Packt Publishing, Birmingham, UK.

Catchpoole, D.R., Kennedy, P., Skillicorn, D.B. & Simoff, S. 2010, 'The curse of dimensionality: a blessing of personalised medicine', *Journal of Clinical Oncology*, vol. 28, no. 34, pp. 723-4.

Chang, K. 2012, *Parallel Coordinates*, GitHub, viewed 6 December 2015, <https://github.com/syntagmatic/parallel-coordinates>.

Chang, K. 2015, *Parallel Coordinates*, syntagmatic.github.io, viewed 22 November 2015, <http://syntagmatic.github.io/parallel-coordinates/>.

Cheng, K.-O., Law, N.-F., Siu, W.-C. & Liew, A.W.-C. 2008, 'Identification of coherent patterns in gene expression data using an efficient biclustering algorithm and parallel coordinates visualization', *BMC Bioinformatics*, vol. 9, no.210, viewed 29 May 2015, < http://www.biomedcentral.com/1471-2105/9/210/>.

Cock, P.J.A., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A., Frieberg, I., Hamelryck, T., Kauff, F., Wilczynski, B. & de Hoon, M.J.L. 2009, 'BioPython: freely available Python tools for computational molecular biology and bioinformatics', *Bioinformatics*, vol. 25, no. 11, pp. 1422-3.

Cook, P. 2015, *UK Temperature 1910-2014*, charts.animateddata.co.uk, viewed 18 November 2015, <http://charts.animateddata.co.uk/uktemp/>.

Creagh, H. 2003, 'CAVE automatic virtual environment', *Proceedings: Electrical Insulation Conference and Electrical Manufacturing & Coil Winding Technology Conference*, IEEE, Indianapolis, IN, pp. 499-504.

Cruz-Neira, C., Leigh, J., Michael, P., Barnes, C., Cohen, S.M., Das, S., Engelmann, R., Hudson, R., Roy, T., Siegel, L., Vasilakis, C., DeFanti, T.A. & Sandin, D.J. 1993, 'Scientists in wonderland: a report on visualization applications in the CAVE virtual reality environment', *IEEE 1993 Symposium on Research Frontiers in Virtual Reality*, IEEE, San Jose, CA, pp. 59-66.

Cruz-Neira, C., Sandin, D.J. & DeFanti, T.A. 1993, 'Surround-screen projection-based virtual reality: the design and implementation of CAVE', *Proceedings of the 20th*

Annual Conference on Computer Graphics and Interactive Techniques, ACM, Anaheim, CA, pp. 135-42.

Dell'Unto, N., Leander, A.M., Ferdani, D., Dellepiane, M., Callieri, M. & Lindgren, S. 2013, 'Digital reconstructions and visualization in archaeology: case-study drawn from the work of the Swedish Pompeii Project," *Digital Heritage International Congress (DigitalHeritage)*, IEEE, Marseille, pp. 621-8.

Dewar, M. 2012, *Getting started with D3*, O'Reilly, Sebastopol, CA.

Díaz-Uríarte, R. & de Andrés, S.A. 2006, 'Gene selection of microarray data using random forest', *BMC Bioinformatics*, vol. 7, viewed 25 April 2015, <http://www.biomedcentral.com/1471-2105/7/3>.

Febretti, A., Nishimoto, A., Mateevitsi, V., Renambot, L., Johnson, A. & Leigh, J. 2014, 'Omegalib: a multi-view application framework for hybrid reality display environments', Virtual Reality, IEEE, Minneapolis, MN, pp. 9-14.

Few, S.C. 2009, *Now you see it*, Analytic Press, Oakland, CA.

GitHub 2015, *Gallery*, viewed 6 December 2015, <https://github.com/mbostock/d3/wiki/Gallery>.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. & Lander E.S. 1999, 'Molecular classification of cancer: class discovery and class prediction by gene expression monitoring', *Science*, vol. 286, 531-7.

Gomez, J., Garcia, L., Salazar, G., Villaveces, J., Gore, S., Garcia, A., Martin, M., Launay, G., Alcantara, R., del-Toro, N., Dumousseau, M., Orchard, S., Velankar, S., Hermjakob, H., Zong, C., Ping, P., Corpas, M. & Jimenez, R. 2013, 'BioJS: an open source JavaScript framework for biological data visualization', *Bioinformatics*, vol. 29, no. 8, pp. 1103-1104.

Hoheisel, J.D. 2006, 'Microarray technology: beyond transcript profiling and genotype analysis', *Nature Reviews Genetics*, vol. 7, pp. 200-10.

Inaba, H., Greaves, M. & Mullighan, C.G. 2013, 'Acute lymphoblastic leukemia', *The Lancet*, vol. 381, no. 9881, pp. 1943-55.

Inselberg, A. & Dimsdale, B. 1990, 'Parallel coordinates: a tool for visualising multidimensional geometry', *Proceedings of the First IEEE Conference on Visualization, 1990. Visualization '90*, IEEE, San Francisco, CA, pp.361-78.

Ioannidis, J.P.A 2005, 'Why most published research findings are false', PLOS Medicine vol. 2, no. 8, viewed 1 June 2015,

<http://www.plosmedicine.org/article/fetchObject.action?uri=info:doi/10.1371/journal.pmed.0020124&representation=PDF>.

Johansson, J., Ljung, P., Jern, M. & Cooper, M. 2005, 'Revealing structure with clustered parallel coordinate displays', *IEEE Symposium on Information Visualization*, IEEE, Minneapolis, MN, pp. 125-32.

Kalifa, N.H., Nguyen, Q.V., Simoff, S. & Catchpoole, D. 2015, 'A visualization system for analyzing biomedical and genomic data sets using Unity3D platform', *Proceedings of the 8th Australian Workshop on Health Informatics and Knowledge Management*, Australian Computer Society, Sydney, Australia, pp. 47-53.

Kallioniemi, O.P., Kononen, J. & Sauter, G. 2012, 'Introducing tissue microarrays to molecular pathology', *Clinical Chemistry*, vol. 58, no. 12, pp. 1717-8.

Kallioniemi, O.P., Wagner, U., Kononen, J. & Sauter, G. 2001, 'Tissue microarray technology for high-throughput molecular profiling of cancer', *Human Molecular Genetics*, vol. 10, no. 7, pp. 657-62.

Kennedy, P.J., Simoff, S.J., Catchpoole, D.R., Skillicorn, D.B., Ubaudi, F. & Al-Oqaily, A. 2008, 'Intergrative visual data mining of biomedical data: investigating cases in chronic fatigue syndrome and acute lymphoblastic leukaemia', in Simoff, S.J., Böhlen, M.H. & Mazeika, A. (eds), *Visual Data Mining*, Springer, Berlin, pp. 367-88.

Kotthoff, L. 2012, *Happy Planet Index*, Larsko.org, viewed 18 November 2015, <http://www.larsko.org/v/hpi/>.

Kriegel, H.-P., Kröger, P. & Zimek, A. 2009, 'Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering', *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 1, pp. 1-58.

Leukaemia Foundation 2015, *Acute lymphoblastic leukaemia (acute lymphoblastic leukaemia)*, viewed 26 April 2015, <http://www.leukaemia.org.au/blood-cancers/leukaemias/acute-lymphoblastic-leukaemia-all>.

Manjrekar, S., Sandilya, S., Bhosale, D., Kanchi, S., Pitkar, A. & Gondhalekar, M. 2014, 'CAVE: an emerging immersive technology – a review', *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, IEEE, Cambridge, UK, pp. 131-6.

Merriam-Webster n.d., *Visualization*, viewed 25 May 2015, <http://www.merriam-webster.com/dictionary/visualization>.

Moustafa, R.E. 2011, 'Parallel coordinate and parallel coordinate density plots', *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 3, no. 2, pp. 134–148.

Murray, S. 2013, *Interactive data visualization for the web*, O'Reilly Media, Sebastopol, CA.

Nan, X., Zhang, Z., Zhang, N., Guo, F., He, Y. & Guan, L. 2014, ‘vDesign: a CAVE-based virtual design environment using hand interactions’, *Journal on Multimodal User Interfaces*, vol. 8, no. 4, pp. 367-79.

Ng, A. 2015, *Personalities of Star Wars Fans*, annalyn-ng.com, viewed 18 November 2015, <http://annalyn-ng.com/starwarsb5/chart.html>.

Nguyen, Q.V., Gleeson, A., Ho, N., Huang, M.L., Simoff, S. & Catchpoole, D. 2011, ‘Visual analytics of clinical and genetic datasets of acute lymphoblastic leukaemia’, in Lu, B.-L., Zhang, L., & Kwok, J. (eds), *ICONIP 2011*, Springer, Berlin, pp. 113-20.

Peat, J., Mellis, C., Williams, K. & Xuan, W. 2001, *Health science research*, Allen & Unwin, Crows Nest, NSW.

Pui, C.-H., Mullighan, C.G., Evans, W.E. & Relling, M.V. 2012, ‘Pediatric acute lymphoblastic leukemia: where are we going and how do we get there?’ *Blood*, vol. 120, no. 6, pp. 1165-74.

Rao, R. & Card, S.K. 1994, ‘The table lens: merging graphical and symbolic representations in an interactive focus+context visualization for tabular information’, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, Boston, MA, pp. 318-22.

Reda, K., Febretti, A., Knoll, A., Aurisano, J., Leigh, J., Johnson, A., Papka, M.E. & Hereld, M. 2013, ‘Visualizing Large, Heterogeneous Data in Hybrid-Reality Environments’, *Computer Graphics and Applications*, vol. 33, no. 4, pp. 38-48.

Root, B. 2015, *Interactive applications using matplotlib*, Packt Publishing Limited, Birmingham.

Ross, M.E., Zhou, X., Song, G., Shurtleff, S.A., Girtman, K., Williams, W.K., Liu, H.-C., Mahfouz, R., Raimondi, S.C., Lenny, N. & Downing, J.R. 2003, ‘Classification of pediatric acute lymphoblastic leukemia by gene expression profiling’, *Blood*, vol. 102, no. 8, pp. 2951-9.

Siirtola, H., Laivo, T., Heimonen, T. & Räihä, K.-J. 2009, ‘Visual perception of parallel coordinates visualizations’, *13th International Conference Information Visualisation*, IEEE, Barcelona, pp. 3-9.

Simon, R. & Altman, D.G. 1994, ‘Statistical aspects of prognostic factor studies in oncology’, *British Journal of Cancer*, vol. 69, pp. 979-85

Square Inc. 2012, *Crossfilter*, viewed 18 November 2015, <http://square.github.io/crossfilter/>.

The Qt Company 2015, *Qt - Qt for Application Development*, viewed 21 November 2015, <http://www.qt.io/application-development/>.

Tran, J., Nguyen, Q.V. & Simoff, S. 2014, 'Intelliviz- a tool for visualizing social networks with hashtags', in Bebis, G., Boyle, R., Parvin, B., Koracin, D., McMahan, R., Jerald, J., Zhang, H., Drucker, S.M., Kambhamettu, C., El Choubassi, M., Deng, Z. & Carlson, M. (eds), *Advances in Visual Computing*, Springer International Publishing, Cham, Switzerland, pp. 894-903.

Tricaud, S., Nance, K & Saadé, P. 2011, 'Visualizing network activity using parallel coordinates', *Proceedings of the 44th Hawaii International Conference on System Sciences*, IEEE, Kauai, HI, pp. 1-8.

Tufte, E.R. 2002, *The visual display of quantitative information*, Graphics Press, Cheshire, CT.

Tukey, J.W. 1974, 'Mathematics and the picturing of data', *Proceedings of the International Congress of Mathematicians*, Canadian Mathematical Congress, Vancouver, pp. 523-31.

University of Technology Sydney 2014, *Data Arena Prototype*, viewed 6 April 2015, <http://www.uts.edu.au/future-students/design-architecture-and-building/facilities/research-facilities/data-arena-prototype>.

Vaingast, S. 2009, *Beginning Python visualization*, Apress, Berkeley, CA.

van Rossum, G., Warsaw, B. & Coghlan, N. 2013, *Style Guide for Python Code*, Python.org, viewed 12 June 2015, <https://www.python.org/dev/peps/pep-0008/>.

Wang, R., Perez-Riverol, Y., Hermjakob, H. & Vizcaíno, J.A. 2015, 'Open source libraries and frameworks for biological data visualisation: a guide for developers', *Proteomics*, vol. 15, pp. 1356-74.

Who Do They Serve 2015, *Overview*, viewed 18 November 2015, <http://www.whodotheyserve.com/>.

