

November 11, 2008



University
of Glasgow

Genome Visualisation and User Studies in
Biologist-Computer Interaction

Joanna Jakubowska

Submitted for the degree of Doctor of Philosophy in Computing Science
at
University of Glasgow
June 2008

Abstract

We surveyed a number of genome visualisation tools used in biomedical research. We recognised that none of the tools shows all the relevant data geneticists who look for candidate disease genes would like to see. The biological researchers we collaborate with would like to view integrated data from a variety of sources and be able to see both data overviews and details. In response to this need, we developed a new visualisation tool, VisGenome, which allows the users to add their own data or data downloaded from other sources, such as Ensembl. VisGenome visualises single and comparative representations of the rat, the mouse, and the human chromosomes, and can easily be used for other genomes. In the context of VisGenome development we made the following research contributions. We developed a new algorithm (CartoonPlus) which allows the users to see different kinds of data in cartoon scaling depending on a selected basis. Also, two user studies were conducted: an initial quantitative user study and a mixed paradigm user study. The first study showed that neither Ensembl nor VisGenome fulfil all user requirements and can be regarded as user-friendly, as the users make a significant number of mistakes during data navigation. To help users navigate their data easily, we improved existing visualisation techniques in VisGenome and added a new technique CartoonPlus. To verify if this solution was useful, we conducted a second user study. We saw that the users became more familiar with the tool, and found new ways to use the application on its own and in connection with other tools. They frequently used CartoonPlus, which allowed them to see small regions of their data in a way that was not possible before.

Thesis Statement

The goal of this PhD was to find out which of the known visualisation techniques and user study techniques are applicable to the development of improved genome maps for comparative genomics, as required in the data analysis carried out at the British Heart Foundation Cardiovascular Research Centre at Glasgow. As biological researchers need user-friendly visualisation techniques to work more effectively and efficiently, we hypothesised that improved zooming is the most essential application feature. This hypothesis was tested by developing a new genome browser (VisGenome) and a new scaling algorithm (CartoonPlus). To identify which type of user evaluation yields more results, we tested two types of user study techniques and compared structured and field-based user studies.

Acknowledgements

I would like to thank my family for constant support, the participants of the experiments I conducted, Helen Purchase who made many useful suggestions, and Gary Gray and Naveed Khan for technical assistance.

Especially I would like to thank Ela Hunt and Matthew Chalmers for supervision and Anna Dominiczak and her team at the BHF GCRC (British Heart Foundation Glasgow Cardiovascular Research Centre) for collaboration in this work.

Declaration of Originality

The material presented in this thesis is the result of my own research carried out at the department of Computing Science at the University of Glasgow working under the supervision of Dr. Ela Hunt and Dr. Matthew Chalmers, except where explicitly stated otherwise. All other referenced material has been given full acknowledgement in the text.

Publications and Presentations

Refereed Publications

1. J. Jakubowska, E. Hunt, J. McClure, M. Chalmers, M. McBride, and A. F. Dominiczak. VisGenome and Ensembl: Usability of Integrated Genome Maps. Data Integration in the Life Sciences (DILS) proceedings (one of five best papers), LNCS 5109, Springer, 77-91, 2008.
2. J. Jakubowska, E. Hunt, and M. Chalmers. CartoonPlus: A New Scaling Algorithm for Genomics Data. International Conference on Computational Science (ICCS) proceedings, LNCS 5103, Springer, 158-167, 2008.
3. J. Jakubowska, E. Hunt, M. Chalmers, M. McBride, and A. F. Dominiczak. VisGenome: visualization of single and comparative genome representations. *Bioinformatics* vol. 23, no. 19, 2641-2642, 2007.
4. E. Hunt, J. Jakubowska, C. Boesinger, and M. Norrie. Defining Mapping Mashups with BioXMash. *Journal of Integrative Bioinformatics*, 4(3):64, 2007.
5. J. Jakubowska. Genome Visualisation. HCI Engage, Doctoral Consortium, 2006.

Other Publications

1. J. Jakubowska, E. Hunt, and M. Chalmers. VisGenome with CartoonPlus: a New Scaling Algorithm for Genomics Data. DILS'08, poster/demo.
2. J. Jakubowska, E. Hunt, and M. Chalmers. CartoonPlus: A New Scaling Algorithm for Genomics Data. Department of Computing Science at the University of Glasgow, Tech. Report: TR-2007-259, 2007. http://www.dcs.gla.ac.uk/publications/PAPERS/8757/VG_VisTech.pdf
3. J. Jakubowska, J. McClure, E. Hunt, and M. Chalmers. Usability of VisGenome and Ensembl A User Study. Department of Computing Science at the University of Glasgow, Tech. Report: TR-2007-244, 2007. http://www.dcs.gla.ac.uk/publications/PAPERS/8510/VG&Ens_TechRep.pdf

4. J. Jakubowska, E. Hunt, and M. Chalmers. Granularity of genomics data in genome visualisation. Department of Computing Science at the University of Glasgow, Tech. Report: TR-2006-221, 2006. <http://www.dcs.gla.ac.uk/publications/PAPERS/8212/AsiaElaMatthewCHI06.pdf>
5. J. Jakubowska, E. Hunt, M. Chalmers, and A. F. Dominiczak. VisGenome genome visualisation at different levels of detail. 3rd International Workshop on Data Integration in the Life Sciences 2006 (DILS'06), poster.
6. J. Jakubowska, E. Hunt, M. Chalmers, D. Leader, M. McBride, and A. F. Dominiczak. System level visualization of eQTLs and pQTLs. BioSysBio: Bioinformatics and Systems Biology Conference in Edinburgh 14-15 July 2005, BMC Bioinformatics, poster.
7. E. Hunt, K. Renaud, R. Sinnott, and J. Jakubowska. Migrating data integration agents for genomics. Workshop on Ubiquitous Computing and e-Research in Edinburgh, 2005.

Presentations

1. VisGenome and Ensembl: Usability of Integrated Genome Maps - DILS'08 in Paris (26th June 2008).
2. CartoonPlus: A New Scaling Algorithm for Genomics Data - ICCS'08 in Krakw (24th June 2008).
3. User Studies in Genome Visualisation - presented at research seminar (SET - Software, Engineering and Software Technology and Teaching Special Interest Group) in Glasgow (14th January 2008).
4. Genome Visualisation - presented at research seminars (GIST - Glasgow Interactive Systems Group and BRC - Bioinformatics Research Centre) in Glasgow (31st May 2007 - GIST, 1st June 2006 - GIST, 3rd November 2006 - BRC), at Napier University (4th May 2007), at ETH Zurich (3rd May 2006) and at HCI Engage Doctoral Consortium (11th September 2006).
5. Visualisation for Functional Genomics and Drug Target Discovery - presented at a research seminar (SET) in Glasgow (25th November 2005).

Contents

Contents	vii
1 Introduction	1
1.1 Motivation	2
1.2 Aims	2
1.3 Methods	2
1.4 Contributions	3
1.5 Thesis Organisation	3
2 Visualisation Background	5
2.1 Introduction	5
2.2 Visualisation Research	6
2.2.1 Data Types and Visual Representations	7
2.2.2 Interaction Techniques	18
2.3 Summary	30
2.4 Conclusion	31
3 Genome Browsers	32
3.1 Introduction	32
3.2 A Classification System	34
3.3 Survey of Genome Browsers	38
3.3.1 Classic Genome Browsers	40

3.3.2	Metabolic Pathways	63
3.3.3	Other Data Analysis Tools	65
3.4	Work Undertaken with Existing Genome Browsers	67
3.4.1	Software Re-engineering for Database Connectivity in SyntenyVista	67
3.4.2	Software Development of DerBrowser	68
3.5	Conclusion	72
4	HCI - Design and Evaluation	74
4.1	Introduction	74
4.2	Design	76
4.3	Evaluation	78
4.3.1	User Studies in Visualisation	83
4.3.2	User Studies in Bioinformatics	85
4.4	Our Users	86
4.5	Medical Researchers' Activities	86
4.5.1	Human Studies	87
4.5.2	Animal Work	87
4.5.3	Office Work	91
4.6	Conclusion	91
5	VisGenome	92
5.1	Introduction	92
5.2	Features	93
5.2.1	Navigation	93
5.2.2	Zooming and Panning	95
5.2.3	Marking a Region of Interest	95
5.2.4	Additional Information	95
5.2.5	Supporting Data	96
5.2.6	Implementation	96

5.2.7	Availability	97
5.3	Caching	97
5.4	Java Web Start	98
5.5	Package Structure	98
5.6	Conclusion	100
6	Initial Quantitative User Study	101
6.1	Introduction	101
6.2	A User Study	101
6.2.1	Participants	102
6.2.2	Methods	102
6.2.3	Search Tasks	103
6.2.4	Questionnaire	104
6.2.5	Interview Questions	104
6.2.6	Task Benchmark	105
6.3	Experimental Results	107
6.3.1	Accuracy and Task Completion	108
6.3.2	Time to Finish	108
6.3.3	Mouse Clicks	109
6.3.4	The User Questionnaire Results	109
6.3.5	Additional Interview Questions	111
6.4	The Frequent Errors	113
6.5	Conclusion	114
7	VisGenome - Extension	116
7.1	Introduction	116
7.2	Visualisation Extensions	117
7.2.1	Homologies	117
7.2.2	Labelling	118

7.2.3	Supporting Data	118
7.2.4	Focus On	118
7.2.5	Colours	119
7.2.6	Additional Information	120
7.2.7	Scaling Algorithm	121
7.3	Conclusion	123
8	Mixed Paradigm User Study	125
8.1	Introduction	125
8.2	User Study	126
8.2.1	Participants	126
8.2.2	Methodology	127
8.3	Experimental Results	130
8.3.1	Overview	130
8.3.2	Log Files Results	137
8.3.3	User Questionnaire Results	138
8.3.4	Video Recording	139
8.3.5	Interview After the Experiment	141
8.3.6	User Diary Results	142
8.3.7	Observations	143
8.3.8	Summary	147
8.4	Conclusion	153
9	Discussion	155
9.1	Weaknesses and Strengths of Both User Studies	155
9.1.1	Weaknesses	156
9.1.2	Strengths	157
9.2	Visualisation of Genome Data	158
9.2.1	Genome Browsers	158

9.2.2	Visualisation Techniques	159
9.3	Conclusion	160
10	Future Work	161
10.1	Introduction	161
10.2	VisGenome	162
10.3	A New User Study	165
10.4	Conclusion	166
11	Conclusion	167
11.1	Contributions	167
11.2	Closing	169
	Glossary	170
	Bibliography	181
	Appendices	192

Appendices

Appendix A:

VisGenome and Ensembl: Usability of Integrated Genome Maps (DILS'08).

Appendix B:

CartoonPlus: A New Scaling Algorithm for Genomics Data (ICCS'08).

Appendix C:

VisGenome: visualisation of single and comparative representations (Bioinformatics'07).

Appendix D:

Genome Visualisation (HCI Engage'06).

Appendix E:

Usability of VisGenome and Ensembl - A User Study (Technical Report'07).

Appendix F:

Granularity of genomics data in genome visualisation (Technical Report'06).

Appendix G:

VisGenome User Manual.

Appendix H:

Ethics Committee Form.

Appendix I:

Participant Consent Form (Initial Quantitative User Study).

Appendix J:

Participant Consent Form (Mixed Paradigm User Study).

Appendix K:

Questionnaire (Initial Quantitative User Study).

Appendix L:

Workload Tests (Initial Quantitative User Study).

Appendix M:

Diary (Mixed Paradigm User Study).

Appendix N:

Interview Form (Mixed Paradigm User Study).

Appendix O:

Statistical Methods.

Appendix P:

Raw Data from Log Files (Mixed Paradigm User Study).

Chapter 1

Introduction

Computer technologies permeate our lives. Computers are used in different science areas and help both old and young people. Medical researchers are trying to find new weapons to fight diseases, and improved computer technologies can be very helpful. Visualisation of genome comparisons is an important research tool in biology and medicine, but none of the genome browsers we studied (see Chapter 3) fulfilled the requirement of showing all the data needed for the biologists' work in one display, or allowed the medical researchers to study QTLs (quantitative trait loci) and other relevant data, which could help them to find disease causes and treatments. Our research aims to improve upon the existing computer technologies and make them more usable. We¹ studied visualisation techniques and conducted user studies, in order to better understand the existing problems and support biomedical research.

Biologists produce a large number of biological data types which are very difficult to visualise. Some research centres collect the data and allow the biologists to exchange and analyse it. However, the computer solutions they produce are often slow and not intuitive. As a result, medical researchers cannot add their new data easily to the existing visualisations or cannot see all the data in a way that best supports their work.