Whyte, A., Jervis, B., Pupius, D., Arvidsson, E., Schneider, F. & Walker, R. n.d., *Google JavaScript Style Guide*, google-styleguide.googlecode.com, viewed 12 June 2015, <https://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml>.

Williams, G. 2011, *Data mining with Rattle and R*, Springer, New York.

Yang, X., Deines, E., Lauer, C. & Aurich, J.C. 2011, 'A human-centered virtual factory', *2011 International Conference on Management Science and Industrial Engineering (MSIE)*, IEEE, Harbin, pp. 1138-42.

Yeoh, E.-J, Ross, M.E., Shurtleff, S.A., Williams, W.K., Patel, D., Mahfouz, R., Behm, F.G., Girtman, K., Raimondi, S.C., Relling, M.V., Patel, A., Cheng, C., Campana, D., Wilking, D., Zhou, X., Li, J., Liu, H., Pui, C.-H., Evans, W.E., Naeve, C., Wong, L. & Downing, J.R. 2002, ‘Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling’, *Cancer Cell*, vol. 1, pp. 133-43.

7. Appendix

7.1 Source Code

As the Data Arena will change as development progresses, it is recommended that contact be first initiated with the Data Arena developers to enquire about changes that may need to be applied to the code presented below before implementation in the Data Arena.

7.1.1 WebView Solution

Omegalib Code

```
from webView import *
from math import *
from cyclops import *

#width = 2648 # Original testing width
width = 3870 # For more features width
#width = 4096 # For more features width
height = 900

cam = getDefaultCamera()
cam.setEyeSeparation(0)

ww = None

ui = UiModule.createAndInitialize()
uiroot = ui.getUi()

cont = Container.create(ContainerLayout.LayoutFree, ui.getUi())

myNode = SceneNode.create("myNode")

c3d = cont.get3dSettings()
c3d.enable3d = True
c3d.position = Vector3(0.04, 2.5, -2)
c3d.scale = 0.001
c3d.node = myNode

if(isMaster()):
    ww = WebView.create(width, height)
    ww2 = WebView.create(width, height)
    ww.loadUrl("file:///da/proj/sam/PCWeb/pcPlot.html") # Change this to where the
pcPlot.html is stored locally

     # Other Visualisations Tested
    ##ww.loadUrl("http://bl.ocks.org/mbostock/raw/4062045/")
    ##ww.loadUrl("http://www.whodotheyserve.com")
    ##ww.loadUrl("http://mbostock.github.io/d3/talk/20111018/tree.html") # Tree
    ##Diagram (Bostock 2011)
    ##ww.loadUrl("http://charts.animateddata.co.uk/uktemp/") # Scatterplot/Line
    ##Plot
    ##ww.loadUrl("http://mbostock.github.io/d3/talk/20111116/iris-splom.html") #
    ##Visual crosstabs, scatterplot matrix with different scales (Bostock 2011)
    ##ww.loadUrl("http://joshua-gould.github.io/public/examples/miserables/") #
    ##Heatmap (Bostock 2012)
```

```
    ##ww.loadUrl("http://bl.ocks.org/oyyd/859fafc8122977a3afd6") # Heatmap
    ##(Bl.ocks.org 2014)
    ##ww.loadUrl("http://www.larsko.org/v/hpi/") # Glyphs/Chernoff Faces (Kotthoff
    ##2012)
    ##ww.loadUrl("http://bl.ocks.org/mbostock/1157787") # Small multiples (Bostock
    ##2011)
    ##ww.loadUrl("http://bl.ocks.org/tylercraft/3630001") # Small multiples
    ##(Bl.ocks.org 2012)
    ##ww.loadUrl("http://annalyn-ng.com/starwarsb5/chart.html ") # Star Plots (Ng
    ##2015)
    ##ww.loadUrl("http://bl.ocks.org/diethardsteiner/raw/3287802/") # Concurrent
    ##Views (Bl.ocks.org 2012)
    ##ww.loadUrl("http://square.github.io/crossfilter/") # Concurrent Views
    ##(Square Inc.  2012)
    ##ww.loadUrl("file:///da/proj/sam/crossfilter/Crossfilter.htm") # Concurrent
    ##Views (Square Inc.  2012) # this works when setting background to white!
    ##ww.loadUrl("file:///da/proj/sam/PCWeb/SVGbasic1.html") # Concurrent Views

    frame = WebFrame.create(cont)
    frame.setView(ww)
else:
    ww = PixelData.create(width, height, PixelFormat.FormatRgba)
    frame = Image.create(cont)
    frame.setDestRect(0, 0, width + 12, height + 12)
    frame.setData(ww)

ImageBroadcastModule.instance().addChannel(ww, "webpage", ImageFormat.FormatNone)

# Multiple Users
cursorImg = loadImage('/da/sw/omegalib/myCursor.png')
cursorClickImg = loadImage('/da/sw/omegalib/myCursor_click.png')
currentUser = 0

cursors = []
labels = []

names = ["A", # Can change to the name of person controlling
    "B",
    "C",
    "D",
    "E",]
cols = ['#FF0000',
    '#00FF00',
    '#0000FF',
    '#FFFF00',
    '#00FFFF',]

for i in range(5):
    cursor = Image.create(cont)
    label = Label.create(cont)
    label.setText(names[i])
    label.setFont('fonts/arial.ttf 18')
    label.setColor(Color('white'))
    label.setPosition(Vector2(32, 12))
    label.setFillEnabled(True)
    label.setFillColor(Color(cols[i]))

    if i == 0:
           cursor.setSize(Vector2(32, 32))
           cursor.setData(cursorImg)
    else:
```

```
            cursor.setData(loadImage('/da/sw/omegalib/myCursor_' + str(i + 1) +
'.png'))
            cursor.setSize(Vector2(24, 24))
        cursors.append(cursor)
        labels.append(label)

    screen = getDisplayPixelSize()

    prevOrientations = [[Quaternion()]] * len(names)

    prevDiffAmt = 0.0

    def diff(q1, q2):
        return ((abs(q2.w) + abs(q2.x) + abs(q2.y) + abs(q2.z)) - (abs(q1.w) +
abs(q1.x) + abs(q1.y) + abs(q1.z)))

    def onEvent():

        global currentUser, cursors, labels
        global cursorClickImg, cursorImg
        global prevOrientations, prevDiffAmt

        e = getEvent()

        if e.getServiceType() == ServiceType.Mocap:
                if e.getExtraDataItems() >= 2:
                        point = Vector2(e.getExtraDataInt(0), e.getExtraDataInt(1))
                        if e.getUserId() > len(cursors):
                                return

                        po = Quaternion()

                        for a in prevOrientations[e.getUserId() - 1]:
                                po.w += a.w
                                po.x += a.x
                                po.y += a.y
                                po.z += a.z

                        po.w /= 1.0 * len(prevOrientations[e.getUserId() - 1])
                        po.x /= 1.0 * len(prevOrientations[e.getUserId() - 1])
                        po.y /= 1.0 * len(prevOrientations[e.getUserId() - 1])
                        po.z /= 1.0 * len(prevOrientations[e.getUserId() - 1])

                        aa = e.getOrientation()
                        diffAmt = diff(aa, po)

                        diffChange = abs(diffAmt - prevDiffAmt)

                        # This will change in future development of the Data Arena and
applications,
                        # check with Data Arena developers to ensure you have the updated
code for
                        # controls
                        if diffChange < 0.075:
                                print "diff change:", diffChange
                                cursors[e.getUserId() - 1].setPosition(point)
                                labels[e.getUserId() - 1].setPosition(point + Vector2(32,
12))

                        prevDiffAmt = diffAmt

                        prevOrientations[e.getUserId() - 1].append(e.getOrientation())
```

```
                if len(prevOrientations[e.getUserId() - 1]) >= 4:
                    prevOrientations[e.getUserId() - 1].pop(0)

            if (e.getUserId() == 1):
                vec = e.getOrientation() * Vector3(0, 1, 0)

                if vec[1] < -0.6:
                    cursors[e.getUserId() - 1].setData(cursorClickImg)
                else:
                    cursors[e.getUserId() - 1].setData(cursorImg)


setEventFunction(onEvent)
```

Parallel Coordinate Webpage

```
<html>
  <head>
    <title>Parallel Coordinates</title>
    <link rel="stylesheet" type="text/css" href="parallel-coordinates-
master/d3.parcoords.css">
    <script src="parallel-coordinates-master/examples/lib/d3.min.js"></script>
    <script src="parallel-coordinates-master/d3.parcoords.js"></script>
    <script src="parallel-coordinates-master/examples/lib/sylvester.js"></script>
    <script src="merged.genes.js"></script>
  </head>
  <body style="background-color:white">
      <div style="overflow:auto">
        <div id="pc"  class="parcoords"
style="width:99%;height:80%;margin:5px;float:left"></div>
        <div id="legend" class="legend"
style="width:50%;height:5%;margin:10px;overflow:auto"></div>
      </div>
      <div id="genes" class="genes"
style="overflow:auto;width:99%;height:30%;margin:5px"></div>
  </body>
    <script type="text/javascript">
      var pcgender = ["Male", "Female", "Unspecified"];

      var pccolors = d3.scale.ordinal()
        .domain(pcgender)
       .range(['#f47a00','#00f47a','#7a00f4']);

      var colors = function (d) { return pccolors(d.Gender) };

      var headers = Object.keys(genes[0]);

      var legend = d3.select("#legend");

      var item1 = legend.append("div").style({overflow:"auto", "float":"left"});
      item1.append("div").style({"background-color":
pccolors(pcgender[0]),width:"10px",height:"10px","float":"left","margin":"5px"});
      item1.append("div").style({"float":"left"}).text(pcgender[0]);
      var item2 = legend.append("div").style({overflow:"auto", "float":"left"});
      item2.append("div").style({"background-color":
pccolors(pcgender[1]),width:"10px",height:"10px","float":"left","margin":"5px"});
      item2.append("div").style({"float":"left"}).text(pcgender[1]);
      var item3 = legend.append("div").style({overflow:"auto", "float":"left"});
      item3.append("div").style({"background-color":
pccolors(pcgender[2]),width:"10px",height:"10px","float":"left","margin":"5px"});
```

```
          item3.append("div").style({"float":"left"}).text(pcgender[2]);

          var pc = d3.parcoords()("#pc")
            .data(genes)
               .hideAxis(["group","Gender"])
            .alpha(0.3)
            .color(colors)
            .smoothness(0.07)
            .showControlPoints(false)
            .composite("darker")
            .margin({top:25,left:50,bottom:25,right:0})
            .render()
            .brushMode("1D-axes")
            .reorderable()
               .interactive();
          var table = d3.select("#genes").append("table");
          var thead = table.append("thead");
          var tbody = table.append("tbody");

          thead.append("tr")
            .selectAll("th")
            .data(headers)
            .enter()
            .append("th")
            .text(function(header){ return header; });
          var rows = tbody.selectAll("tr")
            .data(genes)
            .enter()
            .append("tr");
          var cells = rows.selectAll("td")
            .data(function(row) { return headers.map(function(header) { return {column:
header, value: row[header]}; }); })
            .enter()
            .append("td")
            .html(function(cell) { return cell.value; });
        </script>
        <!--  Parallel Coordinates Plugin for D3.js Licence
            Copyright (c) 2012, Kai Chang
            All rights reserved.

            Redistribution and use in source and binary forms, with or without
            modification, are permitted provided that the following conditions are
met:

            * Redistributions of source code must retain the above copyright notice,
this
                list of conditions and the following disclaimer.

            * Redistributions in binary form must reproduce the above copyright
notice,
                this list of conditions and the following disclaimer in the
documentation
                and/or other materials provided with the distribution.

            * The name Kai Chang may not be used to endorse or promote products
                derived from this software without specific prior written permission.

            THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS"
            AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
            IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE
```

```
            DISCLAIMED. IN NO EVENT SHALL MICHAEL BOSTOCK BE LIABLE FOR ANY DIRECT,
            INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING,
            BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE,
            DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY
            OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
            NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE,
            EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
        -->
    </html>
```

7.1.2 SVG Parser Solution

Omegalib Code

```
from math import *
from euclid import *
from omega import *
from cyclops import *
from omegaToolkit import *
from xml.etree.ElementTree import ElementTree

scene = getSceneManager()
distance = 0
camera = getDefaultCamera()
camera.setPosition(Vector3(0,0,500))

# Storing the position for nested groups
translateX = 0
translateY = 0
translateTickX = 0
translateTickY = 0
translateDimX = 0
translateDimY = 0
svgWidth = 0
svgHeight = 0

# Parsing the SVG
tree = ElementTree()
tree.parse("PCSVG.svg") # Change this path to the location of the SVG
elements = tree.iter()
for element in elements:
  if element.tag == "svg":
    svgWidth = float(element.get("width")) / 2
    svgHeight = float(element.get("height")) / 2
    plane = PlaneShape.create(svgWidth,svgHeight)
    plane.setPosition(Vector3(0,0,distance))
    plane.setEffect("colored -e white")
    plane.getMaterial().setDoubleFace(True)
  if element.tag == "line":
    x1 = 0
    y1 = 0
    x2 = 0
    y2 = 0
    if element.get("x1") is not None:
      x1 = float(element.get("x1")) / 2
    if element.get("y1") is not None:
      y1 = -float(element.get("y1")) / 2
```

```
        if element.get("y2") is not None:
          x2 = float(element.get("x2")) / 2
        if element.get("y2") is not None:
          y2 = -float(element.get("y2")) / 2
        lineSet = LineSet.create()
        line = lineSet.addLine()
        line.setStart(Vector3(x1,y1,0))
        line.setEnd(Vector3(x2,y2,0))
        line.setThickness(0.5)
        lineSet.setEffect("colored -e black")
        xPos = translateX + translateDimX + translateTickX - svgWidth / 2
        yPos = svgHeight / 2 - (translateY + translateDimY + translateTickY)
        lineSet.setPosition(Vector3(xPos,yPos,distance + 0.5))
      if element.tag == "circle":
        cx = float(element.get("cx")) / 2
        cy = float(element.get("cy")) / 2
        radius = float(element.get("r")) / 2
        circle = SphereShape.create(radius, 5)
        circle.setPosition(Vector3(cx, cy, distance))
        circle.setEffect("colored -e red")
      if element.tag == "rect":
        width = float(element.get("width"))
        height = float(element.get("height"))
        x = 0
        y = 0
        if element.get("x") is not None:
          x = float(element.get("x")) / 2
        if element.get("y") is not None:
          y = float(element.get("y")) / 2
        rect = PlaneShape.create(width, height)
        rect.setPosition(Vector3((x - svgWidth / 2 + width / 2),(y + svgHeight / 2 -
width / 2),(distance + 1)))
        rect.setEffect("colored -e red")
        rect.getMaterial().setDoubleFace(True)
      if element.tag == "g":
        if element.get("transform") is not None:
          transform = element.get("transform")
          translatestr = ""
          for index in range(10,len(transform) - 1):
            translatestr += transform[index]
          if "," in translatestr:
            translate = translatestr.split(",")
          else:
            translate = translatestr.split()
          if element.get("class") is not None:
            if element.get("class") == "dimension":
              translateDimX = float(translate[0]) / 2
              translateTickY = 0
              translateTickX = 0
            elif element.get("class") == "tick":
              translateTickX = float(translate[0]) / 2
          if element.get("class") is None:
            translateX = float(translate[0]) / 2
          if len(translate) == 2:
            if element.get("class") == "dimension":
              translateDimY = float(translate[1]) / 2
            elif element.get("class") == "tick":
              translateTickY = float(translate[1]) / 2
            else:
              translateY = float(translate[1]) / 2
      if element.tag == "path":
        if element.get("d") is not None:
```

```
          lineSet = LineSet.create()
          d = element.get("d")
          dList = []
          currentNo = ""
          currentType = 0

          for index in range(0, len(d)):
            if d[index] in ["M", "V", "L", "H"]:
              if currentType != 0:
                dList[currentType - 1].append(currentNo)
                currentNo = ""
              currentType += 1
              dList.append([d[index]])
            else:
              if d[index] == "," or d[index] == " ":
                dList[currentType - 1].append(currentNo)
                currentNo = ""
              else:
                currentNo += d[index]
          dList[currentType - 1].append(currentNo)
          startX = float(dList[0][1])
          startY = float(dList[0][2])
          for i in range(1, len(dList)):
            endX = startX
            endY = startY
            if dList[i][0] == "H":
              endX = float(dList[i][1])
            elif dList[i][0] == "V":
              endY = float(dList[i][1])
            elif dList[i][0] == "L":
              endX = float(dList[i][1])
              endY = float(dList[i][2])
            line = lineSet.addLine()
            line.setStart(Vector3(startX / 2, -startY / 2,0))
            line.setEnd(Vector3(endX / 2, -endY / 2,0))
            line.setThickness(0.5)
            startX = endX
            startY = endY
          if element.get("class") == "domain":
            lineSet.setPosition(Vector3(translateX + translateDimX - svgWidth / 2,
svgHeight / 2 - (translateY + translateDimY), distance + 0.5))
            lineSet.setEffect("colored -e black")
          else:
            lineSet.setPosition(Vector3(translateX - svgWidth / 2, svgHeight / 2 -
translateY, distance + 0.5))
            lineSet.setEffect("colored -e blue")
      if element.tag == "text":
        if element.text is not None:
          t = Text3D.create('fonts/arial.ttf', 4, element.text)
          xPos = translateX + translateDimX - svgWidth / 2
          yPos = svgHeight / 2 - (translateY + translateDimY)
          if element.get("text-anchor") is None:
            xPos += translateTickX - t.getBoundMaximum()[0]
            yPos -= translateTickY + t.getBoundCenter()[1]
          else:
            xPos -= t.getBoundCenter()[0]
            yPos += t.getBoundCenter()[1]
          if element.get("x") is not None:
            xPos += (float(element.get("x")) / 2)
          if element.get("y") is not None:
            yPos -= (float(element.get("y")) / 2)
          t.setPosition(Vector3(xPos, yPos, distance + 4))
```

```
        t.setFontResolution(600)
        t.setColor(Color('black'))
```