During my PhD I experimented with existing visualisation tools and developed a genome browser - VisGenome (see Chapters 5 and 7) - which shows genetic data in overview and detail. I implemented a new scaling algorithm CartoonPlus (see Chapter 7). A controlled experiment involving biomedical researchers comparing the new browser, VisGenome, and Ensembl was the next part of my PhD. The experiment was designed as a response to the lack of visualisation support for candidate gene study. The experiment showed that neither Ensembl nor VisGenome was perfect. However, the subjects were more successful in

¹In the thesis we mainly used first person plural (we), to describe the thesis author's findings. However, in some situations to stress that some part of the work was being carried out by the author (particularly the user studies) we used first person singular (I) while referring to the author.

VisGenome, and they liked the mouse manipulation offering improved navigation through biological data. The experiment led us to improve VisGenome and conduct another experiment (Chapter 8). Following suggestions from HCI (human-computer interaction) specialists, the second user study was conducted in the biologists' everyday environment. The study showed how biologists use a number of different tools in their work to carry out different activities and use those alongside visualisation tools, and revealed strengths and weaknesses of VisGenome.

1.1 Motivation

Motivation for this work came from cooperation with medical researchers from the British Heart Foundation Glasgow Cardiovascular Research Centre (BHF GCRC), who work with animals and try to find genes responsible for hypertension. They use a number of applications to visualise their data, but none of them fulfil all their requirements. On the other hand, there is a number of visualisation techniques available but only small number of them is used in biology. Therefore, we decided to develop a universal genome browser and evaluate it via user studies.

1.2 Aims

Visualisation techniques help medical researchers do their work more effectively and present different resolutions of data. Our research aims to help the biologists to find genes responsible for diseases, especially for hypertension. An application for genome visualisation, VisGenome, which compares data from different sources and presents various degrees of granularity of genomic data was developed. The medical researchers from the BHF Glasgow Cardiovascular Research Centre are using the tool now for analysing their research data. The application was evaluated during user studies with the biologists from the hospital. The aim of the user studies was to find out what visualisation techniques are useful and what could be improved in genome browsers and how. We also wanted to see how the new visualisation technique, CartoonPlus, is used for biological data, and if is helpful or not. A more general question we tackled was how biologists use various sources of data and how computer tools help them to do their work more effectively and efficiently.

1.3 Methods

We employed the methodology of iterative software development. In user studies we used two methods: a structured user study with tasks, followed by a questionnaire and interview; and an open-ended folkloristic (environmentally valid) user study. More details are given in Chapters 6 and 8.

1.4 Contributions

This thesis makes the following contributions related to HCI, visualisation, and bioinformatics.

- VisGenome, which is a novel genome browser, was designed and implemented with the needs of the BHF users in mind. Its usefulness and usability was assessed. We demonstrated that the new visualisation solution is useful and better than the existing solution.
- A novel scaling algorithm, CartoonPlus, was designed, implemented and tested via a user study. It was found to be very useful, especially for small items of data.
- A novel genome browser classification was suggested and a number of known genome browsers were positioned within this classification.
- Feedback from the user studies contributes to human-computer interaction. The user studies extended our understanding of user requirements in this application area. A structured user study confirmed that the application is useful and how it could be extended. The mixed paradigm user study showed constraints resulting from the working environment.

1.5 Thesis Organisation

This thesis is organised as follows:

- Chapters 2, 3, and 4 outline background knowledge from visualisation, biology, and human-computer interaction which forms the foundation of our research. Chapter 2 ('Visualisation Background') introduces the field of information visualisation. Attention is paid to techniques used in genome browsers and to interaction techniques, especially the ones used in the user studies. Chapter 3 ('Genome Browsers') studies the known problems in biological data visualisation and introduces the most popular genome browsers which are used by our collaborators. After carrying out our literature survey in visualisation, biology, and human-computer interaction, we decided to improve some existing genome browsers what is described in the chapter. A novel genome browser classification is presented. This material was published as a technical report [48]. Chapter 4 ('HCI - Design and Evaluation') summarises basic knowledge from the fields of human-computer interaction and present results from observation medical researchers from the BHF Glasgow Cardiovascular Research Centre during their work. The work introduces the user, who takes part in further experiments.
- Chapter 5 ('VisGenome') presents the development of VisGenome. The work was published as an application note in Bioinformatics [51]. The application is available via Java Web Start from <http://www.dcs.gla.ac.uk/~asia/VisGenome>. After developing a new application, we evaluated it

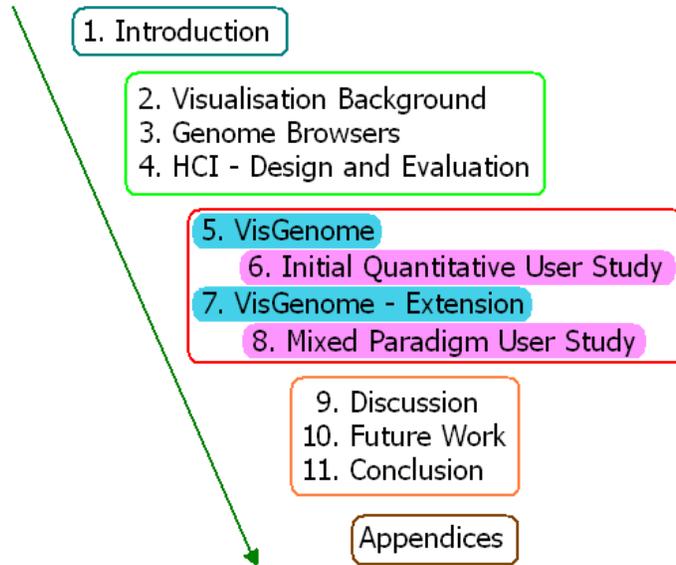


Figure 1.1: Thesis structure. The chapters are divided into 5 blocks: *Introduction* - introduces the research work; *Chapters 2, 3, and 4* present background knowledge and introduce research with existing genome browsers and describes our users; *Chapters 5, 6, 7, and 8* present the work on VisGenome (*‘VisGenome’* and *‘VisGenome - Extension’*) and the user studies (*‘Initial Quantitative User Study’* and *‘Mixed Paradigm User Study’*); *Chapters 9, 10, and 11* discuss, present future work, and conclude.

during a user study. Chapter 6 (*‘Initial Quantitative User Study’*) describes an experiment which was conducted with the biological researchers from the Western Infirmary and the Bioinformatics Research Centre at Glasgow. Parts of this chapter were published as a technical report [53] and in the DILS’08 proceedings [52]. After the first experiment we improved VisGenome according to user feedback and added a novel CartoonPlus algorithm. The new version of the application is described in Chapter 7 (*‘VisGenome - Extension’*). CartoonPlus was published as a technical report [49] and in the ICCS’08 proceedings [50]. The second version of VisGenome was also evaluated via an user study, presented in Chapter 8 (*‘Mixed Paradigm User Study’*). Chapter 8 contains an analysis of results from the study.

- Chapter 9 (*‘Discussion’*) discusses the research. Chapter 10 (*‘Future Work’*) provides possible directions for future work. Chapter 11 (*‘Conclusion’*) concludes the thesis.

Chapter 2

Visualisation Background

This chapter describes the field of visualisation, the area of the research that forms the foundation of our research in genome visualisation. It briefly defines information visualisation (InfoVis), and summarises the techniques, especially the ones used in genome browsers. The most popular genome browsers with their visualisation techniques are presented in the next chapter of the thesis.

2.1 Introduction

Visualisation aims to provide insight into user data, especially in the exploration and analysis of very large data-sets, such large data sets are profuse in genomics. A very common situation, in biology and elsewhere, is a high volume of information that is to be understood and a basic question facing the researchers: how to understand and organise the entire data set. Visualisation is a possible solution in this situation.

We can find a short definition of visualisation in [11]. It defines visualisation as “the use of computer-supported, interactive, visual representations of data to amplify cognition”. The discussion clearly states that visualisation - meaning scientific visualisation in this case - is concerned with non-abstract (concrete) data sets. In [69], the authors stress that visualisation used in scientific tools helps researchers to understand and steer computations. Since then (1987) visualisation is recognised as a research field or rather as a sub-field of science, statistics, and graphics. In the mid-to-late 80s, the existing techniques were limited by machines, computational power and screen space, but visualisation could solve a number of problems in representing large amounts of data. Interactive computer graphics were first used in 1964 with chemical data [69]. Since then computer graphics have been an integral part of all fields in academic and industrial research. However, work in data graphics is dated from the time of Playfair [112], which is about 1786. Playfair is considered to be the person who laid the foundation for the art and science of statistical diagrams (see Figure 2.1).

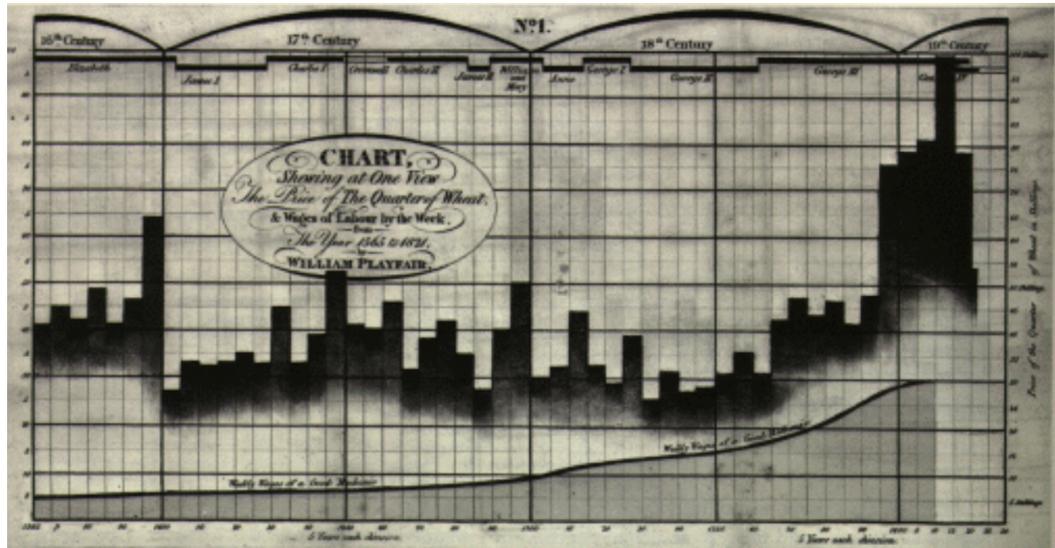


Figure 2.1: Playfair’s parallel time-series bar chart of prices of wheat, wages and monarchs over 250+ years. The picture is taken from Playfair - “A Letter on our Agricultural Distress, their Causes and Remedies.”[112].

The InfoVis definition in [11] presents information visualisation as “the use of computer-supported, interactive, visual representations of abstract data to amplify cognition”. This clearly suggests that InfoVis is concerned with making useful aspects of an abstract set of information visible. The term InfoVis was first used in [12] in 1991. In the same publication, the authors presented a definition of scientific visualisation as “use of interactive visual representations of scientific data, typically physically based, to amplify cognition.” In the same way we can define genome visualisation as use of interactive visual representations of genomics data to amplify cognition. Genome visualisation is simply visualisation applied to genomics data and it could be placed on the boundary of scientific visualisation. However, we found it useful to place genome visualisation in the context of InfoVis because all techniques used for abstract data could be also applied to non-abstract data, and the solution might produce new useful results. Therefore in this chapter we present visualisation techniques, mainly InfoVis techniques that could be applied to genomics data.

2.2 Visualisation Research

InfoVis offers many different visualisation techniques, and a number of them are strictly concerned with data representation. Shneiderman in [102] offered the Type by Task Taxonomy (TTT) of information visualisations. He presents seven data types (1D, 2D, 3D, temporal, multi-dimensional, tree, and network data) and seven tasks at a high level of abstraction (overview, zoom, filter, details-on-demand, relate,

history, and extract). Each type from the data type taxonomy has attributes and a basic task. The research work presented in this thesis is strictly concerned with 1D, 2D and multi-dimensional data, therefore we mainly focus on these three data types. However, in the section ‘Data Types and Visual Representations’ we survey existing data types and common graphical data representations for each data type. The section ‘Interaction Techniques’ presents the tasks (according to Shneiderman’s classification [102]) - and interaction techniques which could be used to manipulate the data types.

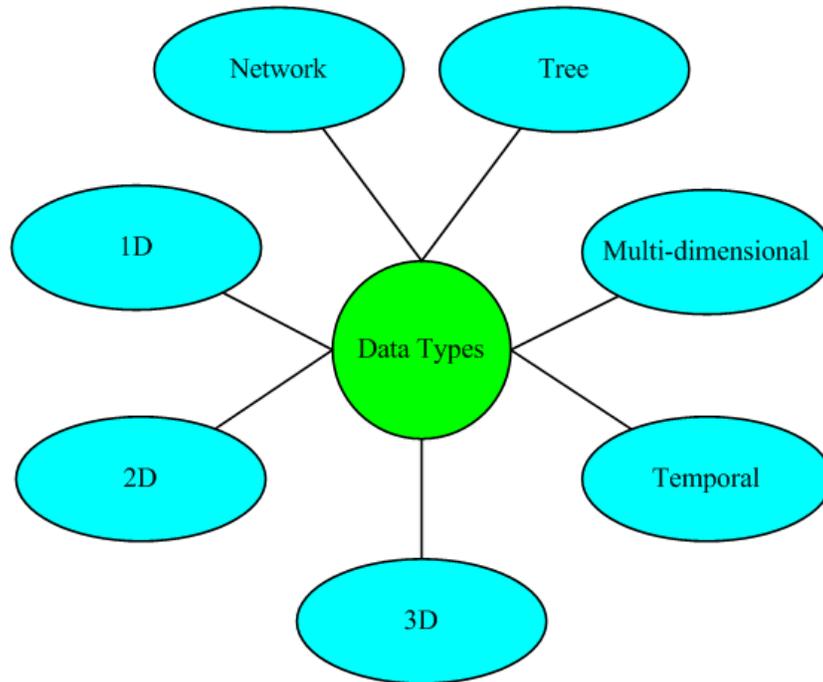


Figure 2.2: Seven data types according to Shneiderman’s classification [102].

2.2.1 Data Types and Visual Representations

This section presents data types, based on Shneiderman’s classification [102], see Figure 2.2, and common graphic data representations that could be applied to the data types. We survey all seven categories defined by Shneiderman, however, we present in more detail only the ones important in genomics and biological data. The most popular interaction techniques are sketched in the following sections. It is important to remember that one graphic data representation could be used for a few data types. The same applies to interaction techniques, as each interaction technique could potentially be implemented for any data type.

1D (1-dimensional) data consists of linear data types. In [102] the author provides samples of linear data: textual documents or program source code. Linear data is very common in genomics. As

an example, here, we could take lists of genes organised alphabetically. On the other hand, instead of alphabetical order, we could also use the order in which the genes appear on a chromosome. Biologists have not sequenced all genetic data yet. They conduct experiments and discover new genes. Therefore, additional attributes, in our example, could be the date of gene discovery and the discoverer name. In 1D data, the users have a common problem with finding specific information, for example, all genes discovered by Mr X or the number of genes discovered before 1995. The most significant problem with 1D data is also its high volume. Users may get lost in the data when they try to find a specific item. It is very important to provide user-friendly manipulation techniques which allow the users to find the information they need to see. Some search tools which could help the users find interesting items in 1D data types are the perspective wall [64] or the alphaslides [1]. Both are described in the next section - 'Interaction Techniques'.

As an example of 1D data with annotations we present genetic or physical maps, see Figure 2.3, which are the fundamental organisational framework for all the genome databases. Both genetic and physical maps provide order of items along a chromosome. A genetic map provides an estimate of the genetic distance between two items and is limited to ordering these items. We could say, for example, that gene A lies between marker X and marker Y on genetic map for chromosome Z. On the other hand, physical maps mark a position on a chromosome for the true distance measured in base pairs (bp), between

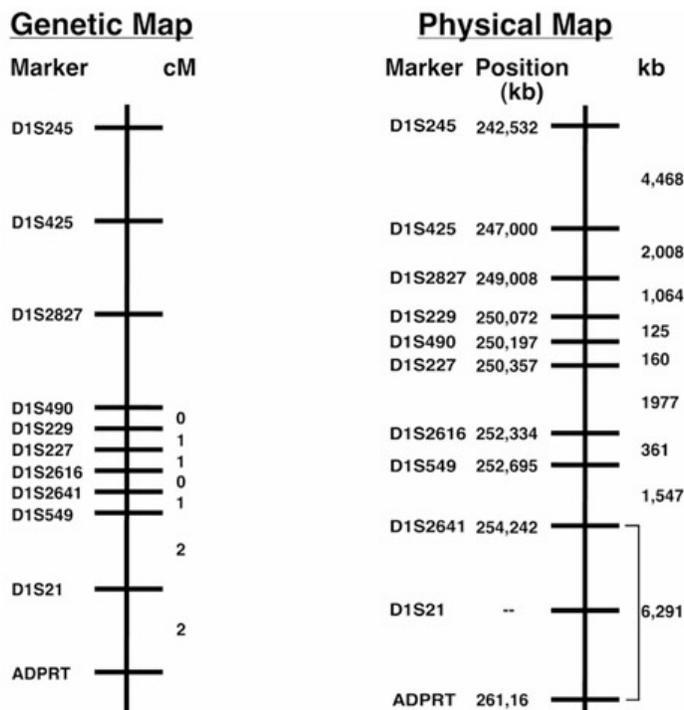


Figure 2.3: Genetic and physical maps of the 1q3242 region. The figure is taken from [36].

items of interest. In this situation, we could say that our gene A lies between 100 bp and 200 bp. Physical maps are more precise than genetic maps.

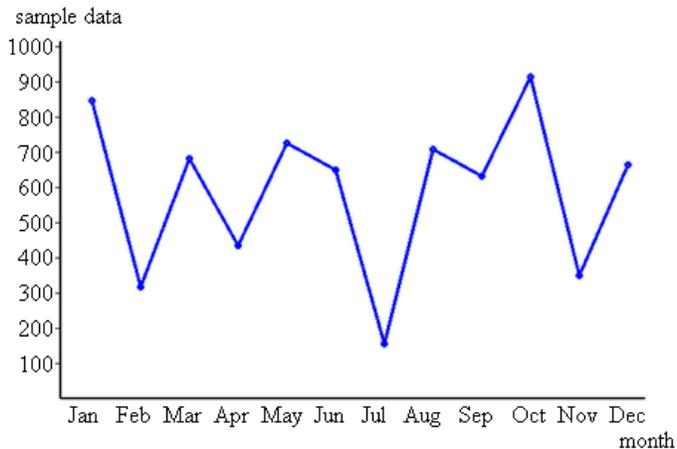


Figure 2.4: An example of a line graph representation. The one presented here represents a 2D data type.

A **2D** (2-dimensional) data type contains planar or map data. A number of visual data representations can be used for representing 2D data types. The simplest one is a line graph, presented in Fig. 2.4, and a bar graph, see Figure 2.1. The bar graph was introduced by Playfair [112] and originally helped to show series of data where values were not connected to one another, or had missing data. A line graph is commonly used in genomics, as in most other areas, see Figure 3.14. Scatter plots, see Figure 2.5, are similar to line graphs in that they use horizontal and vertical axes to plot data points (for 2D data type). Scatter plots on Figure 2.5 are shown in 2D and 3D, but they also could be shown in 1D - a line graph with one axis only. Another data representation used in a 2D environment is a box plot [70], which shows

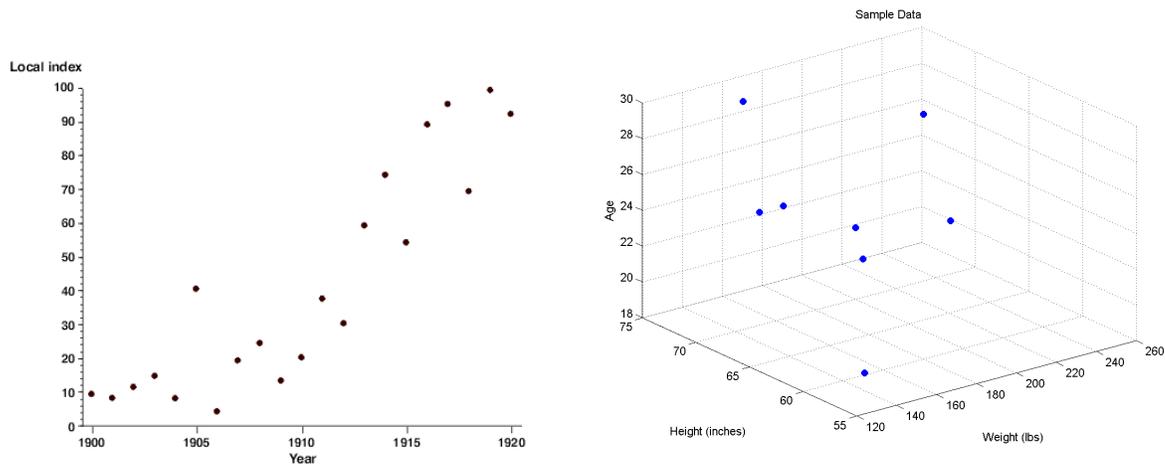


Figure 2.5: 2D and 3D scatter plots.

a 2D plot with a point and its standard deviation.

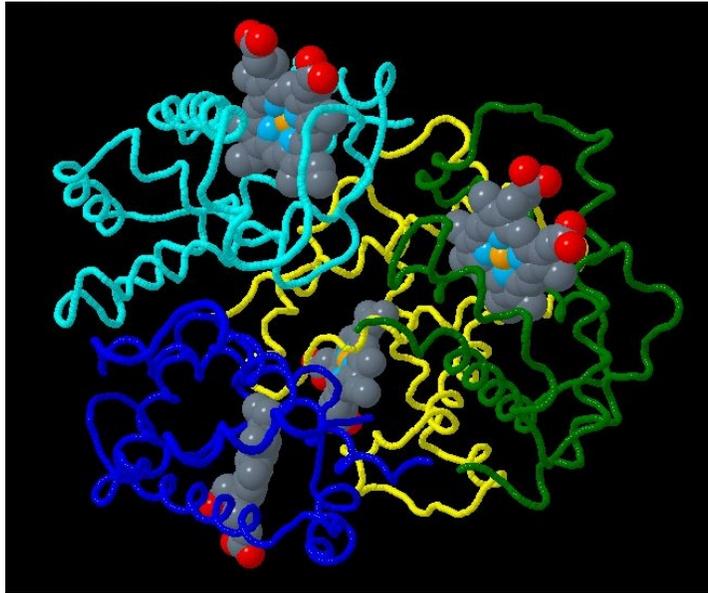


Figure 2.6: An interactive 3D visualisation presented by Jmol. The figure is taken from [139].

A **3D** (3-dimensional) data type contains real-world objects such as the human body or buildings which have items with 3 dimensional relationships. A 3D data representation is only rarely used for abstract data in biology. However, Jmol [40], see Figure 2.6, displays molecules in 3D as molecules naturally possess a 3D structure. In the next chapter, when we survey the existing genome browsers, we see that only a small number of tools for 3D gene representation are available. Therefore, we do not focus on this representation. 3D data representation is not common for abstract data, because it makes the data representation unclear and is not easy to navigate. We suggest that this is the most likely reason why such a small number of tools for 3D gene representation is on offer.

Temporal data is common in medical records or historical presentations. The difference between temporal data and 1D data type is start and end time for each item. The items could also overlap. Micro array time series and yeast cycles [106] are good examples of temporal data in genomics. We see that biological data could oscillate, be periodic, and the periods might have biological significance. The data (genes) have active periods when they can influence other active genomics data - overlapping items. A gene could be also active for a short time which is marked as a point in time. The time-series explorer presented in [22] also visualises temporal data. The authors show timing, activity and change-in-activity of micro array time-course data, which relate to the recorded activity of genes in time during a biological process.

Another example of temporal data is LifeLines [85] which visualises summaries of personal histories. The authors present in a single screen different facets of people's life. Figure 2.7 shows stories and aspects

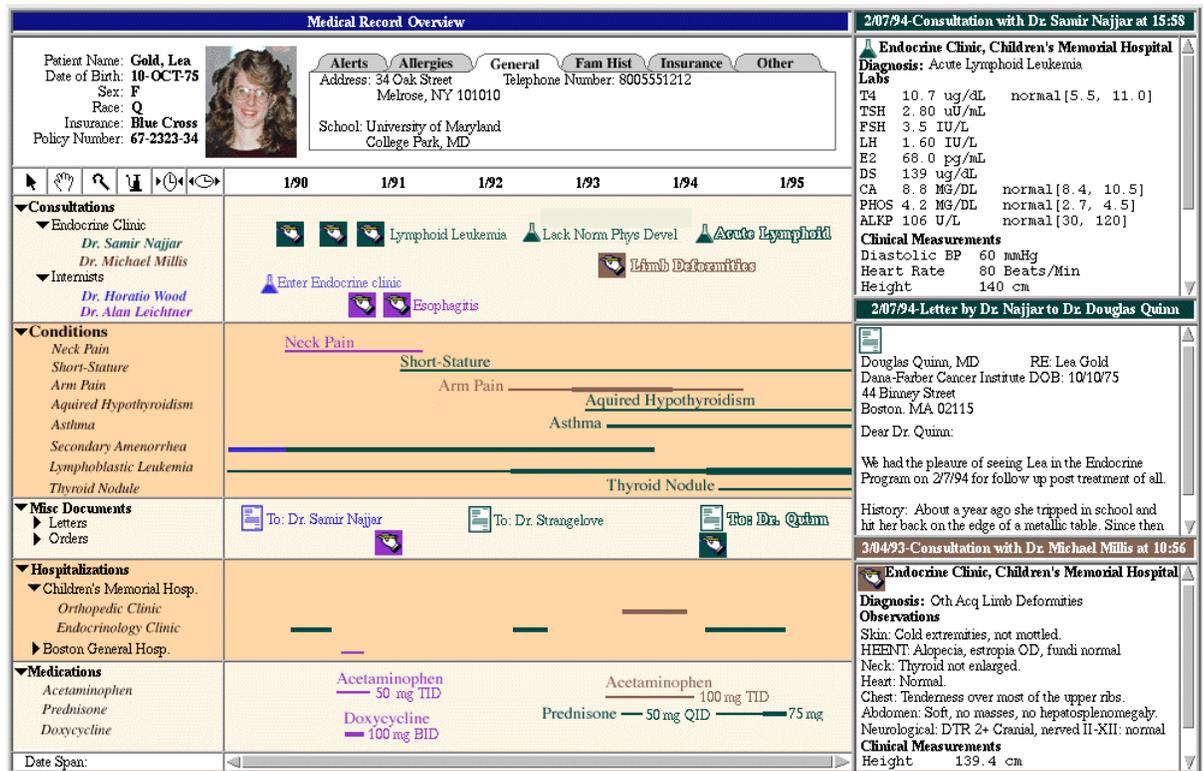


Figure 2.7: A screenshot from [85] presenting the medical record for Lea Gold.