SVG File Code

```
   <svg width="960" height="500">
     <g transform="translate(10,30)">
       <g class="foreground">
         <path
d="M33.57142857142857,232.9891606135794L100.71428571428571,268.5666100815976L167.85714
285714283,221.99363021735493L235,299.604384761830 6L302.1428571428571,199.2867553805673
7L369.2857142857142,124.42944596668994L436.4285714285714,215.45400949648248L503.571428
57142856,435.8688524590164L570.7142857142857,333.88232140264125L637.8571428571428,54.3
4105164678585L704.9999999999999,240.95862126697966L772.1428571428571,433.2012012012012
3L839.2857142857142,413.2365145228216L906.4285714285713,283.4685724757191"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,188.96849826726032L100.71428571428571,247.0551128229448L167.8571
4285714283,220.1218738598977L235,212.59831727670704L302.1428571428571,277.613053436559
4L369.2857142857142,170.5905393621431L436.4285714285714,267.1888035661645L503.57142857
142856,428.327868852459L570.7142857142857,218.81052998683205L637.8571428571428,127.435
776402798L704.9999999999999,333.85997998799974L772.1428571428571,424.49849849849846L83
9.2857142857142,449.50207468879665L906.4285714285713,237.16772509841817" style="fill:
none; stroke: steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,154.72812730367738L100.71428571428571,313.83894973384326L167.857
14285714283,264.9097175887733L235,386.694912975219L302.1428571428571,202.312366313956 1
8L369.2857142857142,76.32677928053455L436.4285714285714,355.43880409633874L503.5714285
7142856,455.4754098360656L570.7142857142857,296.54735974641864L637.8571428571428,353.7
0538580935965L704.9999999999999,408.03538346034037L772.1428571428571,446.0480480480481
L839.2857142857142,354.0663900414938L906.4285714285713,381.46720757268446"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,357.16372063463524L100.71428571428571,276.392424840106 5L167.8571
4285714283,133.8046249398794L235,361.7952077717225L302.1428571428571,160.300592477616 5
L369.2857142857142,133.25733152773356L436.4285714285714,183.02964946993148L503.5714285
7142856,458.49180327868856L570.7142857142857,124.66072676319936L637.8571428571428,127.
99230294911854L704.9999999999999,383.57785286051956L772.1428571428571,456.684684684684
67L839.2857142857142,385.56016597510376L906.4285714285713,429.1528249069527 3"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,274.8890864679394L100.71428571428571,238.025875480345 2L167.85714
285714283,274.8954420009049L235,263.6040657621544L302.1428571428571,405.8653391129872L
369.2857142857142,123.86377562976865L436.4285714285714,198.94770996494466L503.57142857
142856,460L570.7142857142857,173.3264576406446L637.8571428571428,287.90272373540 86L704
.9999999999999,302.61988025232824L772.1428571428571,456.6846846846846 7L839.28571428571
42,241.45228215767636L906.4285714285713,251.02269151295695" style="fill: none; stroke:
steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,349.3774334053391L100.71428571428571,196.20197102194393L167.8571
4285714283,156.55271395030596L235,369.6210487856219L302.1428571428571,258.090821329194
2L369.2857142857142,78.09316913748017L436.4285714285714,147.82846951391488L503.5714285
7142856,453.96721311475414L570.7142857142857,283.94746672921804L637.8571428571428,406.
2951882800706L704.9999999999999,344.2219359747479L772.1428571428571,452.6786786786787L
839.2857142857142,237.6348547717842L906.4285714285713,460" style="fill: none; stroke:
steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,327.3791602605047L100.71428571428571,291.9790664793302L167.85714
285714283,244.23220235180182L235,398.924655800276 4L302.1428571428571,352.398114671865 7
6L369.2857142857142,121.88348613523324L436.4285714285714,221.88464254585483L503.571428
```

```
57142856,460L570.7142857142857,151.94431930021827L637.8571428571428,235.61710361251076
L704.9999999999999,339.11998916564505L772.1428571428571,458.06606606606607L839.2857142
857142,423.7344398340249L906.4285714285713,395.7587660265523" style="fill: none;
stroke: steelblue; stroke-opacity: 0.7;"></path>
        <path
d="M33.57142857142857,263.54543480051814L100.71428571428571,89.3927373137901L167.85714
285714283,88.80763234499813L235,273.3801817741761L302.1428571428571,335.2946298434572L
369.2857142857142,319.6965410322853L436.4285714285714,27.895016997389142L503.571428571
42856,458.49180327868856L570.7142857142857,231.22430768352464L637.8571428571428,0L704.
9999999999999,0L772.1428571428571,455.85585585585585L839.2857142857142,420.87136929460
58L906.4285714285713,28.46597536855516" style="fill: none; stroke: steelblue; stroke-
opacity: 0.7;"></path>
        <path
d="M33.57142857142857,160.71555357263242L100.71428571428571,246.8272729690319L167.8571
4285714283,268.5610032133769L235,336.21213849408036L302.1428571428571,281.142932858846
45L369.2857142857142,118.33578147253604L436.4285714285714,228.13919409258503L503.57142
857142856,420.7868852459016L570.7142857142857,239.7456490145049L637.8571428571428,326.
19399970232394L704.9999999999999,217.53736690989902L772.1428571428571,419.801801801801
8L839.2857142857142,429.46058091286307L906.4285714285713,279.8804951564896"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
        <path
d="M33.57142857142857,460L100.71428571428571,55.47773015570502L167.85714285714283,323.
9267398304239L235,59.13829285097617L302.1428571428571,88.2647788146661L369.28571428571
42,278.7313986260759L436.4285714285714,12.418867420362469L503.57142857142856,460L570.7
142857142857,133.59540395935667L637.8571428571428,163.62711828368535L704.9999999999999
,96.97590013459667L772.1428571428571,455.16516516516515L839.2857142857142,0L906.428571
4285713,141.8136401837563" style="fill: none; stroke: steelblue; stroke-opacity:
0.7;"></path>
        <path
d="M33.57142857142857,316.62979713196535L100.71428571428571,266.0601168341318L167.8571
4285714283,310.1866672801993L235,379.9141668312668L302.1428571428571,216.8278786272691
4L369.2857142857142,115.10460394194047L436.4285714285714,219.09953068967175L503.571428
57142856,458.49180327868856L570.7142857142857,309.84889610381543L637.8571428571428,384
.35738130169455L704.9999999999999,259.27490466249793L772.1428571428571,455.02702702702
703L839.2857142857142,441.86721991701245L906.4285714285713,278.36027121128814"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
        <path
d="M33.57142857142857,344.40681883106674L100.71428571428571,375.93448468435713L167.857
14285714283,214.4075740594054L235,294.35033026667065L302.1428571428571,390.73879876582
936L369.2857142857142,98.65952693437661L436.4285714285714,314.7147131862841L503.571428
57142856,432.8524590163934L570.7142857142857,229.25468235913223L637.8571428571428,344.
83519380833906L704.9999999999999,331.81726580678475L772.1428571428571,423.117117117117
1L839.2857142857142,400.8298755186722L906.4285714285713,432.13310332189604"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
        <path
d="M33.57142857142857,277.04402838718784L100.71428571428571,148.70016487973072L167.857
14285714283,222.7468181215604L235,391.10819931173967L302.1428571428571,143.30462435404
795L369.2857142857142,176.8897543888617L436.4285714285714,104.61436178672986L503.57142
857142856,435.8688524590164L570.7142857142857,117.31488066205227L637.8571428571428,279
.6726839743999L704.9999999999999,167.72614853146723L772.1428571428571,427.675675675675
7L839.2857142857142,395.10373443983406L906.4285714285713,234.62462007039346"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
        <path
d="M33.57142857142857,143.5043176728756L100.71428571428571,128.98411702915075L167.8571
4285714283,269.6067475434046L235,259.78328423149355L302.1428571428571,291.861666887504
9L369.2857142857142,0L436.4285714285714,309.42739050309297L503.57142857142856,458.4918
0327868856L570.7142857142857,314.14096659331767L637.8571428571428,268.54459824371156L7
04.9999999999999,372.1675240211937L772.1428571428571,448.12012012012013L839.2857142857
142,214.73029045643153L906.4285714285713,317.1783030969396" style="fill: none; stroke:
steelblue; stroke-opacity: 0.7;"></path>
        <path
d="M33.57142857142857,148.89081118325967L100.71428571428571,248.42164223757746L167.857
```

```
14285714283,279.3299019333945L235,278.9170517382236L302.1428571428571,270.228851577731
9L369.2857142857142,235.34095203890504L436.4285714285714,160.70339290372243L503.571428
57142856,452.4590163934426L570.7142857142857,232.92903267443353L637.8571428571428,202.
6842295507219L704.9999999999999,294.627246912406L772.1428571428571,431.95795795795794L
839.2857142857142,438.0497925311203L906.4285714285713,158.565752308013" style="fill:
none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,284.2478390237172L100.71428571428571,282.74186792778977L167.8571
4285714283,276.11505707496167L235,250.67944149091548L302.1428571428571,132.76950159950
007L369.2857142857142,173.70438610707635L436.4285714285714,204.48720477241727L503.5714
2857142856,444.91803278688525L570.7142857142857,33.289522511688105L637.8571428571428,2
18.55648402117748L704.9999999999999,363.9454583146881L772.1428571428571,436.1021021021
0215L839.2857142857142,376.01659751037346L906.4285714285713,335.1230498000386"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,261.99880810616185L100.71428571428571,346.84438145723544L167.857
14285714283,80.96077007100551L235,98.63861324163953L302.1428571428571,0L369.2857142857
142,162.7592266203776 3L436.4285714285714,17.273058872679748L503.57142857142856,444.918
03278688525L570.7142857142857,337.9574476535149L637.8571428571428,197.32338244987346L7
04.9999999999999,405.49489378431474L772.1428571428571,436.6546546546547L839.2857142857
142,366.47302904564316L906.4285714285713,441.78114790872837" style="fill: none;
stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,189.98506387973202L100.71428571428571,300.62423820688844L167.857
14285714283,105.59573463402232L235,294.0227250176423L302.1428571428571,206.86441159211
79L369.2857142857142,118.14360428518859L436.4285714285714,225.5438210301074L503.571428
57142856,458.49180327868856L570.7142857142857,239.52642115231166L637.8571428571428,226
.40507324955863L704.9999999999999,344.84511299153974L772.1428571428571,452.95495495495
5L839.2857142857142,276.76348547717845L906.4285714285713,67.1869222068460 6"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,189.17151609110158L100.71428571428571,118.06003705361684L167.857
14285714283,164.45652519263277L235,309.65563799471624L302.1428571428571,187.1214673758
7295L369.2857142857142,195.80417156495284L436.4285714285714,172.726198883240 8L503.5714
2857142856,334.81967213114757L570.7142857142857,368.3970079566907L637.8571428571428,35
4.60521783504487L704.9999999999999,357.03990697882347L772.1428571428571,312.6066066066
066L839.2857142857142,442.8215767634855L906.4285714285713,128.27143877209247"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,314.5208725861208L100.71428571428571,287.7795753017254L167.85714
285714283,232.57303502503993L235,259.7167394152848L302.1428571428571,239.2678263832364
7L369.2857142857142,50.01778882996398L436.4285714285714,316.1614105182346L503.57142857
142856,460L570.7142857142857,270.83661293945806L637.8571428571428,346.82558312602316L7
04.9999999999999,371.6221685276891L772.1428571428571,460L839.2857142857142,460L906.428
5714285713,321.809899338401" style="fill: none; stroke: steelblue; stroke-opacity:
0.7;"></path>
          <path
d="M33.57142857142857,154.31667784735833L100.71428571428571,228.52796085436066L167.857
14285714283,187.24694794943187L235,170.73011050927747L302.1428571428571,445.7854587442
503L369.2857142857142,168.68073700206733L436.4285714285714,209.10532019349236L503.5714
2857142856,98.03278688524591L570.7142857142857,158.74894661657595L637.8571428571428,35
8.69456316046853L704.9999999999999,311.74636915434616L772.1428571428571,110.5105105105
1053L839.2857142857142,438.0497925311203L906.4285714285713,159.0267609460798"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,223.51031993897155L100.71428571428571,376.58487317114253L167.857
14285714283,252.67249303476217L235,287.768365432675 9L302.1428571428571,239.52412500709
838L369.2857142857142,73.66703518590413L436.4285714285714,240.28491569167602L503.57142
857142856,458.49180327868856L570.7142857142857,289.5611843566821L637.8571428571428,248
.15460015734294L704.9999999999999,340.742547765173 3L772.1428571428571,440.936936936936
95L839.2857142857142,77.30290456431533L906.4285714285713,216.05376201399017"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
```

```
          <path
d="M33.57142857142857,182.23101342004L100.71428571428571,183.39400715365286L167.857142
85714283,304.3380586296092L235,300.3798878122651L302.1428571428571,267.7873895019781L3
69.2857142857142,76.4758071109382L436.4285714285714,199.72227347240838L503.57142857142
856,460L570.7142857142857,219.5036097386618L637.8571428571428,317.08809083370545L704.9
999999999999,393.04526454978924L772.1428571428571,460L839.2857142857142,460L906.428571
4285713,231.01067278345428" style="fill: none; stroke: steelblue; stroke-opacity:
0.7;"></path>
          <path
d="M33.57142857142857,185.26569924260158L100.71428571428571,312.24814842721565L167.857
14285714283,239.82898875592034L235,295.11602222240776L302.1428571428571,269.7317761078
195L369.2857142857142,114.29274583560668L436.4285714285714,150.12923529773494L503.5714
2857142856,456.983606557377L570.7142857142857,0L637.8571428571428,305.73944844889536L7
04.9999999999999,312.55785006427425L772.1428571428571,443.6996996996997L839.2857142857
142,264.356846473029L906.4285714285713,155.49565569057125" style="fill: none; stroke:
steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,228.94209257612775L100.71428571428571,294.13284120574946L167.857
14285714283,131.69297735297945L235,180.40854214365922L302.1428571428571,307.1010240587
555L369.2857142857142,184.2262533575914L436.4285714285714,132.8191066350259L503.571428
57142856,456.983606557377L570.7142857142857,206.94196741616685L637.8571428571428,345.6
528672577661L704.9999999999999,352.3514626159971L772.1428571428571,448.94894894894895L
839.2857142857142,439.00414937759336L906.4285714285713,186.4137551894751" style="fill:
none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,282.58186245716286L100.71428571428571,201.37551580319035L167.857
14285714283,275.13785309620295L235,201.04681969377884L302.1428571428571,234.3970167900
2064L369.2857142857142,43.68414298075249L436.4285714285714,349.1915689555352L503.57142
857142856,458.49180327868856L570.7142857142857,290.1212430358791L637.8571428571428,157
.23293147073187L704.9999999999999,430.66955173733174L772.1428571428571,446.73873873873
873L839.2857142857142,146.01659751037343L906.4285714285713,278.4384624620068"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,137.99232223502577L100.71428571428571,217.41707258697178L167.857
14285714283,460L235,59.86260758125002L302.1428571428571,272.3386776202465L369.28571428
57142,327.7708644504006L436.4285714285714,9.481086560504563L503.57142857142856,460L570
.7142857142857,195.4210865331989L637.8571428571428,335.1703131976781L704.9999999999999
,353.2490310659348L772.1428571428571,452.8168168168168L839.2857142857142,141.244813278
0083L906.4285714285713,187.42966010123007" style="fill: none; stroke: steelblue;
stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,210.64440371605548L100.71428571428571,441.22258098830326L167.857
14285714283,281.22484105672225L235,165.78489221242063L302.1428571428571,138.6560411894
9798L369.2857142857142,200.7837107965107L436.4285714285714,94.11824584396301L503.57142
857142856,441.9016393442623L570.7142857142857,285.18176526583744L637.8571428571428,225
.9067423614211L704.9999999999999,320.79604458389633L772.1428571428571,443.147147147147
17L839.2857142857142,442.8215767634855L906.4285714285713,172.54890586592364"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,216.60476094183136L100.71428571428571,351.52019555090646L167.857
14285714283,213.16981595118918L235,196.9807607956816L302.1428571428571,200.54212648356
003L369.2857142857142,88.64739840570323L436.4285714285714,339.5153782369793L503.571428
57142856,457.43606557377046L570.7142857142857,148.57026548365053L637.8571428571428,302
.2061406306478L704.9999999999999,289.99384212434L772.1428571428571,452.4024024024024L8
39.2857142857142,384.6058091286307L906.4285714285713,118.83965548836359" style="fill:
none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,0L100.71428571428571,460L167.85714285714283,320.16558805452104L2
35,247.50619221028768L302.1428571428571,209.77190558216145L369.2857142857142,112.94299
84851787L436.4285714285714,328.08276233556643L503.57142857142856,238.29508196721312L57
0.7142857142857,154.1822703934412L637.8571428571428,460L704.9999999999999,260.03457176
0289L772.1428571428571,247.26726726726727L839.2857142857142,447.5933609958506L906.4285
```

```
714285713,96.50340909808972" style="fill: none; stroke: steelblue; stroke-opacity:
0.7;"></path>
          <path
d="M33.57142857142857,249.86350996686454L100.71428571428571,205.2551646802795L167.8571
4285714283,232.7504335812663L235,319.2095113833737L302.1428571428571,355.1439550247023
L369.2857142857142,134.09209425847234L436.4285714285714,225.6233259745132L503.57142857
142856,453.96721311475414L570.7142857142857,164.23478132442716L637.8571428571428,292.0
272798792285L704.9999999999999,230.26763749265325L772.1428571428571,450.05405405405406
L839.2857142857142,446.63900414937757L906.4285714285713,89.10314496407655"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,233.84232763813486L100.71428571428571,182.5267375531315L167.8571
4285714283,303.4387185057853L235,238.9240437660138L302.1428571428571,438.0862783698347
6L369.2857142857142,182.657360471726L436.4285714285714,208.23515564858351L503.57142857
142856,460L570.7142857142857,254.7758884102381L637.8571428571428,376.28285598860316L70
4.9999999999999,178.05866943427745L772.1428571428571,456.13213213213214L839.2857142857
142,52.48962655601662L906.4285714285713,148.39478556488658" style="fill: none; stroke:
steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,432.1594890905337L100.71428571428571,224.28310558951148L167.8571
4285714283,0L235,155.2784894994396L302.1428571428571,379.3458138522402L369.28571428571
42,165.40884834873876L436.4285714285714,264.03543261166L503.57142857142856,440.3934426
2295083L570.7142857142857,203.0934908040714L637.8571428571428,267.948459526695L704.999
9999999999,202.38938724096934L772.1428571428571,406.2642642642643L839.2857142857142,26
3.402489626556L906.4285714285713,184.94904993902173" style="fill: none; stroke:
steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,258.41831194022967L100.71428571428571,194.02602396074792L167.857
14285714283,282.0750437963643L235,334.9887376422399L302.1428571428571,193.859433265819
75L369.2857142857142,221.93886521238895L436.4285714285714,283.8194819984601L503.571428
57142856,452.4590163934426L570.7142857142857,115.92301209948177L637.8571428571428,176.
83019710403784L704.9999999999999,230.30392732216563L772.1428571428571,448.948948948948
95L839.2857142857142,451.4107883817427L906.4285714285713,60.511307842505374"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,238.88159914919748L100.71428571428571,265.8600561122955L167.8571
4285714283,280.89700650892297L235,85.96993057120426L302.1428571428571,183.604459671771
02L369.2857142857142,114.75630587921228L436.4285714285714,275.3026977391243L503.571428
57142856,460L570.7142857142857,224.83529985590837L637.8571428571428,299.08069145882484
L704.9999999999999,233.42283655858458L772.1428571428571,454.1981981981982L839.28571428
57142,312.0746887966805L906.4285714285713,207.5253928835473" style="fill: none;
stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,227.02388174358893L100.71428571428571,220.83645437662594L167.857
14285714283,269.4898257677101L235,257.94647667639435L302.1428571428571,237.20077987469
008L369.2857142857142,152.12320567726283L436.4285714285714,247.90714738634617L503.5714
2857142856,438.88524590163934L570.7142857142857,182.12583014060434L637.8571428571428,6
2.4840424400926L704.9999999999999,183.85455842772123L772.1428571428571,420.49249249249
25L839.2857142857142,436.1410788381743L906.4285714285713,49.8959008675977"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,146.4031661151217L100.71428571428571,410.3151108727545L167.85714
285714283,294.8457239519824L235,117.62052206522623L302.1428571428571,277.0277688390845
L369.2857142857142,133.38515410909113L436.4285714285714,0L503.57142857142856,460L570.7
142857142857,242.88163507447229L637.8571428571428,280.93244881035906L704.9999999999999
,306.30934625274637L772.1428571428571,456.96096096096096L839.2857142857142,387.4688796
680498L906.4285714285713,117.37291552134874" style="fill: none; stroke: steelblue;
stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,257.10152603068326L100.71428571428571,177.09502524166834L167.857
14285714283,257.40127329917266L235,269.37426915226774L302.1428571428571,131.6803270930
738L369.2857142857142,213.85227673733417L436.4285714285714,236.3725895623515L503.57142
```

```
857142856,0L570.7142857142857,125.83622199303247L637.8571428571428,423.235504241883L70
4.9999999999999,380.1597541407879L772.1428571428571,1.3813813813814013L839.28571428571
42,439.00414937759336L906.4285714285713,264.72883294471495" style="fill: none; stroke:
steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,294.35788056881364L100.71428571428571,193.89044140786012L167.857
14285714283,320.96791334290873L235,341.0775883253537L302.1428571428571,406.48280300592
47L369.2857142857142,149.2293910983891L436.4285714285714,272.2429767804842L503.5714285
7142856,460L570.7142857142857,246.03303559350005L637.8571428571428,93.6583317386404L70
4.9999999999999,219.73612735801714L772.1428571428571,453.5075075075075L839.28571428571
42,316.8464730290456L906.4285714285713,150.79778582396537" style="fill: none; stroke:
steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,148.6452211369639L100.71428571428571,336.2918792300293L167.85714
285714283,222.81384655331664L235,272.53685419952876L302.1428571428571,324.442635673588
4L369.2857142857142,145.49357297202917L436.4285714285714,333.51348657492497L503.571428
57142856,401.1803278688525L570.7142857142857,242.07037780317896L637.8571428571428,332.
2360783313133L704.9999999999999,301.0164746183761L772.1428571428571,398.1141141141141L
839.2857142857142,391.2863070539419L906.4285714285713,179.63437028050038" style="fill:
none; stroke: steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,107.27683285775258L100.71428571428571,284.6209095865153L167.8571
4285714283,213.295557257378L235,187.34285324809366L302.1428571428571,229.0692612012341
L369.2857142857142,180.0089209175277L436.4285714285714,271.98836585422754L503.57142857
142856,441.9016393442623L570.7142857142857,301.63070580102436L637.8571428571428,288.24
798537135064L704.9999999999999,231.05714289471L772.1428571428571,443.83783783783787L83
9.2857142857142,400.8298755186722L906.4285714285713,162.54769261878147" style="fill:
none; stroke: steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,245.7972475169769L100.71428571428571,310.1968251787304L167.85714
285714283,197.61468402525554L235,328.4362060923375L302.1428571428571,203.6404247666999
4L369.2857142857142,112.15278894320211L436.4285714285714,277.8892911144901L503.5714285
7142856,460L570.7142857142857,271.72608431783567L637.8571428571428,282.6470200506048L7
04.9999999999999,229.30313446828166L772.1428571428571,437.2072072072072L839.2857142857
142,38.17427385892118L906.4285714285713,238.74172369560142" style="fill: none; stroke:
steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,137.99921253692548L100.71428571428571,237.257234541417L167.85714
285714283,238.8023954063941L235,289.58427109037626L302.1428571428571,263.1372920176418
L369.2857142857142,86.10328572137603L436.4285714285714,299.64486801839894L503.57142857
142856,414.75409836065575L570.7142857142857,258.06145181368123L637.8571428571428,149.8
6408964300153L704.9999999999999,266.5340802259437L772.1428571428571,413.3093093093093L
839.2857142857142,437.09543568464727L906.4285714285713,181.82169058406473"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,445.24884796078095L100.71428571428571,307.9854533527762L167.8571
4285714283,358.66183071541184L235,319.6667937101424L302.1428571428571,167.106324178008
38L369.2857142857142,105.14160952691024L436.4285714285714,219.9535991415417L503.571428
57142856,435.8688524590164L570.7142857142857,242.48428457424703L637.8571428571428,93.4
8081052922527L704.9999999999999,85.75730256119415L772.1428571428571,430.4384384384384L
839.2857142857142,435.18672199170123L906.4285714285713,242.93934396188385"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,95.38638651811861L100.71428571428571,411.4632300471148L167.85714
285714283,179.70322553785442L235,282.3586131303604L302.1428571428571,240.7965322076889
L369.2857142857142,167.05975456715154L436.4285714285714,354.2033070154788L503.57142857
142856,241.31147540983608L570.7142857142857,307.04289364892014L637.8571428571428,239.3
9444196381095L704.9999999999999,322.9294833388922L772.1428571428571,242.70870870870868
L839.2857142857142,437.09543568464727L906.4285714285713,246.04315873948693"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,194.66825443188702L100.71428571428571,355.1709851563919L167.8571
```

```
4285714283,264.5183824214157L235,334.54766495018873L302.1428571428571,221.741846334399
6L369.2857142857142,152.08146015213737L436.4285714285714,197.91658448964347L503.571428
57142856,438.88524590163934L570.7142857142857,66.82168002710563L637.8571428571428,147.
2941889392104L704.9999999999999,193.8312373911469L772.1428571428571,439.8318318318318
L839.2857142857142,403.6929460580913L906.4285714285713,158.960196647141" style="fill:
none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,392.0143754807966L100.71428571428571,248.39029514358256L167.8571
4285714283,202.78192094852125L235,22.19994788435532L302.1428571428571,434.308429082511
5L369.2857142857142,145.07427057364515L436.4285714285714,295.35696638482597L503.571428
57142856,404.1967213114754L570.7142857142857,354.6096306859435L637.8571428571428,296.2
0660840722064L704.9999999999999,438.0894106014513L772.1428571428571,387.75375375375376
L839.2857142857142,283.44398340248966L906.4285714285713,348.65391493368213"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,200.1243892079043L100.71428571428571,198.62996910721162L167.8571
4285714283,230.72446143345303L235,216.4698605581938L302.1428571428571,65.903954267542
56L369.2857142857142,109.91774091657359L436.4285714285714,23.564192449044942L503.571428
57142856,458.49180327868856L570.7142857142857,224.3974150374131L637.8571428571428,285.
03157491867063L704.9999999999999,280.15425807240524L772.1428571428571,453.369369369369
4L839.2857142857142,423.7344398340249L906.4285714285713,315.665055322805 9"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,239.0806796576676L100.71428571428571,212.29322441815626L167.8571
4285714283,297.4636125892499L235,288.14715900186496L302.1428571428571,120.023093376743
74L369.2857142857142,77.25234776415688L436.4285714285714,460L503.57142857142856,441.90
16393442623L570.7142857142857,370.8039471936874L637.8571428571428,325.61937870766 09L70
4.9999999999999,421.8874129953528L772.1428571428571,423.1171171171171L839.285714285714
2,409.4190871369294L906.4285714285713,295.92365389597666" style="fill: none; stroke:
steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,142.69372287077374L100.71428571428571,0L167.85714285714283,289.7
5609889706556L235,0L302.1428571428571,192.70476442862807L369.2857142857142,263.5839055
340046L436.4285714285714,112.41608930680765L503.57142857142856,144.78688524590163L570.
7142857142857,208.63984153638228L637.8571428571428,305.11787969637044L704.999999999999
9,110.89325136273125L772.1428571428571,156.0960960960961L839.2857142857142,399.8755186
721992L906.4285714285713,0" style="fill: none; stroke: steelblue; stroke-opacity:
0.7;"></path>
          <path
d="M33.57142857142857,283.150312363919L100.71428571428571,202.39824660157154L167.85714
285714283,264.35181928836784L235,256.92100399584183L302.1428571428571,228.47602642488 0
22L369.2857142857142,86.9088635274695L436.4285714285714,332.7740418159105 5L503.5714285
7142856,458.49180327868856L570.7142857142857,282.0977316419626L637.8571428571428,401.9
412727775296 6L704.9999999999999,246.81136432665932L772.1428571428571,454.1981981981982
L839.2857142857142,393.19502074688796L906.4285714285713,300.09395082557694"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,188.33360616361048L100.71428571428571,244.24279426821326L167.857
14285714283,162.95216527690596L235,222.74170948588193L302.1428571428571,383.6275530485
15L369.2857142857142,460L436.4285714285714,330.179156513827 7L503.57142857142856,348.39
344262295083L570.7142857142857,252.9963747475913L637.8571428571428,211.68646212072895L
704.9999999999999,392.79627599730196L772.1428571428571,326.2822822822822 7L839.28571428
57142,287.26141078838174L906.4285714285713,254.14074930648994" style="fill: none;
stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,84.32474506417951L100.71428571428571,315.8260496678027L167.85714
285714283,186.1755009876779L235,460L302.1428571428571,311.51867345586703L369.285714285
7142,63.9807434108647L436.4285714285714,414.008096822559L503.57142857142856,460L570.71
42857142857,317.0326049318823L637.8571428571428,374.420105887606L704.9999999999999,342
.496152971273 6L772.1428571428571,451.7117117117117L839.2857142857142,227.136929460580 9
2L906.4285714285713,123.3084744049085" style="fill: none; stroke: steelblue; stroke-
opacity: 0.7;"></path>
```

```
          <path
d="M33.57142857142857,151.9574876932146L100.71428571428571,313.9159157695051L167.85714
285714283,156.40278193190449L235,295.11602222240776L302.1428571428571,256.279316284616
4L369.2857142857142,270.2669577342218L436.4285714285714,51.2675196111487L503.571428571
42856,456.983606557377L570.7142857142857,265.8948515458518L637.8571428571428,63.307584
35925234L704.9999999999999,248.70791113900543L772.1428571428571,449.2252252252524L839
.2857142857142,438.0497925311203L906.4285714285713,101.75414052182232" style="fill:
none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,221.0182453910409L100.71428571428571,127.26869190981256L167.8571
4285714283,218.67521887226135L235,403.27907556861237L302.1428571428571,243.77519922769
7L369.2857142857142,68.6464154021873L436.4285714285714,98.9704862546999L503.5714285714
2856,458.49180327868856L570.7142857142857,329.6650395848769L637.8571428571428,299.3584
6569284087L704.9999999999999,353.82765223649227L772.1428571428571,451.2972972972973L83
9.2857142857142,419.91701244813277L906.4285714285713,181.56502562352682" style="fill:
none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,302.46333642540986L100.71428571428571,276.06187865383674L167.857
14285714283,319.2410492721423L235,140.34856278590593L302.1428571428571,201.81150504457
79L369.2857142857142,123.15816542632074L436.4285714285714,277.6493130000867L503.571428
57142856,458.49180327868856L570.7142857142857,90.88193790281446L637.8571428571428,358.
5145967553315L704.9999999999999,340.017154395255L772.1428571428571,455.02702702702703L
839.2857142857142,237.6348547717842L906.4285714285713,310.99683418323843" style="fill:
none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,169.77507016017802L100.71428571428571,222.99660046361328L167.857
14285714283,17.182461295408313L235,280.5034630938703L302.1428571428571,200.11925268318
532L369.2857142857142,184.7911109497757L436.4285714285714,174.62456234108006L503.57142
857142856,452.4590163934426L570.7142857142857,201.77584000734728L637.8571428571428,364
.56987944121965L704.9999999999999,225.1989563606035L772.1428571428571,449.777777777777
77L839.2857142857142,122.1576763485477L906.4285714285713,144.27419385414467"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,123.94102800308576L100.71428571428571,192.20075658521654L167.857
14285714283,358.1838121626255L235,213.58326587046574L302.1428571428571,338.01851255938
976L369.2857142857142,157.02472141407299L436.4285714285714,264.7851203389722L503.57142
857142856,450.95081967213116L570.7142857142857,193.98126187577355L637.8571428571428,36
7.47672811549825L704.9999999999999,381.2311105520569L772.1428571428571,438.72672672672
67L839.2857142857142,397.96680497925314L906.4285714285713,212.06077609903102"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,206.07022758324783L100.71428571428571,376.2902614584765L167.8571
4285714283,247.34020489796194L235,307.9058505838899L302.1428571428571,195.285165344791