(consultations, conditions, misc documents, hospitalizations, and medications) which include events (for example: neck pain, asthma, or doxycycline) and periods (2 weeks of neck pain), of Lea Gold's life. Lines which represent stories and aspects have different colours and sizes suited to periods. Each facet's background has a different colour, which makes a view clearer. Moreover, the application offers vertical lines across each year, which together with colours allows the users to find information more easily, especially in zoomed out view. The visualisation also shows discrete events. Icons are used as for example to show lymphoid leukemia, see Figure 2.7. Plaisant and colleagues stress that LifeLines provide a complete visualisation environment to hold such as overview, zooming, filtering, details on demand, colour coding, filtering and dynamic highlighting. The users can see interrelationships between periods and events and all detail information by using mouse clicks.

Nowadays, a number of people need calendars and timed information. They enter data by months or years, so it is very important they do not miss any information and see correlations between previous and actual events.

A **multi-dimensional** data type is composed from n attribute items which become points in an n -dimensional space. As an example for a multi-dimensional data type, we present parallel coordinates [47], see Figure 2.8, where vertical bars represent each dimension. Each element of the data set has values

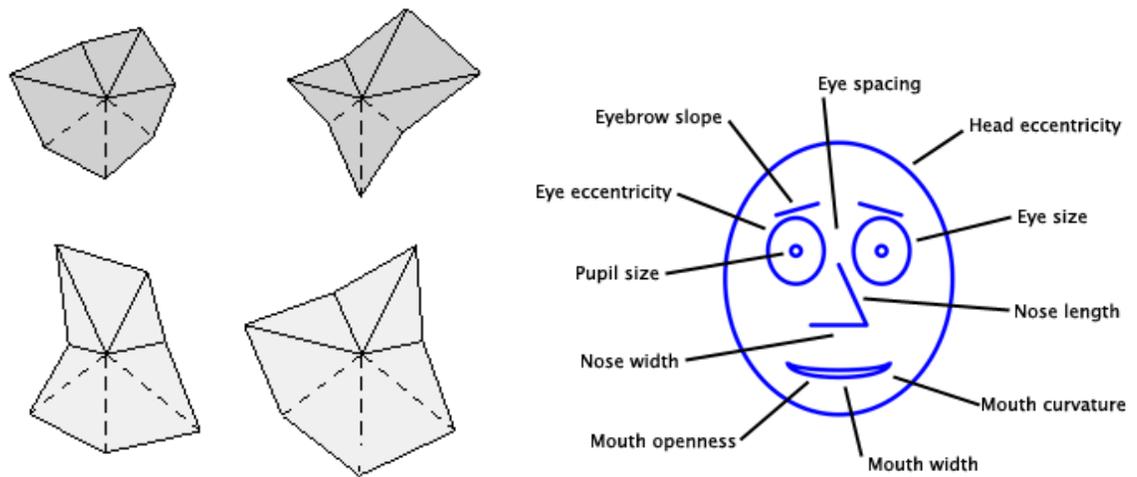
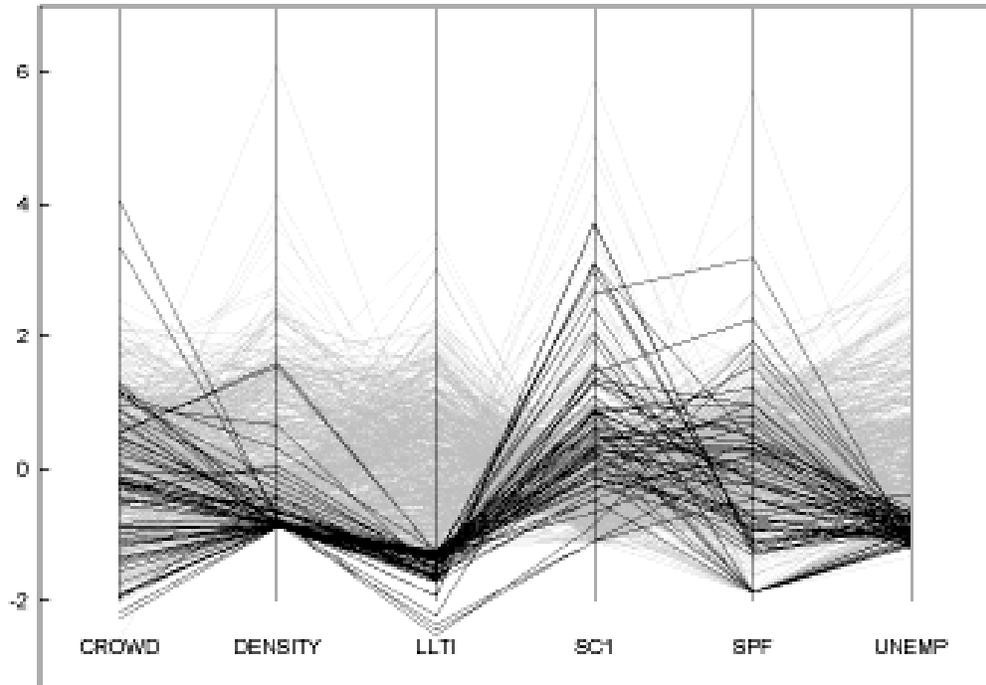


Figure 2.8: Left to right: parallel coordinates [47], star plots [14], which are similar to parallel coordinates in that they show a single record of data where its attributes are shown radially, and a Chernoff face [16] is a method for diagramming multi-dimensional data through the use of facial features. The figure of Chernoff face is taken from [123].

for each dimension, which are shown as points along the vertical axis and then connected together. Star plots and Chernoff faces are other examples of how multi-dimensional data could be represented.

Multi-dimensional data is widely used in genomics. Attributes such as genes, micro array probes, markers, or QTLs are connected with each chromosome. Therefore, we can treat a chromosome as an



Figure 2.9: Tree-Maps [151] and an example of dendrogram [126]. A dendrogram is defined as a stacked tree, where the height of the branches shows an additional variable.

n-attribute item in an n-dimensional genomics space. Each of the attributes has coordinates that allows us to precisely localise it on a chromosome. There are also interrelationships between the attributes. As can be seen in the next chapter, chromosomes cannot be treated as the traditional orthogonal (independent) dimensions, because some of genes from one chromosome are in homology relation with genes from other chromosome from different species.

Trees are very common, especially in biology, because a number of data sets have a hierarchical structure. We define trees as a hierarchically ordered data connected by links or branches. Each tree item has a link to its parent (except the root) and its children or child (except the leaves).

One of the many tree graphical representations is the Tree-Map [54], see Figure 2.9. Tree-Maps were introduced by Johnson and Shneiderman however recently they were evaluated by other developers [151], who introduced additional interaction techniques into Tree-Maps. They found, for example, that this tree representation could be more useful if it allows the users to zoom or to control labelling. The visualisation technique represents hierarchical information in a space-filling manner. All the display space is used. “Tree-Maps partition the display space into a collection of rectangular bounding boxes representing the tree structure” [101]. Johnson and Shneiderman point out that in a typical tree drawing more than 50 percent of the pixels are used as background. This use of display is acceptable only when there is little data. On the other hand, for large trees, traditional node and link diagrams require too much display space.

The authors present nested and non-nested Tree-Maps. Nested Tree-Maps represent tree objects separated at each level. Because of this, the representation occupies more space than non-nested Tree-Maps.

Johnson and Shneiderman stress that a variety of display properties such as colour, texture, shape,

border and blinking determine how the node is drawn within the bounding box. Depending on what kind of information the users require, mapping of information to the Tree-Map display could be different. The authors [54] provide an example where they represent a file hierarchy as a Tree-Map. They find that in a representation where file creation date corresponds to file size and file modification date corresponds to colour, the users can easily localise old files changed recently.

The first version of Tree-Maps presented in [54] used pop-up dialogue windows. The latest versions of the graphical representation offer zooming, sound, hue/saturation control, many border variations, labeling control, dynamic queries filters, and other improvements.

Tree-Maps are also used in biology for showing gene expression results from hierarchical clustering [68]. Baehrecke and colleagues [3] applied Tree-Maps to gene ontology and gene expression data visualisation. They presented data from micro array experiments. Tree-Maps are very effective visualisation techniques in biology, where a high volume of data could be represented within a small area. They are already used for representing gene activities. Tree-Maps are also a potential solution for genomics data in our research, where millions of genes could be visualised. They are not good for representing details but could be used as an overview of the relationships between genes from different species.

Cone tree [94], see Figure 2.10, is a graphical representation for trees. Cone trees are hierarchies laid out uniformly in 3D. Shadows of cones and nodes are projected onto the floor. The transparent display of cones' shadows helps make the cone easily perceived and does not interrupt the view of cones behind it. Unfortunately, transparency is not used for the cones themselves, and because of the 3D representation the user can not see all tree items; some of them are covered by the items close to the front. Shadows of cones give a 3D depth hint for the users and present additional structural information about the tree hierarchy. This additional information is different for cone trees and cam trees. The authors presented cam trees as cone trees lying horizontally instead of vertically, therefore their shadows provide different information about the data. The cone tree shadows present additional information about clustering in the hierarchy. On the other hand, cam tree shadows convey information about the hierarchy of a 2D projection.

The basic manipulation techniques offered by the tree representation are selecting and searching. Cone trees give the users the possibility of selecting a node with the mouse. When a node is selected, the cone tree rotates and the node is brought to the front and highlighted. Moreover, each node in the path from the selected node up to the top is brought to the front and highlighted. When the users search for an element in the cone, all nodes except search matches are made invisible.

Robertson and colleagues introduce 'gardening' operations for the cone tree, which are pruning and growing. The two operations could be done via a menu or directed at a node by gestures. The operations allow the users to select, focus on, or hide some structures in the tree. Similar to Tree-Maps, cone tree, with its interaction techniques, has great potential and could be used for representing genomics data.

Heatmaps, similar to Tree-Maps and cone trees, represent hierarchical structure, see Figure 2.11.

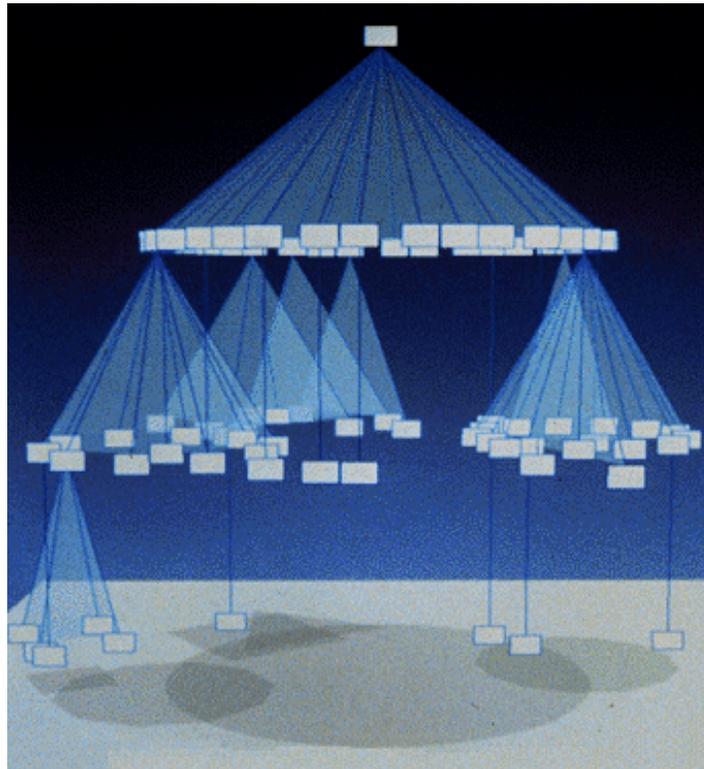
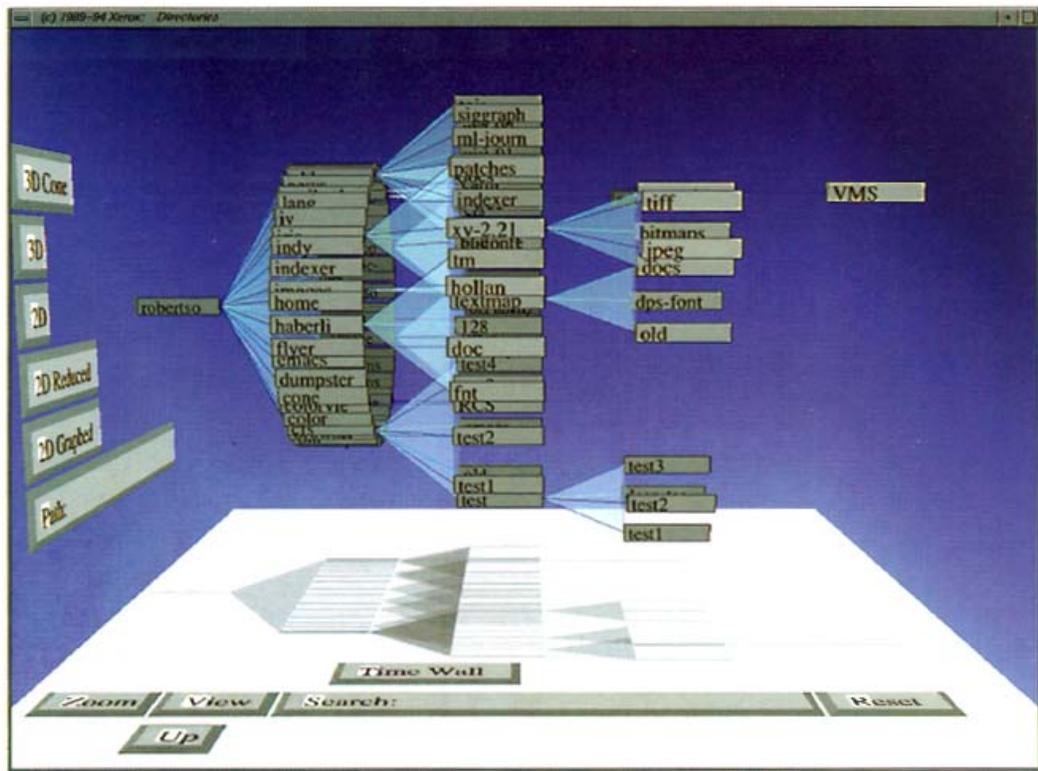


Figure 2.10: Examples of a cam tree (above) and of a cone tree (below) [94].

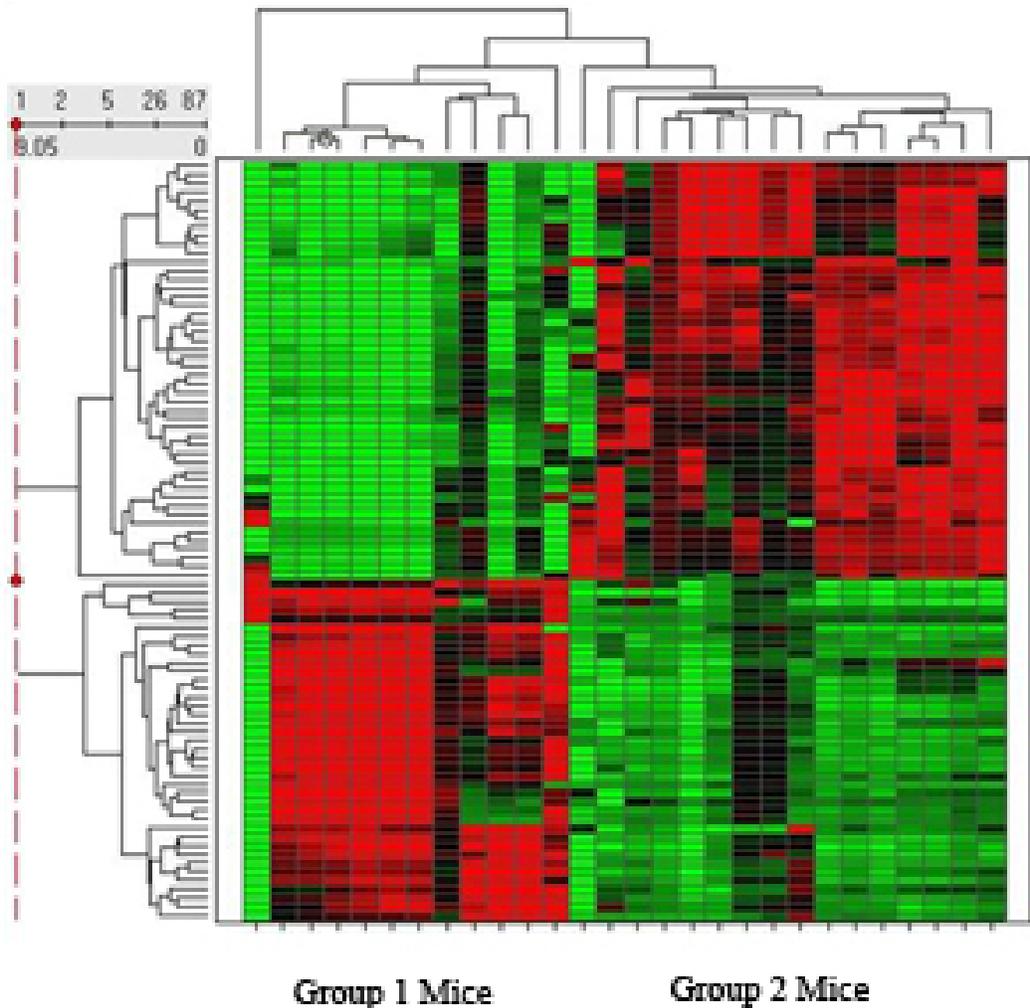


Figure 2.11: An example of a heatmap showing hierarchical clustering results. The figure is adapted from [134].

Heatmaps are very popular in biology and are used to visually display genes, proteins or metabolites. Heatmaps use colour to show an additional dimension.

The last data type in Shneiderman’s classification is a **network**. In some situations a tree structure can be inadequate for expressing relationships between items. In this situation, an item is linked to an arbitrary number of other items. As an example of network data representation we present a graph. The World Wide Web is a network of enormous size and complexity.

Munzner and Burchard [78] present a visualisation of the structure of a section of the World Wide Web (WWW). They constructed a graphical representation in 3D hyperbolic space for a small part of the web. The authors used hyperbolic space because it allows “more information to be seen amid less clutter”. They stress that the web is so huge and interconnected, that it is difficult to study its structure.

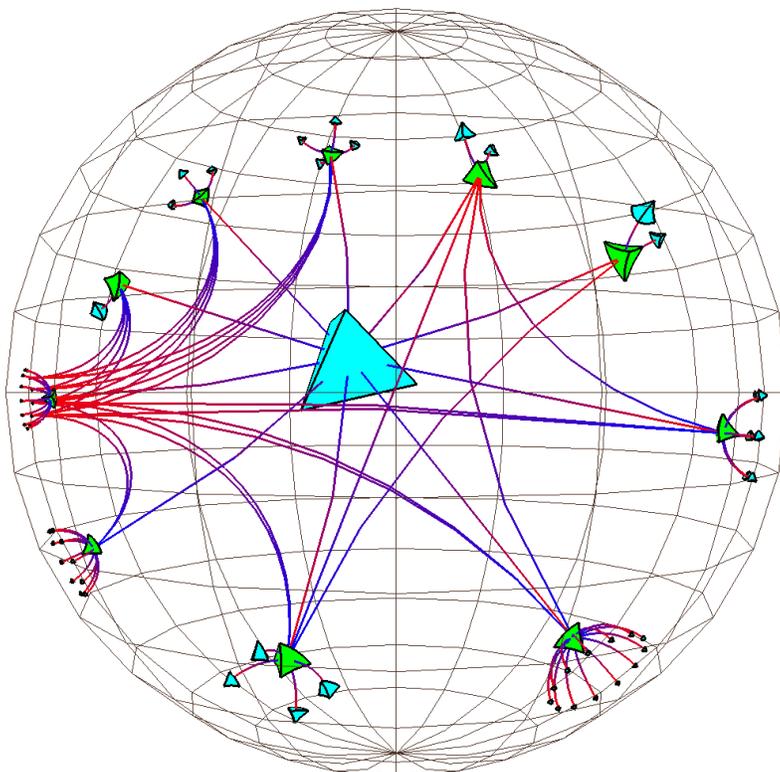


Figure 2.12: “A hierarchical tree structure with links back up the tree” presented by Munzner and Burchard in [78].

Because of this inconvenience, Munzner and Burchard present the WWW as “a hierarchical tree structure with links back up the tree” (a directed graph with cycles), see Figure 2.12. This means that they did not present the real WWW structure, because their structure is much simpler than the real network. On the other hand, they did not show a tree, as they disturb hierarchical ordering by linking back up the tree.

We do not focus on network data used in biology in the thesis, however, it is worth mentioning that this data type is widely used in biomolecular interaction networks. Shannon and colleagues [100], for instance, visualise a biomolecular interaction network and provide basic functionality to lay it out, see Figure 2.13. More information about metabolic pathways and this work is presented in the next chapter.

As can be seen from the discussion presented here, the worlds of data types and graphical data representations are very rich and they contain simple types like line graphs or histograms, more complex types like a permutation matrix (Bertin’s sortable bar charts for the display of multi-dimensional data) or Chernoff faces, as well as very large and complex networks. In this section we presented, through examples, that biological data could be treated as 1D, 2D, 3D, temporal, multi-dimensional, tree, or network data type. However, current traditions and constraints treat biological data mainly as 1D or 2D data types [84, 113]. Many data representations presented here are very interesting and useful, and could

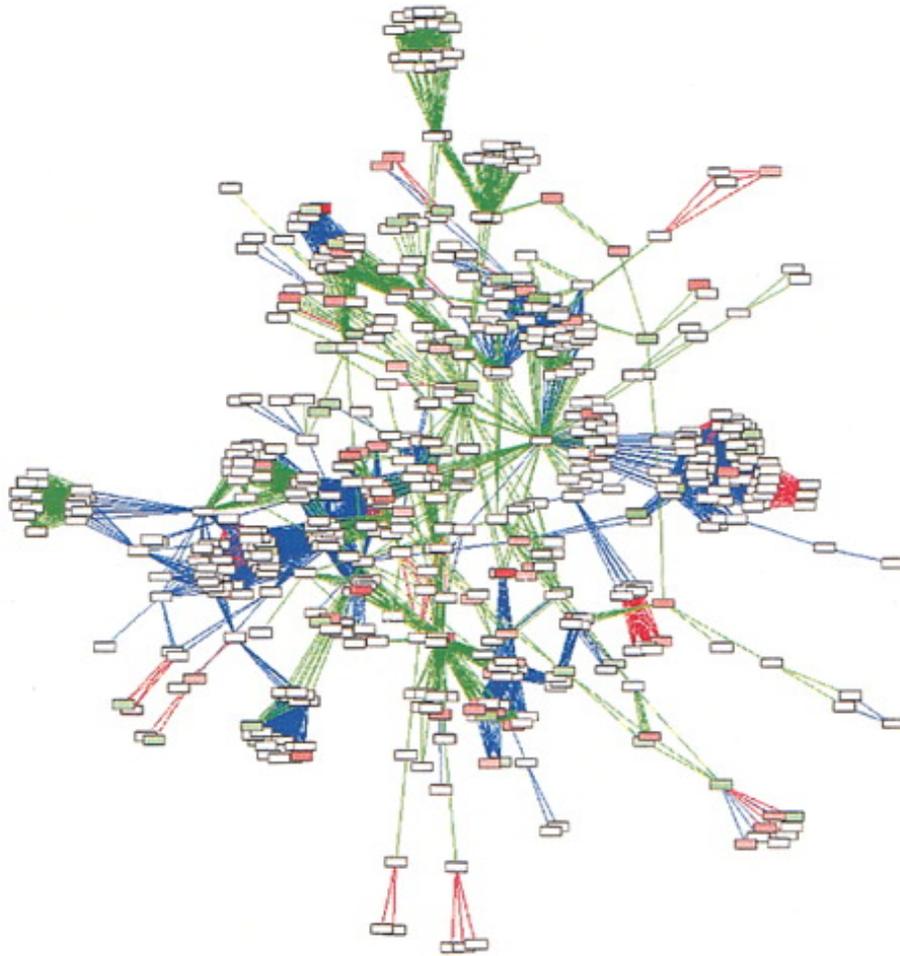


Figure 2.13: The largest connected component of the *Halobacterium* inferred protein network, from [100].

not be discussed separately from interaction techniques. In the thesis we focus on biological data which is mainly represented in 1D, 2D or is multi-dimensional. We added multi-dimensional data to 1D and 2D data because this data type is very popular in genomics, where attributes such as genes, markers or QTLs are connected with each chromosome. In the section ‘Interaction Techniques’ we describe user tasks and interaction techniques used to manipulate data.