8L369.2857142857142,187.70657409300654L436.4285714285714,352.34298886845926L503.571428
57142856,458.49180327868856L570.7142857142857,279.89631852577213L637.8571428571428,356
.82301035487234L704.9999999999999,460L772.1428571428571,451.02102102102106L839.2857142
857142,311.1203319502075L906.4285714285713,296.25676606446643" style="fill: none;
stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,99.89904209824672L100.71428571428571,291.27184545627813L167.8571
4285714283,273.700017639061L235,223.52446421500568L302.1428571428571,209.3270741448823
5L369.2857142857142,39.62374416080645L436.4285714285714,301.4266567418027L503.57142857
142856,288.0655737704918L570.7142857142857,260.4192931440408L637.8571428571428,331.297
612213221L704.9999999999999,407.66865457210224L772.1428571428571,283.0450450450451L839
.2857142857142,393.19502074688796L906.4285714285713,311.11310370103564" style="fill:
none; stroke: steelblue; stroke-opacity: 0.7;"></path>
          <path
d="M33.57142857142857,299.2258788612171L100.71428571428571,254.76853678103353L167.8571
4285714283,266.68546705713635L235,276.9190009746181L302.1428571428571,281.604421813776
4L369.2857142857142,145.15687499327393L436.4285714285714,188.37842996290868L503.571428
57142856,458.49180327868856L570.7142857142857,240.10988697304464L637.8571428571428,374
.9135463842996L704.9999999999999,450.5684749199147L772.1428571428571,448.6726726726726
```

```
6L839.2857142857142,356.92946058091286L906.4285714285713,378.04277959975474"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,264.2721155687657L100.71428571428571,331.3028490508334 6L167.85714285714283,230.51405263451963L235,350.7952643385255L302.1428571428571,205.2777830358326 4L369.2857142857142,141.44869018797047L436.4285714285714,158.2363008251633L503.57142857142856,460L570.7142857142857,166.5874845017667L637.8571428571428,281.8983011205374L704.9999999999999,454.48676845642444L772.1428571428571,440.7987987987988L839.2857142857142,119.29460580912861L906.4285714285713,208.13173841886106" style="fill: none;
stroke: steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,120.03594940121783L100.71428571428571,240.9806576573781L167.85714285714283,299.09572970384994L235,54.44858958558785L302.1428571428571,460L369.2857142857142,197.94205965257245L436.4285714285714,132.97275115949114L503.57142857142856,460L570.7142857142857,300.3855600524738L637.8571428571428,284.2104781952328 4L704.9999999999999,334.175499894592 9L772.1428571428571,450.33033033033036L839.2857142857142,167.96680497925314L906.4285714285713,379.61503415417087" style="fill: none; stroke: steelblue;
stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,173.7091864629107L100.71428571428571,389.37219383119316L167.85714285714283,261.8604279170459L235,227.61825011985675L302.1428571428571,367.0003217929546L369.2857142857142,217.01496205236995L436.4285714285714,285.87051445997463L503.57142857142856,22.622950819672155L570.7142857142857,216.92768235789111L637.8571428571428,366.7514830643618L704.9999999999999,361.4081944012857L772.1428571428571,0L839.2857142857142,427.551867219917L906.4285714285713,213.5135637239103" style="fill: none; stroke:
steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,277.91737415302237L100.71428571428571,408.77808384932115L167.85714285714283,184.36623730343683L235,290.7893294608256L302.1428571428571,315.4021465482973L369.2857142857142,125.90805030086806L436.4285714285714,217.04313286381094L503.57142857142856,289.57377049180326L570.7142857142857,460L637.8571428571428,199.623333545959 1L704.9999999999999,351.686552295266L772.1428571428571,262.7387387387387 3L839.2857142857142,433.278008298755 2L906.4285714285713,288.4893809280074" style="fill: none; stroke:
steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,394.46338564185186L100.71428571428571,352.33776853899695L167.85714285714283,170.08010982232759L235,352.5889883908228L302.1428571428571,190.93528175812526L369.2857142857142,114.0187030874823L436.4285714285714,313.21046012770813L503.57142857142856,460L570.7142857142857,380.9512285026007L637.8571428571428,363.646084497459L704.9999999999999,387.4638887707959L772.1428571428571,458.7567567567568L839.2857142857142,350.24896265560164L906.4285714285713,357.8633327646236" style="fill: none; stroke:
steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,161.7200611567788L100.71428571428571,296.287635349859 6L167.85714285714283,206.43497047922378L235,248.05604008268037L302.1428571428571,229.5076567794204 2L369.2857142857142,88.81149895668304L436.4285714285714,158.84697683986403L503.57142857142856,456.983606557377L570.7142857142857,384.7277709724141L637.8571428571428,220.1766707065549 4L704.9999999999999,335.453708334082L772.1428571428571,451.987987987988L839.2857142857142,432.32365145228215L906.4285714285713,158.150670129475 9" style="fill:
none; stroke: steelblue; stroke-opacity: 0.7;"></path>
         <path
d="M33.57142857142857,223.86787739114328L100.71428571428571,218.73925733205078L167.85714285714283,72.40355761235041L235,147.61986366473135L302.1428571428571,423.9709250600996L369.2857142857142,143.48069980213035L436.4285714285714,123.1165768551328 5L503.57142857142856,441.9016393442623L570.7142857142857,262.95411530065013L637.8571428571428,241.70172864706245L704.9999999999999,393.9809373207148L772.1428571428571,439.4174174174174L839.2857142857142,334.02489626556013L906.4285714285713,180.6043487327251 3"
style="fill: none; stroke: steelblue; stroke-opacity: 0.7;"></path>
        </g>
        <g class="dimension" transform="translate(33.57142857142857)">
         <g class="axis">
```

```
            <g class="tick" transform="translate(0,437.8767170921213)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">7.4</text>
            </g>
            <g class="tick" transform="translate(0,388.66027494872407)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">7.6</text>
            </g>
            <g class="tick" transform="translate(0,339.44383280532656)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">7.8</text>
            </g>
            <g class="tick" transform="translate(0,290.22739066192906)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">8.0</text>
            </g>
            <g class="tick" transform="translate(0,241.01094851853176)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">8.2</text>
            </g>
            <g class="tick" transform="translate(0,191.79450637513406)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">8.4</text>
            </g>
            <g class="tick" transform="translate(0,142.57806423173676)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">8.6</text>
            </g>
            <g class="tick" transform="translate(0,93.36162208833906)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">8.8</text>
            </g>
            <g class="tick" transform="translate(0,44.145179944941766)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">9.0</text>
            </g>
            <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0,
0, 0); shape-rendering: crispEdges;"></path>
            <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255)
0px 1px 0px;">211034_s_at</text>
          </g>
          <g class="brush"></g>
        </g>
        <g class="dimension" transform="translate(100.71428571428571)">
```

```
          <g class="axis">
            <g class="tick" transform="translate(0,425.77050248427634)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.8</text>
            </g>
            <g class="tick" transform="translate(0,374.7996179399782)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.0</text>
            </g>
            <g class="tick" transform="translate(0,323.82873339567993)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.2</text>
            </g>
            <g class="tick" transform="translate(0,272.857848851382)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.4</text>
            </g>
            <g class="tick" transform="translate(0,221.88696430708387)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.6</text>
            </g>
            <g class="tick" transform="translate(0,170.91607976278567)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.8</text>
            </g>
            <g class="tick" transform="translate(0,119.94519521848751)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.0</text>
            </g>
            <g class="tick" transform="translate(0,68.97431067418934)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.2</text>
            </g>
            <g class="tick" transform="translate(0,18.003426129891388)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.4</text>
            </g>
            <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0,
0, 0); shape-rendering: crispEdges;"></path>
            <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255)
0px 1px 0px;">233250_x_at</text>
          </g>
          <g class="brush"></g>
        </g>
```

```
        <g class="dimension" transform="translate(167.85714285714283)">
          <g class="axis">
            <g class="tick" transform="translate(0,447.5221282567897)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.0</text>
            </g>
            <g class="tick" transform="translate(0,397.12481114701)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.2</text>
            </g>
            <g class="tick" transform="translate(0,346.7274940372304)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.4</text>
            </g>
            <g class="tick" transform="translate(0,296.33017692745096)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.6</text>
            </g>
            <g class="tick" transform="translate(0,245.93285981767133)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.8</text>
            </g>
            <g class="tick" transform="translate(0,195.53554270789166)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.0</text>
            </g>
            <g class="tick" transform="translate(0,145.13822559811203)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.2</text>
            </g>
            <g class="tick" transform="translate(0,94.74090848833241)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.4</text>
            </g>
            <g class="tick" transform="translate(0,44.34359137855298)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.6</text>
            </g>
            <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0,
0, 0); shape-rendering: crispEdges;"></path>
            <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255)
0px 1px 0px;">205228_at</text>
          </g>
          <g class="brush"></g>
```

```
        </g>
        <g class="dimension" transform="translate(235)">
          <g class="axis">
            <g class="tick" transform="translate(0,441.1063910286905)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.0</text>
            </g>
            <g class="tick" transform="translate(0,398.44945756145165)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.1</text>
            </g>
            <g class="tick" transform="translate(0,355.7925240942124)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.2</text>
            </g>
            <g class="tick" transform="translate(0,313.13559062697357)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.3</text>
            </g>
            <g class="tick" transform="translate(0,270.4786571597343)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.4</text>
            </g>
            <g class="tick" transform="translate(0,227.8217236924955)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.5</text>
            </g>
            <g class="tick" transform="translate(0,185.1647902252566)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.6</text>
            </g>
            <g class="tick" transform="translate(0,142.5078567580174)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.7</text>
            </g>
            <g class="tick" transform="translate(0,99.85092329077857)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.8</text>
            </g>
            <g class="tick" transform="translate(0,57.19398982353932)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.9</text>
```

```
            </g>
            <g class="tick" transform="translate(0,14.537056356300479)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.0</text>
            </g>
            <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0,
0, 0); shape-rendering: crispEdges;"></path>
            <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255)
0px 1px 0px;">222724_at</text>
          </g>
          <g class="brush"></g>
        </g>
        <g class="dimension" transform="translate(302.1428571428571)">
          <g class="axis">
            <g class="tick" transform="translate(0,438.9706411251396)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.4</text>
            </g>
            <g class="tick" transform="translate(0,401.1126464631168)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.5</text>
            </g>
            <g class="tick" transform="translate(0,363.25465180109404)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.6</text>
            </g>
            <g class="tick" transform="translate(0,325.39665713907124)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.7</text>
            </g>
            <g class="tick" transform="translate(0,287.53866247704866)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.8</text>
            </g>
            <g class="tick" transform="translate(0,249.68066781502583)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.9</text>
            </g>
            <g class="tick" transform="translate(0,211.82267315300308)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.0</text>
            </g>
            <g class="tick" transform="translate(0,173.96467849098042)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
```

```
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.1</text>
            </g>
            <g class="tick" transform="translate(0,136.10668382895747)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.2</text>
            </g>
            <g class="tick" transform="translate(0,98.24868916693487)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.3</text>
            </g>
            <g class="tick" transform="translate(0,60.39069450491192)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.4</text>
            </g>
            <g class="tick" transform="translate(0,22.53269984288927)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.5</text>
            </g>
            <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0,
0, 0); shape-rendering: crispEdges;"></path>
            <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255)
0px 1px 0px;">218994_s_at</text>
          </g>
          <g class="brush"></g>
        </g>
        <g class="dimension" transform="translate(369.2857142857142)">
          <g class="axis">
            <g class="tick" transform="translate(0,452.0153740464906)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.0</text>
            </g>
            <g class="tick" transform="translate(0,415.0724314575902)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.5</text>
            </g>
            <g class="tick" transform="translate(0,378.1294888686898)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.0</text>
            </g>
            <g class="tick" transform="translate(0,341.1865462797894)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.5</text>
            </g>
            <g class="tick" transform="translate(0,304.2436036908891)" style="opacity:
1;">
```

```
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.0</text>
            </g>
            <g class="tick" transform="translate(0,267.3006611019887)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.5</text>
            </g>
            <g class="tick" transform="translate(0,230.35771851308832)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">7.0</text>
            </g>
            <g class="tick" transform="translate(0,193.4147759241879)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">7.5</text>
            </g>