2.2.2 Interaction Techniques

Shneiderman [102] presents seven task types, see Figure 2.14, at a high level of abstraction: overview, zoom, filter, details-on-demand, relate, history, and extract. The tasks represent the ways in which the users interact with the data types presented in the previous section. The interaction techniques are an integral part of the thesis, which focuses on the users’ (medical researchers) activities and observes how they work and interact with genome browser applications. This section presents an overview of existing

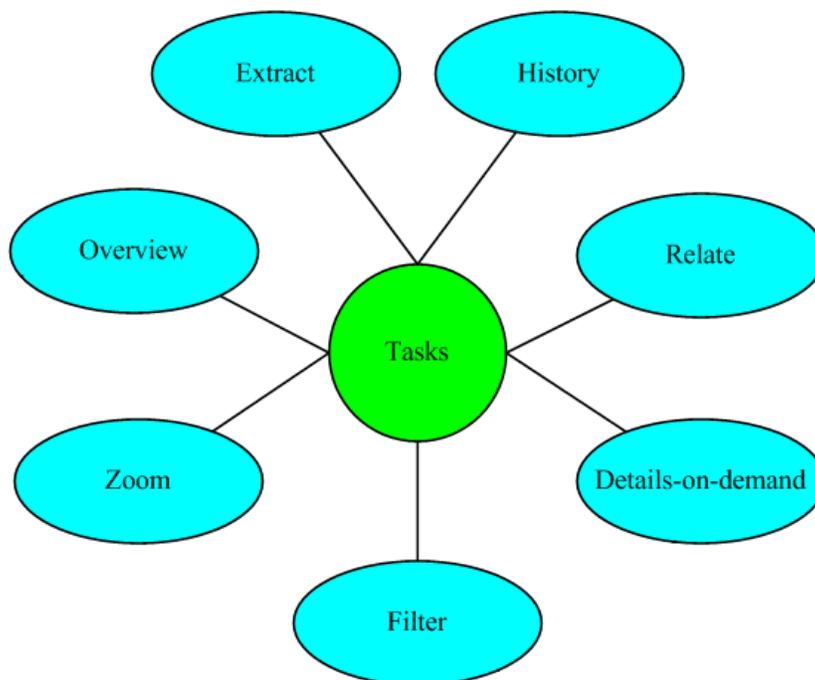


Figure 2.14: Seven task types according to Shneiderman’s classification [102].

interaction techniques and focuses on the ones which are or could be used in genome browsers. The existing genome browsers are presented in the next chapter.

First, we describe **overview techniques**. Because the majority from the overview techniques presented here show not only overview but also context for presented data (fisheye, magic lenses, hyperbolic browser, document lenses), we also call them **focus+context techniques** (not all overview techniques, only the techniques selected here). One of the most popular and widely used is fisheye, see Figure 2.15. Furnas [33] proposed generalised **fisheye views** as a solution to visualise not only information, but also to provide a context in which the information is placed. This technique mimics the perceptual structure of the human eye: it displays local detail and global context simultaneously. Furnas presents a new viewing strategy, “based on an analogy to a very wide angle, or fisheye lens”. This solution allows the software to show places nearby in great detail while still showing areas further away in less detail. The most important idea for the interaction technique is “to provide a balance of local detail and global context”. Furnas formalises generalised fisheye views as views showing only the most interesting points of desired size. He composes a simple formula which allows fisheye views to be defined in any type of structure where the necessary components can be defined. He also suggests that fisheye views should be more useful than other approaches in navigating around or examining unknown parts of a large file. He reports on an experiment which clearly shows that fisheye views show the necessary structural information and the subject does not feel lost in the information space. Furnas [32] presents fisheye views as something

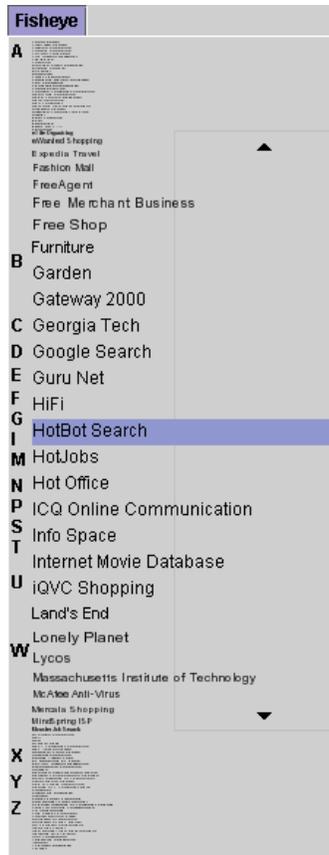


Figure 2.15: Fisheye menu [33, 32].

occurring naturally in many human contexts and which can be used for a wide variety of computer generated information structures.

This interaction technique is widely used in different scientific areas. Toyoda and colleagues [111] used fisheye to visualise biomolecular network graphs. They applied the interaction technique in GSCope, which allowed them to understand biological data structures more effectively.

Bier and colleagues introduced another focus+context technique known as **magic lenses** in [6]. They describe the technique as ‘magic’ filters which, when superposed on an object, could show hidden information, see Figure 2.16. A magic filter is a Toolglass sheet which is situated between an application and a traditional cursor.

Bier used magic lenses for 2D data, but the technique can also be used for different data types. Wang and colleagues [115] propose using the focus+context for pictures, for example X ray photos. They created two kinds of magic lenses: some of which could be configured by the users to specify the desired magnification patterns, while the others were feature-adaptive. They allow the users to control the available screen area by giving more space to more resolution-important features. Wang and colleagues stress that their magic lenses allow the users to highlight and exaggerate an object for the closer inspection of its spatial and



Figure 2.16: The Magic Lenses logo is taken from [6].

semantic context. In a way similar to other focus+context techniques, magic lenses can be used to choose and magnify regions or features of interest to see detail information more clearly, while still showing the context.

Magic lenses were applied to biological data in [86]. Pook and colleagues used this interaction technique for data transformation (viewing under another representation, as for example “transformation of the cytogenetic representation of a chromosome into its Genethon genetic map”) in the application called Zomit. They motivated the solution as intuitive for biologists who are not familiar with programming.

The **perspective wall** described by Mackinlay and colleagues in [64] was one from the first attempts to visualise large 2D data spaces. The authors present the perspective wall technique, see Figure 2.17, which took advantage of hardware support for 3D interactive animation “to imitate the architecture of the eye system”. It folds a 2D layout onto a 3D wall, which integrates an area showing details with two perspective regions for context. This intuitive deformation of the layout allows for more effective space

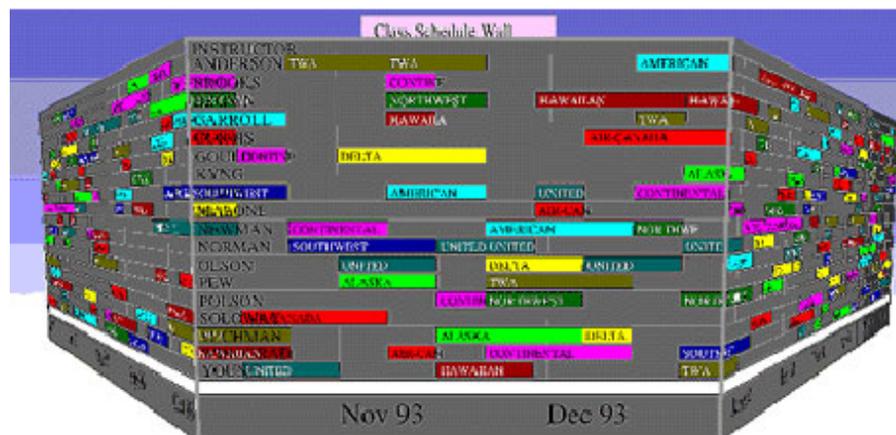


Figure 2.17: The perspective wall uses advances in hardware support for interactive 3D animation to address the integration problems of the bifocal display. A physical metaphor of folding is used to distort an arbitrary 2D layout into a 3D visualisation (wall). The wall has a panel in the center for viewing details and two perspective panels on either side for viewing context. The figure is taken from www.parc.xerox.com.

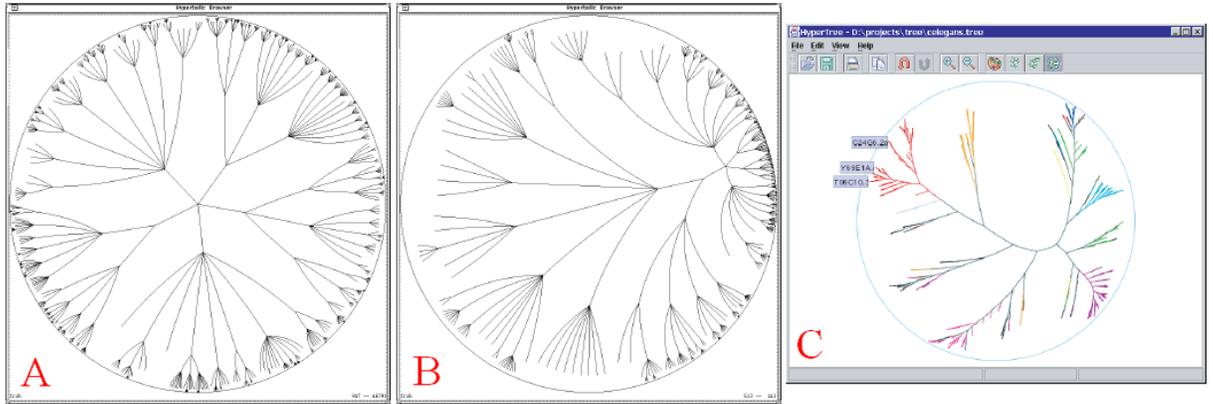


Figure 2.18: A and B show the hyperbolic browser [59]. C shows HyperTree screenshot from [7].

utilisation and smooth transitions of views.

Lamping and Rao [59] developed a **hyperbolic browser** which is a focus+context technique based on hyperbolic geometry. This browser was designed to visualise and manipulate large data sets. The main point of the technique is to lay out the data hierarchy on the hyperbolic plane and map this plane onto a display region. The hyperbolic browser reserves more display space for a focused part of the hierarchy, but it still displays the context of the entire hierarchy. The authors developed procedures for focus manipulation by using pointer clicks as well as interactive dragging. The users can click on any visible point to focus on it or drag any visible point to any other position. Lamping and Rao smoothly animated transitions across such manipulations. The browser can handle “arbitrarily large hierarchies supporting a context that includes as many nodes as shown in 3D approaches and with modest computational requirements”. However, as many other techniques, this one also is limited by space and speed. For very complicated and huge hierarchical structures, the presented context could be not so clear. Lamping and Rao have developed a number of variations of the core hyperbolic browser, which can be applied to the common graphs in other applications. Bingham and Sudarsanam [7] applied the technique in HyperTree to genomic, protein, and expression data, see Figure 2.18 C. Their application visualises phylogenetic and phenetic trees, or hierarchical clusters of gene expression in hyperbolic space.

Another example of an early visualisation technique from the focus+context set is **document lens** developed by Robertson and Mackinlay [93]. The technique is based on a common strategy for understanding paper documents when their structure is unknown. The document lens is used in Information Visualizer [92]. It is an experimental interaction technique addressing a set of goals, such as providing detail with its context or using 3D for more effective usage of space, when information is placed in a rectangular presentation.

Leung and Apperley [62] provide a taxonomy of distortion-oriented techniques that existed before

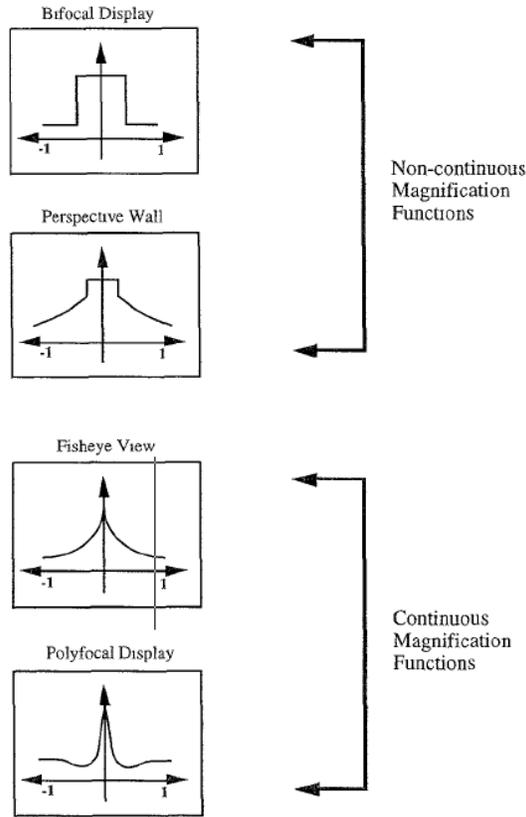


Figure 2.19: A taxonomy of distortion-oriented presentation techniques, from [62].

1994 which demonstrates clearly their underlying relationships, see Figure 2.19. The survey includes such techniques as bifocal display, polyfocal display, fisheye view, perspective wall, and graphical fisheye views. The authors stress that the main problem for the presented techniques is “the relatively small window through which an information space can be viewed”. This problem causes disadvantages in navigation, item interpretation, and correlation between items, where one of the items cannot be seen in its full context.

Leung and Apperley also discuss nondistortion-oriented techniques as techniques without suitable context to support navigation of large information spaces. On the other hand, they suggest that the nondistortion-oriented techniques could be adequate for small text-based tools where the users could display only part of a data set, and use scrolling or paging. Alternatively, the whole data set could be divided into small pieces with hierarchical access.

Leung and Apperley present three basic interaction methods (scrolling, pointing and selecting, and dragging) which effect a change of viewport. Since 1994 the technology has made progress and new input devices such as tactile, two-handed and immersive¹ are available now. However, the basic interaction

¹The users see where their hands are at all times, no matter whether they touch the surface or not. The solution allows them get to where they want to go quickly and efficiently.

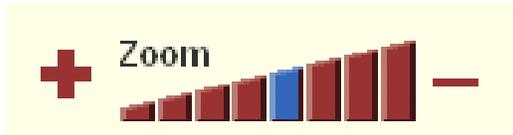


Figure 2.20: The zoom buttons offered by Ensembl (<http://www.ensembl.org>).

methods seem to be still the same, for example, for tactile input the users navigate by pointing and selecting. The only difference is that they do not use a keyboard or mouse, only their own fingers or special pencils.

The authors point out two problems associated with the presentation of data in a confined space: a spatial problem and an information density problem. Nowadays, users use large screens, but, on the other hand, they also collect more data. Therefore, we are still familiar with the same problems. Those are partially solved by distortion-oriented techniques, but if the user has enormous quantities of data, he still has too little room or the density of information seems to be too high.

The next Shneiderman task category is **zoom**, see Figure 2.14, which is widely used for all data types. The users can zoom either by pointing to a location and issuing a command [5] or only by issuing a zooming command (see Figure 2.20).

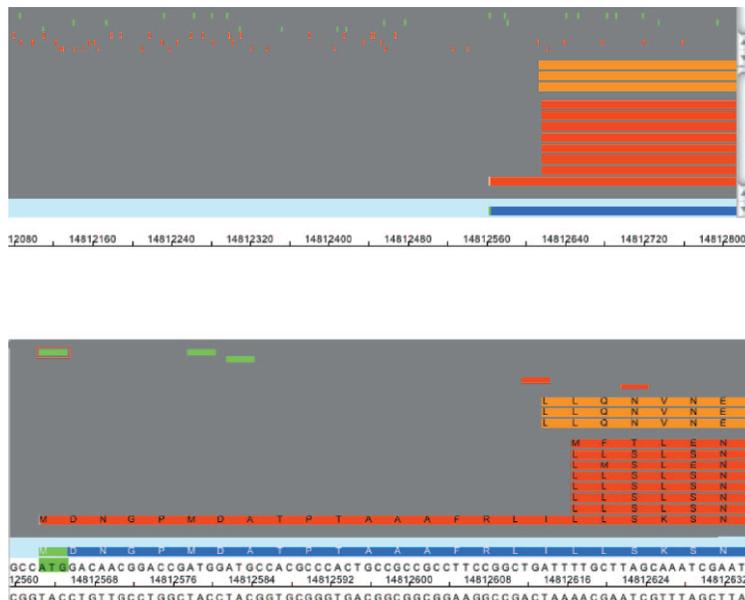


Figure 2.21: Semantic zooming which provides additional information about biological data. The figure is taken from [63]. The technique used in Apollo shows additional information for genes' similarity when zoomed in on, see the bottom panel.

Semantic zooming seeks a balance of overview and detail. A physical zoom, on the one hand,

changes the size and visible detail of objects. A semantic zoom, on the other hand, also changes the type and meaning of information displayed by each object. The Pad++ project [5] developed a prototype interface based on semantic zooming. The Pad++ is an alternative zooming graphical interface to a traditional window. It supports visual searching with zooming in addition to traditional mechanisms such as content-based search. This interface is also an alternative icon-based approach to interface design. It allows the users to create and manipulate 2D graphical data objects, such as coloured text, text files, hypertext, graphics and images, of any size, and navigate through the object space. Objects are represented differently, depending on their size, by defining procedural objects, that is, objects which are rendered as a result of one of a set of rendering procedures. Pad++ supports viewing at different scales and tries to direct one into natural spatial ways of thinking. It uses a three button mouse. The left button is mode dependent, the middle button zooms in, and the right button zooms out. Pad++ zooms around the current cursor position, which allows the users to control the zooming dynamically by moving the mouse during the activity. The Pad++ strategy is to visually explore the database. The system uses “parallel lazy loading”, which means that “only the portion of the database that is visible in the current view is loaded”.

Pad++ offers multiscale layouts of hypertext where the parent-child relationships between links are graphically represented. After selecting a hyperlink, the linked data is loaded and made smaller, then the view is animated to center on the new data. Pad++ can read in hypertext files written in HTML (Hyper Text Markup Language) and Mosaic [99]. It also provides a graphical interface for accessing the directory structure of a filesystem and its history.

Semantic zooming is widely used in biological applications where a huge volume of data should be visualised, see Figure 2.21.

Bederson and colleagues [4] offer toolkits for navigation called Piccolo and Jazz. Piccolo puts all zooming and panning functionality and about 140 public methods into one base object class, called PNode. Every node can have a visual characteristic, which makes the overall number of objects smaller than in other techniques which require two objects, an object and an additional object having a visual representation, as in Jazz [4]. A Jazz node has no visual appearance on the screen, and it needs a special object (visual component), which is attached to a certain node in a scene graph and which defines geometry and color attributes. Piccolo supports the same core feature set as Jazz (except for embedded Swing widgets), but it primarily uses compile-time inheritance to extend functionality and Jazz uses runtime composition instead. Piccolo supports hierarchies, transforms, layers, zooming, internal cameras, and region management which automatically redraws the portion of the screen that corresponds to objects that have changed. We used Piccolo toolkit during designing VisGenome (see Chapter 5), this allows us to implement zooming and panning navigation via mouse buttons.

Holmquist [43] developed **flip zooming** which allows him to visualise data sets as collections of linearly ordered visual elements. The technique lays out the elements in 2D space in a left-to-right and

top-to-bottom fashion, still reflecting their linear ordering. The user moves an element into focus by cycling on it, or by moving in to focus forwards or backwards to an adjacent element in the sequence. The chosen element zooms up to a readable size, while the other elements become smaller accordingly. The main intention for flip zooming technique was displaying documents, but it was also used to display other data. Flip zooming allows the users to overview the whole data set and focus on selected items. It supports random access to any visual element, space efficiency and preservation of linear ordering. It does not introduce spatial distortion of the individual elements. When the user zooms in on an element, the element may change its position to a completely different part of the display, the context elements may also move, and the number of rows and columns may change. Holmquist implemented flip zooming in the Zoom Browser (WWW text browsing), the Flip Zooming Image Browser (image browsing), and the Hierarchical Image Browser (for more complex image collections). Panning and scrollbars are used with zooming for focussing. They are very common and well known in biology. Almost every genome browser implements either panning or scrollbars.

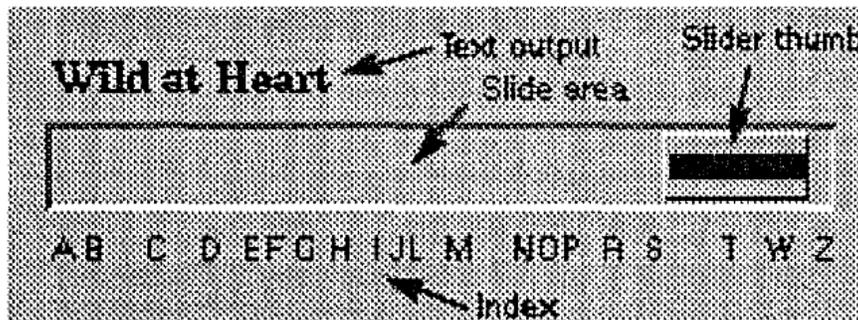


Figure 2.22: Alphaslider for selecting movie titles from [1]. Alphaslider allows the users to visualise and rapidly select text. It uses rapid serial visual presentation of text as a method for rapidly scanning and searching lists or menus in a graphical user interface.

Second main category of interaction techniques are **techniques for linking and filtering**. In Shneiderman’s classification we see here a **filter** task. As an example, we present the **Alphaslider**, developed by Ahlberg and Shneiderman [1], which is used for representing large 1D data sets, see Figure 2.22. Alphaslider can be used in direct manipulation of a database querying system to support dynamic queries. The main advantages of Alphaslider are its small size, one line of text, and the mapping of a huge number of items to a small number of pixels - “each movement of the slider thumb corresponds to a large number of items”. Ahlberg and Shneiderman conducted also an experiment with four different designs of Alphaslider. They concluded that an expert user needs only 50% of time (13 seconds) required by a casual Alphaslider user (24 seconds), while searching in a list of 10,000 film titles. Their work inspired our VisGenome work, where we developed a *scaling algorithm* and conducted a user study with biologists and medical researchers.

Also **value bars** [18] proposed by Richard Chimera are an example of a navigation and visualisation tool for data exploration. The value bars added to a text window allow the users to see additional information about a number of items.

Eick and colleagues [28] present a visualisation technique designed to help analyse line oriented data, and a software tool (Seesoft) supporting the technique. They use four ideas in their visualisation: reduced representation, colouring by statistics, direct manipulation, and option to read the actual code. Displaying files as columns and lines of code as thin rows generates a reduced representation. A statistic associated with a code line determines the colour of the row. The users can manipulate the display in Seesoft to find interesting items in the code and statistics by using direct manipulation and interaction graphics. Seesoft also allows the users to read the code. The users open up reading windows and position boxes over the reduced representation when they want to display the code text.

The idea presented in Seesoft could be also used in our collaborators' work. Some of the biologists from the BHF Glasgow Cardiovascular Research Centre work in the same project and use the same equipment in their medical experiments. Always, before they use the equipment, they have to check who used it before or if it was used not more than 5 times (probes) or even if the battery is still working. They put this information in their notebooks. However, if they noted what equipment was used in their shared files, this could save time.

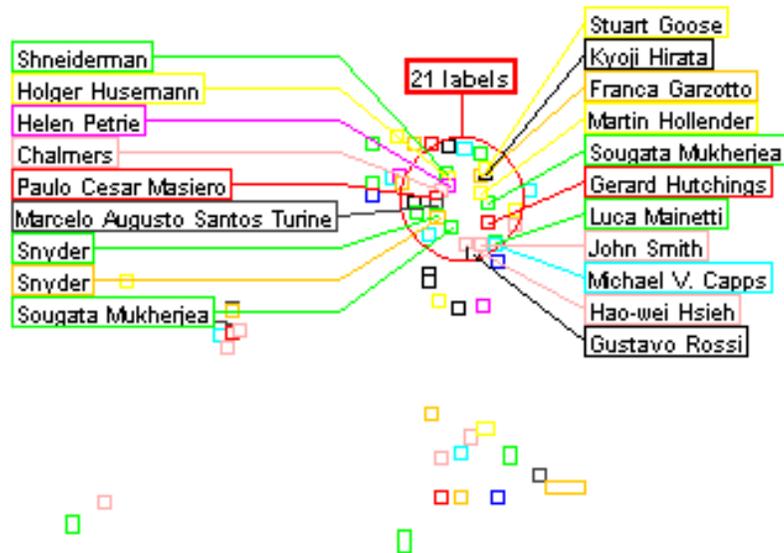


Figure 2.23: An example of excentric labelling, adapted from [30].