            <g class="tick" transform="translate(0,156.4718333352875)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">8.0</text>
            </g>
            <g class="tick" transform="translate(0,119.5288907463871)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">8.5</text>
            </g>
            <g class="tick" transform="translate(0,82.58594815748677)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">9.0</text>
            </g>
            <g class="tick" transform="translate(0,45.64300556858638)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">9.5</text>
            </g>
            <g class="tick" transform="translate(0,8.700062979685995)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">10.0</text>
            </g>
            <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0,
0, 0); shape-rendering: crispEdges;"></path>
            <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255)
0px 1px 0px;">225219_at</text>
          </g>
          <g class="brush"></g>
        </g>
        <g class="dimension" transform="translate(436.4285714285714)">
          <g class="axis">
```

```
            <g class="tick" transform="translate(0,412.2063099229548)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.2</text>
            </g>
            <g class="tick" transform="translate(0,363.4302704101216)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.3</text>
            </g>
            <g class="tick" transform="translate(0,314.65423089728796)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.4</text>
            </g>
            <g class="tick" transform="translate(0,265.8781913844547)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.5</text>
            </g>
            <g class="tick" transform="translate(0,217.10215187162146)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.6</text>
            </g>
            <g class="tick" transform="translate(0,168.32611235878784)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.7</text>
            </g>
            <g class="tick" transform="translate(0,119.5500728459546)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.8</text>
            </g>
            <g class="tick" transform="translate(0,70.77403333312093)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.9</text>
            </g>
            <g class="tick" transform="translate(0,21.99799382028771)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.0</text>
            </g>
            <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0,
0, 0); shape-rendering: crispEdges;"></path>
            <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255)
0px 1px 0px;">234143_at</text>
          </g>
          <g class="brush"></g>
        </g>
        <g class="dimension" transform="translate(503.57142857142856)">
```

```
          <g class="axis">
            <g class="tick" transform="translate(0,460)" style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">0</text>
            </g>
            <g class="tick" transform="translate(0,384.59016393442624)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">50</text>
            </g>
            <g class="tick" transform="translate(0,309.1803278688525)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">100</text>
            </g>
            <g class="tick" transform="translate(0,233.7704918032787)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">150</text>
            </g>
            <g class="tick" transform="translate(0,158.36065573770495)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">200</text>
            </g>
            <g class="tick" transform="translate(0,82.95081967213113)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">250</text>
            </g>
            <g class="tick" transform="translate(0,7.540983606557385)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">300</text>
            </g>
            <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0,
0, 0); shape-rendering: crispEdges;"></path>
            <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255)
0px 1px 0px;">Initial_Blastsx109</text>
          </g>
          <g class="brush"></g>
        </g>
        <g class="dimension" transform="translate(570.7142857142857)">
          <g class="axis">
            <g class="tick" transform="translate(0,449.95148541025173)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.0</text>
            </g>
            <g class="tick" transform="translate(0,392.8608962974271)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.1</text>
```

```
            </g>
            <g class="tick" transform="translate(0,335.77030718460236)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.2</text>
            </g>
            <g class="tick" transform="translate(0,278.6797180717779)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.3</text>
            </g>
            <g class="tick" transform="translate(0,221.58912895895324)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.4</text>
            </g>
            <g class="tick" transform="translate(0,164.49853984612852)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.5</text>
            </g>
            <g class="tick" transform="translate(0,107.40795073330379)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.6</text>
            </g>
            <g class="tick" transform="translate(0,50.31736162047913)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.7</text>
            </g>
            <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0,
0, 0); shape-rendering: crispEdges;"></path>
            <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255)
0px 1px 0px;">238291_at</text>
          </g>
          <g class="brush"></g>
        </g>
        <g class="dimension" transform="translate(637.8571428571428)">
          <g class="axis">
            <g class="tick" transform="translate(0,418.20524760264493)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.6</text>
            </g>
            <g class="tick" transform="translate(0,369.3013331632326)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.7</text>
            </g>
            <g class="tick" transform="translate(0,320.39741872382046)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
```

```
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.8</text>
            </g>
            <g class="tick" transform="translate(0,271.49350428440806)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">3.9</text>
            </g>
            <g class="tick" transform="translate(0,222.58958984499577)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.0</text>
            </g>
            <g class="tick" transform="translate(0,173.68567540558362)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.1</text>
            </g>
            <g class="tick" transform="translate(0,124.78176096617106)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.2</text>
            </g>
            <g class="tick" transform="translate(0,75.87784652675892)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.3</text>
            </g>
            <g class="tick" transform="translate(0,26.97393208734636)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">4.4</text>
            </g>
            <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0,
0, 0); shape-rendering: crispEdges;"></path>
            <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255)
0px 1px 0px;">1554266_at</text>
          </g>
          <g class="brush"></g>
        </g>
        <g class="dimension" transform="translate(704.9999999999999)">
          <g class="axis">
            <g class="tick" transform="translate(0,457.781884976142)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.4</text>
            </g>
            <g class="tick" transform="translate(0,417.45985218468724)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.6</text>
            </g>
            <g class="tick" transform="translate(0,377.13781939323235)"
style="opacity: 1;">
```

```
            <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-rendering: crispEdges;"></line>
            <text dy=".32em" x="-9" y="0" style="text-anchor: end;">5.8</text>
          </g>
          <g class="tick" transform="translate(0,336.8157866017774)" style="opacity: 1;">
            <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-rendering: crispEdges;"></line>
            <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.0</text>
          </g>
          <g class="tick" transform="translate(0,296.49375381032246)" style="opacity: 1;">
            <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-rendering: crispEdges;"></line>
            <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.2</text>
          </g>
          <g class="tick" transform="translate(0,256.1717210188676)" style="opacity: 1;">
            <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-rendering: crispEdges;"></line>
            <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.4</text>
          </g>
          <g class="tick" transform="translate(0,215.84968822741286)" style="opacity: 1;">
            <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-rendering: crispEdges;"></line>
            <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.6</text>
          </g>
          <g class="tick" transform="translate(0,175.5276554359579)" style="opacity: 1;">
            <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-rendering: crispEdges;"></line>
            <text dy=".32em" x="-9" y="0" style="text-anchor: end;">6.8</text>
          </g>
          <g class="tick" transform="translate(0,135.20562264450297)" style="opacity: 1;">
            <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-rendering: crispEdges;"></line>
            <text dy=".32em" x="-9" y="0" style="text-anchor: end;">7.0</text>
          </g>
          <g class="tick" transform="translate(0,94.88358985304804)" style="opacity: 1;">
            <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-rendering: crispEdges;"></line>
            <text dy=".32em" x="-9" y="0" style="text-anchor: end;">7.2</text>
          </g>
          <g class="tick" transform="translate(0,54.561557061593156)" style="opacity: 1;">
            <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-rendering: crispEdges;"></line>
            <text dy=".32em" x="-9" y="0" style="text-anchor: end;">7.4</text>
          </g>
          <g class="tick" transform="translate(0,14.239524270138377)" style="opacity: 1;">
            <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-rendering: crispEdges;"></line>
            <text dy=".32em" x="-9" y="0" style="text-anchor: end;">7.6</text>
          </g>
          <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0, 0, 0); shape-rendering: crispEdges;"></path>
          <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255) 0px 1px 0px;">219483_s_at</text>
```

```
          </g>
          <g class="brush"></g>
        </g>
        <g class="dimension" transform="translate(772.1428571428571)">
          <g class="axis">
            <g class="tick" transform="translate(0,460)" style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">0</text>
            </g>
            <g class="tick" transform="translate(0,390.93093093093097)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">50</text>
            </g>
            <g class="tick" transform="translate(0,321.8618618618619)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">100</text>
            </g>
            <g class="tick" transform="translate(0,252.79279279279277)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">150</text>
            </g>
            <g class="tick" transform="translate(0,183.7237237237237)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">200</text>
            </g>
            <g class="tick" transform="translate(0,114.65465465465468)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">250</text>
            </g>
            <g class="tick" transform="translate(0,45.58558558558558)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">300</text>
            </g>
            <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0,
0, 0); shape-rendering: crispEdges;"></path>
            <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255)
0px 1px 0px;">Initial_WBCx109</text>
          </g>
          <g class="brush"></g>
        </g>
        <g class="dimension" transform="translate(839.2857142857142)">
          <g class="axis">
            <g class="tick" transform="translate(0,460)" style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">0</text>
            </g>
            <g class="tick" transform="translate(0,412.28215767634856)"
style="opacity: 1;">
```

```
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">50</text>
            </g>
            <g class="tick" transform="translate(0,364.56431535269707)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">100</text>
            </g>
            <g class="tick" transform="translate(0,316.8464730290456)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">150</text>
            </g>
            <g class="tick" transform="translate(0,269.12863070539413)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">200</text>
            </g>
            <g class="tick" transform="translate(0,221.41078838174272)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">250</text>
            </g>
            <g class="tick" transform="translate(0,173.69294605809128)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">300</text>
            </g>
            <g class="tick" transform="translate(0,125.97510373443984)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">350</text>
            </g>
            <g class="tick" transform="translate(0,78.25726141078836)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">400</text>
            </g>
            <g class="tick" transform="translate(0,30.539419087136924)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">450</text>
            </g>
            <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0,
0, 0); shape-rendering: crispEdges;"></path>
            <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255)
0px 1px 0px;">Initial_plateletx109</text>
          </g>
          <g class="brush"></g>
        </g>
        <g class="dimension" transform="translate(906.4285714285713)">
          <g class="axis">
```

```
            <g class="tick" transform="translate(0,450.4347974445983)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">7.8</text>
            </g>
            <g class="tick" transform="translate(0,392.3000385458727)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">8.0</text>
            </g>
            <g class="tick" transform="translate(0,334.16527964714743)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">8.2</text>
            </g>
            <g class="tick" transform="translate(0,276.0305207484217)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">8.4</text>
            </g>
            <g class="tick" transform="translate(0,217.89576184969638)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">8.6</text>
            </g>
            <g class="tick" transform="translate(0,159.76100295097058)"
style="opacity: 1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">8.8</text>
            </g>
            <g class="tick" transform="translate(0,101.6262440522453)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">9.0</text>
            </g>
            <g class="tick" transform="translate(0,43.49148515352001)" style="opacity:
1;">
              <line x2="-6" y2="0" style="fill: none; stroke: rgb(0, 0, 0); shape-
rendering: crispEdges;"></line>
              <text dy=".32em" x="-9" y="0" style="text-anchor: end;">9.2</text>
            </g>
            <path class="domain" d="M-6,0H0V460H-6" style="fill: none; stroke: rgb(0,
0, 0); shape-rendering: crispEdges;"></path>
            <text text-anchor="middle" y="-9" style="text-shadow: rgb(255, 255, 255)
0px 1px 0px;">212034_s_at</text>
          </g>
          <g class="brush"></g>
        </g>
      </g>
    </svg>
```

Code Used to Generate the SVG File

The code used to generate the SVG file was an adapted version of Bostock's parallel coordinates example which can be found at the URL 'http://bl.ocks.org/mbostock/1341021' (Bostock 2011c). It was adapted by simplifying the features, removing interactivity, removing separate styling, adding inline styling, and using the genes dataset. Categorical variables were not included in the generated SVG image.

```
    <!DOCTYPE html>
    <meta charset="utf-8">
    <style>
    </style>
    <head>
        <script src="parallel-coordinates-master/examples/lib/d3.min.js"></script>
    </head>
    <body>
        <script src="merged.genes.js"></script> <!-- Genes Data -->
        <script>

        var m = [30, 10, 10, 10],
            w = 960 - m[1] - m[3],
            h = 500 - m[0] - m[2];

        var x = d3.scale.ordinal().rangePoints([0, w], 1),
            y = {},
            dragging = {};

        var line = d3.svg.line(),
            axis = d3.svg.axis().orient("left"),
            foreground;

        var svg = d3.select("body").append("svg")
            .attr("width", w + m[1] + m[3])
            .attr("height", h + m[0] + m[2])
          .append("g")
            .attr("transform", "translate(" + m[3] + "," + m[0] + ")");

          // Extract the list of dimensions and create a scale for each.
          x.domain(dimensions = d3.keys(genes[0]).filter(function(d) {
            return d != "name" && d != "group" && d != "Gender" && (y[d] =
d3.scale.linear()
                .domain(d3.extent(genes, function(p) { return +p[d]; }))
                .range([h, 0]));
          }));

          // Add blue foreground lines for focus.
          foreground = svg.append("g")
              .attr("class", "foreground")
            .selectAll("path")
              .data(genes)
            .enter().append("path")
              .attr("d", path)
              .style({"fill": "none", "stroke": "steelblue", "stroke-opacity": ".7"});
// NEW CODE

          // Add a group element for each dimension.
          var g = svg.selectAll(".dimension")
              .data(dimensions)
            .enter().append("g")
```

```
            .attr("class", "dimension")
            .attr("transform", function(d) { return "translate(" + x(d) + ")"; });

        // Add an axis and title.
        g.append("g")
            .attr("class", "axis")
            .each(function(d) { d3.select(this).call(axis.scale(y[d])); })
          .append("text")
            .style({"text-shadow": "0 1px 0 #fff"})
            .attr("text-anchor", "middle")
            .attr("y", -9)
            .text(String);

        // Style inline (NEW CODE)
        g.selectAll(".axis").selectAll("line").style({"fill": "none", "stroke":
"#000", "shape-rendering": "crispEdges"})
        g.selectAll(".axis").selectAll("path").style({"fill": "none", "stroke":
"#000", "shape-rendering": "crispEdges"})

        // Add and store a brush for each axis.
        g.append("g")
            .attr("class", "brush")
          .selectAll("rect")
            .attr("x", -8)
            .attr("width", 16);

        function position(d) {
        var v = dragging[d];
        return v == null ? x(d) : v;
      }

      function transition(g) {
        return g.transition().duration(500);
      }

      // Returns the path for a given data point.
      function path(d) {
        return line(dimensions.map(function(p) { return [position(p), y[p](d[p])];
}));
      }
   </script>
```

7.2 Example Empirical Method: Evaluating the Implemented Parallel Co-Ordinates in a Data Arena with Parallel Coordinates on a PC Using a Factorial Design

Construct validity is the extent a test agrees with another test in a way that is expected (Peat et al. 2001). Construct validity is used when an optimum standard is unavailable. Construct validity often uses a factorial design. A factorial design is one that allows more than one question to be addressed in a single experiment (*The Cambridge Dictionary of Statistics* 2006). The hypothesis is that the Data Arena is superior for the analysis of complex high-dimensional data compared to a two-monitor PC. The implementation that was developed on a PC for the Data Arena can also be easily modified for use on a PC.

There are eight sub-studies to this experiment. The first six sub-studies involve individual participants and the last two studies involve group participants. The schematic of the design can be seen in Figure B and C. Data Arena is DA in the figure and personal computer is PC, The arrow indicates the order that the hardware will be used. For example in sub-study 1, Tester-Information Technology-1 evaluated the simple data on a PC first then evaluates the same data in the Data Arena. Tester-Information Technology-2, however, evaluates the same simple data in the Data Arena first, and then evaluates the data on a PC. The process is identical for Tester-Statistician 1 and 2 and Tester-Clinicians 1 and 2. In sub-study 2, Tester-Information Technology-1 now uses the Data Arena to evaluate the data first and then moves to the PC. Sub-studies 3 to 8 use a complex data set. Sub-studies 7 and 8 use groups of three rather than individuals but the same process is followed. Sub-studies 1-6 are to be analysed separately as well as one study. Sub-studies 7-8 are to be analysed separately and as one. Both quantitative measures and qualitative outcomes are measured in the evaluation process.

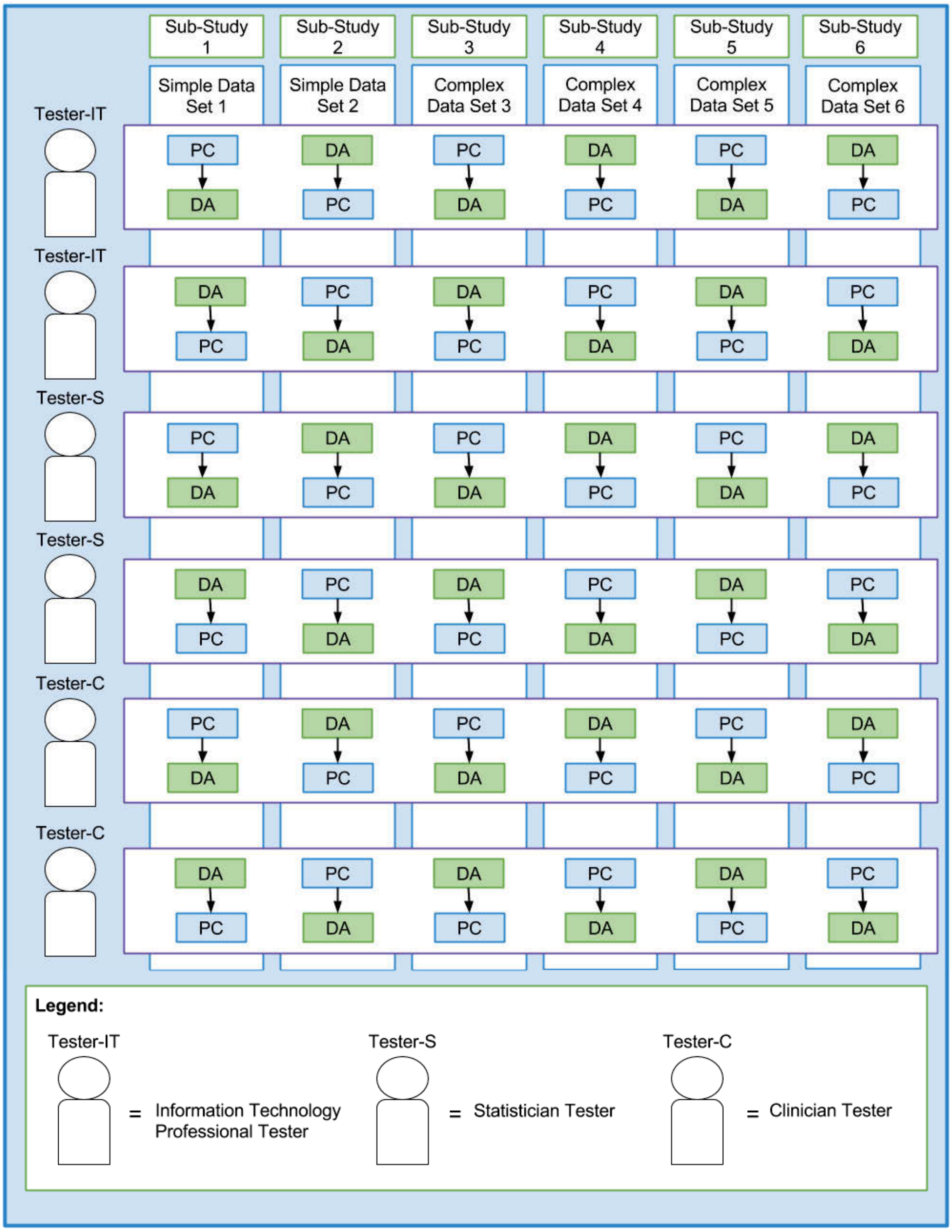


Figure Z - Study Design for Individual Testing

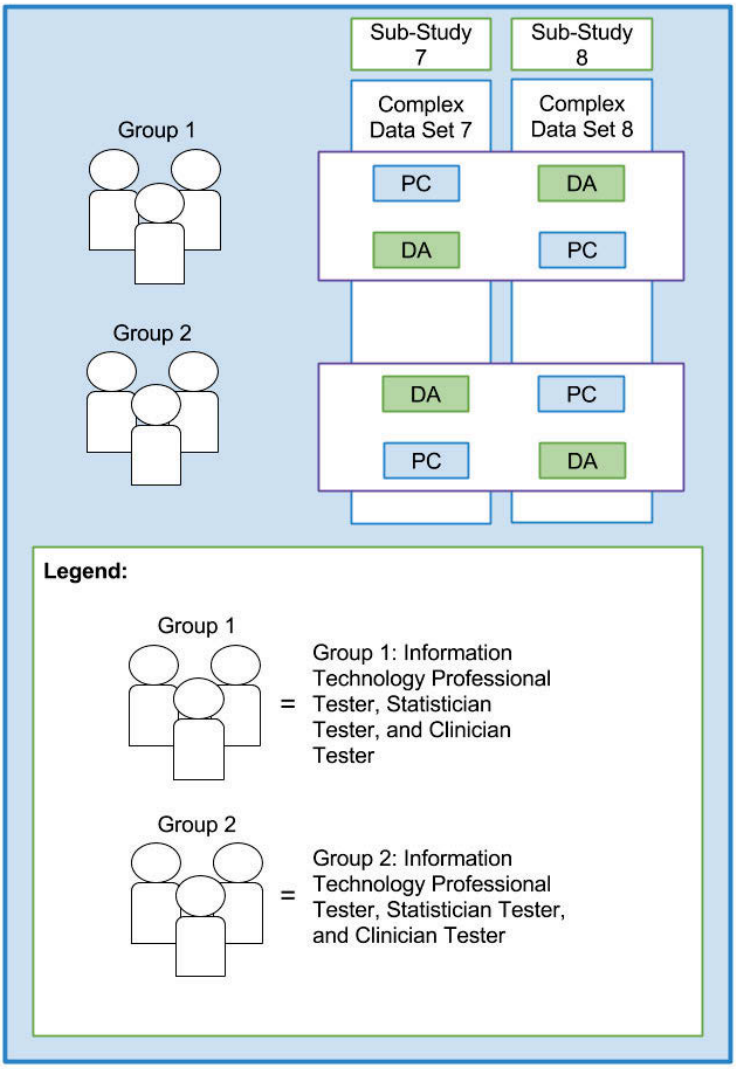


Figure AA - Study Design for Group Testing

The factors are the participant and their knowledge domain (information technology (IT) professional, statistician, or clinician), visualisation hardware (the Data Arena or PC), and data complexity (simple data or complex data). The participants will be members of the acute lymphoblastic leukaemia project as they have ethical clearance to view and work with the acute lymphoblastic leukaemia data set. The data used will be sections of the acute lymphoblastic leukaemia data set. A specified order and repetition are used to determine construct validity. Sub-studies 1 and 2 will use simple data to facilitate training and familiarity with the process.

The quantitative outcomes that will be recorded are the time taken to find the first data pattern, the average time between discoveries, the number of patterns found, the number of outliers found, and the number of clusters found. To record these outcomes an observer will be present in the room, the participants will be filmed, and the participants will also keep a record. The qualitative outcomes will be how the participants interact with visualisation and the level of comfort in using the visualisation. The qualitative outcomes will be derived from the video of the participants and a short unstructured and structured feedback session after the test. Each participant will be blind to the other participants' results.

The expected product is that users using the parallel coordinates in the Data Arena will have more patterns, outliers, and clusters identified than users using the parallel coordinates on the PC. Additionally the Data Arena will have a smaller average time between findings than the PC. We expect that the Data Arena will perform better with group analysis, which is reflected in the feedback and video of group analysis.

There are a few limitations to using construct validity to evaluate parallel coordinates in the Data Arena. A participant will know some of the outcomes of the first test as it is based on their actions and information gained. Additionally, it requires that there be no change in the parallel coordinates and environment between tests. The Data Arena is relatively new so there may be changes to its environment.

To overcome the limitations the above method has included a few characteristics. By switching the order in which the tests are taken between the Data Arena and PC, it will counteract bias created in the intra-participant study. As this is done multiple times for all participants it improves validity. Also before each pair of tests the environment in the PC will be compared to the Data Arena environment. This will ensure that the PC environment remains close to the Data Arena environment. Also each sub-study will be organised over two days to remove variability between participants based on environment changes. If possible, each pair of tests will occur at the same or similar time over those two days. Changes to the Data Arena will also be recorded during testing period to give context to the tests, possible confounding variables and for the analysis of the outcomes.