Fekete and Plaisant [30] describe **excentric labelling**, see Figure 2.23, which is a dynamic technique used to label a neighbourhood of objects situated around the cursor. They propose an informal taxonomy of labelling methods. Information visualisation systems often lack adequate labelling strategies,

Relate is the next task in Shneiderman’s classification. It shows relationships between items. As an example we discuss **table lens**. Rao and Card [91] use the table lens for “visualising and making sense of large tables”. The table lens supports effective interaction with large tables by uniting symbolic and graphical representations into a single coherent view. The fisheye technique used in the table lens, together with the combining of graphical representations with table visualisation and manipulation, are another contribution of the table lens. The technique allows for distortion in each of two dimensions, independently. This means that rows and columns can be scanned totally by a single horizontal or vertical eye motion. Rao and Card define four types of cell regions in the technique: focal (see Figure 2.24 - G4, H4, G5, H5, G6, and H6), column focal (columns G and H), row focal (rows 4, 5, and 6), and nonfocal (cells: A1-F3, I1-N3, A7-F10, and I7-N10).

Zoom, adjust and slide are the basic manipulation operations used for controlling the focal area. Rao and Card offer a small number of keyboard and pointer gestures. They mainly use two mouse buttons for interacting with the table lens - for “touching” and for “grasping”. The table lens offers text, colour, shading, length, and position to representing underlying cell values.

A number of biologists keep their experimental data as Excel files. The table lens could represent their data more clearly and allow them to interact with it more easily.

The two last tasks presented by Shneiderman are **history** and **extract**. They are very important and useful for all scientific fields, including biology, however we do not focus on them in the thesis. We only mention that we plan to represent history in VisGenome. We would like to mark the chromosome regions depending on who viewed them, and when and how often they were viewed. The feedback received from the mixed paradigm user study conducted with the biologists (described in Chapter 8 - ‘Mixed Paradigm User Study’) stresses that our application needs also a user-friendly technique for extracting data.

Edit Wear and Read Wear [42] is a prime example of working with history. Hill and colleagues describe two applications which visualise the idea of “computational wear” in document processing. They suggest an analogy to physical wear. Edit Wear means to graphically portray the document’s authorship history and Read Wear suggests a representation of the document’s readership history. To visualise Edit Wear and Read Wear, the authors used a technique called attribute-mapped scroll bars. This allows for mapping users’ marks onto document scroll bars in a position corresponding to line positions. Edit Wear and Read Wear not only modify document processing, but also move part of reading and editing from being private to semi-public.

Hill and colleagues also present the idea of Menu Wear which gathers a statistics of previous menu-selections by category of user and context. According to the authors Menu Wear should be incorporated into the menu items.

An example which could help in our future work with extracting data in VisGenome is Visage [95]. Visage allows users to manipulate data and supports visual display during data exploration and analysis.

Objects representing information among visualisations and application interfaces throughout the Visage environment can be dragged by a user. One simply drags copies of a data set and composes it into new data aggregates. VisGenome could extract data to Excel format, which would be helpful for biologists, especially since they use Excel very frequently during their work (see Chapter 8). Biological data could be also simply dragged between different genome browsers. In the next chapter of the thesis we see that some genome browsers support data extraction.

A number of extraction tasks could be carried out on different data types. However, there is also a subset of tasks, such as searching or filtering, that need to be supported for all kinds of data. Therefore, it is important that proper visualisation techniques for each data category are used, that allow the users to navigate and interact with their data.

2.3 Summary

The visualisation world is rich and contains a lot of techniques that could be used in genomics and biology. We introduce some of the graphical data representations and interaction techniques which are used in biology or could be potentially employed in our work. We could implement such interaction techniques for genomics data received from biological experiments. We might use them in graphical tools, like, for example, fisheye or excentric labelling in VisGenome. On the other hand, some of the techniques, such as the table lens or flip zooming, could be implemented with numerical data collected from biological experiments. The majority of the visualisation techniques surveyed here are not common in biology or are not used by medical researchers. In my opinion, focus+context techniques in particular could be very useful in representing detailed genome data while still showing context on a chromosome, where a gene is situated. Here, we present only a small subset of the known visualisation techniques, which, we hope, shows a good deal of variety. As described in later chapters, some of the techniques we mentioned are used in our research or could be used in the future work with VisGenome.

The majority of the visualisation techniques presented in this chapter, such as Tree-Maps, fisheye, semantic zooming or hyperbolic browser are used in the field of biology. Tree-Maps and hyperbolic browser are applied to hierarchical data to show a lot of data on a small screen. They are useful for biological data, however, when they are especially used for hierarchical data. Zooming and semantic zooming are common in genomics and allow the biologists to easily navigate their data. As we present in this, and the next chapter, fisheye is uncommon in genomics, however, a few biological applications have applied the technique. On the other hand, there are techniques, such as cone tree, which are still not used in biology, but their potential could be used to visualise genomics data. We think that each of the presented visualisation techniques has valuable features (showing detail and context in fisheye or semantic zooming, representing a number of data in Tree-Maps or cone tree) which can be used during VisGenome evaluation.

Properly used visualisation techniques could save the users time, and make their work more effective. Often the researchers that we studied have a lot of data and only a small screen for data representation. Therefore, in this chapter we stress that key to proper visualisation is not only data representation but also interaction techniques which provide easy access to the data and support view manipulation.

2.4 Conclusion

We described some of the existing techniques, terminology and problems connected with visualisation. We divided visualisation techniques according to Shneiderman's classification into seven categories of data types and seven tasks. We also briefly described each from the data types and tasks, and gave examples of their use in biological tools.

Some of the presented visualisation techniques we used during VisGenome development. On the other hand, some could be used in future work. We want to show the users detailed data and overview, which allow them not to become lost themselves. Therefore we used zooming and panning in VisGenome. The techniques allow biologists to quickly find their data and easily navigate it. The data presented in VisGenome is not hierarchical, but we can take the hierarchical subset of the data, as a chromosome and its relationship to other chromosomes from different species. For this subset we could apply Tree-Maps, cone trees, or a hierarchical browser, which are perfect for representing high volumes of data. We studied visualisation techniques and surveyed existing genome browsers, to find solution for presenting biological data. In this chapter we recognised that some of the visualisation techniques properly implemented could be very useful in biologists work. On the other hand, as can be seen in next chapters, we recognised that none from existing genome browsers fulfill biologists requirements. Therefore, we decided to design VisGenome, which implements some of the visualisation techniques (zooming, panning) and fulfill requirements for genome browsers presented in next chapter.

During our experiments we paid attention to how visualisation techniques applied to biological applications could help in medical researchers' work. Therefore, the following chapter describes the most popular genome browsers and the visualisation techniques employed by those tools.

Chapter 3

Genome Browsers

This chapter presents a survey of genome browsers and focuses on visualisation techniques, especially interaction techniques, presented in the biological tools. We introduce some biological background, then we present our genome browser classification system, outlined in [48] and survey genome browsers while focusing on the visualisation techniques they use. In the next chapter we present our work undertaken with existing genome browsers (DerBrowser and SyntenyVista) in a first stage of the PhD research. The work presented in this chapter was the motivation to create a new genome browser - VisGenome - which is presented in Chapter 5 of the thesis.

3.1 Introduction

Biomedical research is a large area of science that aims to prevent and treat diseases that cause illness and death in people and in animals. This field of research aims to improve our lives through treatment or prevention of diseases, better diagnosis, new medications, new crops, and better understanding of the environmental impact our technologies have. Scientists use animal models of disease to learn more about health problems, and to assure the safety of new medical treatments. In this context genome browsers allow one to visualise not only human data but also data from other species.

Genomics is the study of genomes and of the relationship between genomes and the way an organism functions. Each living organism has a genome which encodes information passed down from generation to generation. A bacterial genome consists of several million DNA (deoxyribonucleic acid) molecules (e.g. the tuberculosis genome is about 5 million long). A human or mouse has a genome of around 3 billion letters of DNA code. A genome is encoded in DNA or RNA (ribonucleic acid) molecules of four types (A, C, G and T for DNA). It encodes all proteins and signalling molecules needed by an organism. Only 1.5% of the human or mouse genome is translated into proteins, which are the building blocks of our bodies.

Chemically, proteins are strings of amino acids (AA), where three letters of DNA correspond to one amino acid. Proteins are described via an alphabet of 21 letters, and in our bodies they fold into 3-D structures which may change conformation as they perform their various functions. We do not know exactly how many genes humans have, with the current estimate being between 20 and 30 thousand. They give rise to probably around 1 million proteins. The process of translation from DNA to protein is complex, and it is important to remember that a stretch of DNA of some 30 thousand letters gives rise to a protein of some 300 letters. The parts of DNA which translate into protein are called *exons* while the parts which control the process are called *introns* or untranslated regions. Biologists want to know for each protein what gene produced it, which parts of the gene were used in this particular protein and which control regions were activated during the production process. The process of protein production is dependent on the type of cell, developmental stage, the environment, and many other factors which collectively influence the health of an organism.

Genomes of a very large number of animals are known relatively well, and are publicly available (e.g. <http://www.ensembl.org>), along with genome maps which show how genes are arranged and structured. Mammalian genomes are split into around 20 chromosomes, and the set of chromosomes forms a *karyotype*, while bacterial genomes form a circle. Groups of genes that are shared between related organisms are often collocated in the so-called *synteny groups*, and biologists study such gene groups, as there is proof of synchronised activity over groups of genes, and of similar gene functions being shared between related organisms. Similar gene functions arise from similar DNA and protein sequences, and the biologists *align* genomes and genes to understand what sequences are shared, and what functions are common to a group of organisms.

Genes and the resulting proteins interact and their interactions can be conceptualised as *pathways*. Such pathways stand for chemical and structural reactions which orchestrate all the processes which keep us alive. Pathways may be shared by groups of organisms but there are known cases where they diverge. Pathway visualisation tools include [10].

Very large numbers of genes have no known function, and genes responsible for many common diseases like hypertension are not known. It is assumed that such diseases are controlled by a number of genes, and are under strong influence of the environment (diet, smoking, exercise, etc.). The search for disease genes uses the techniques of gene mapping, where populations of subjects are tested and a statistical correlation between a part of a chromosome, containing a number of genes, and the disease is expressed as a quantitative trait locus (QTL) [45]. The study of QTLs leads to the identification of candidate genes which then may be proven to be the causes of diseases. The study of QTLs is easier in animals (e.g. rat, mouse) bred to be genetically identical, while human genomes have a variant DNA letter every few thousand letters for any two humans, and therefore statistical correlations are harder to make. Diseases are often studied in animals, and then the candidate human gene will be sequenced

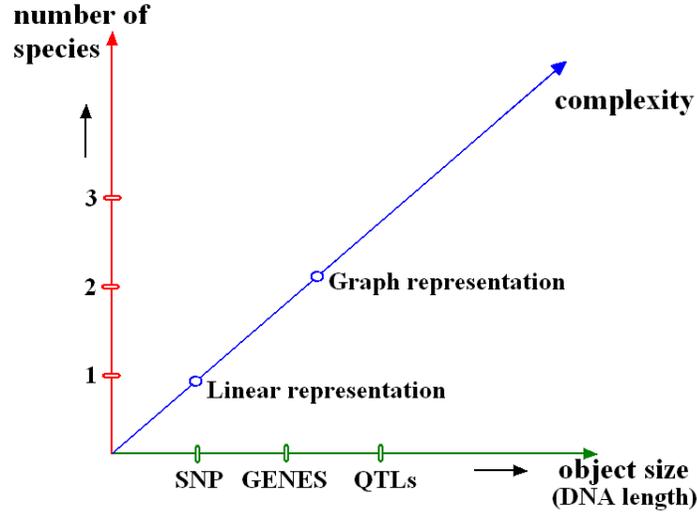


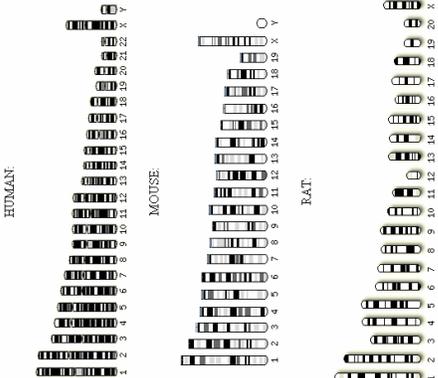
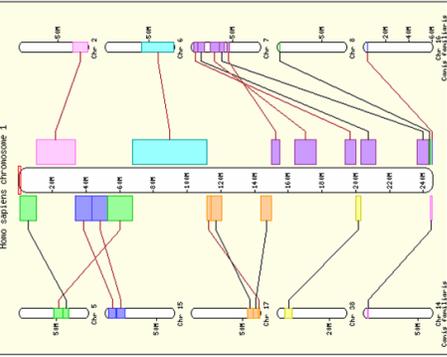
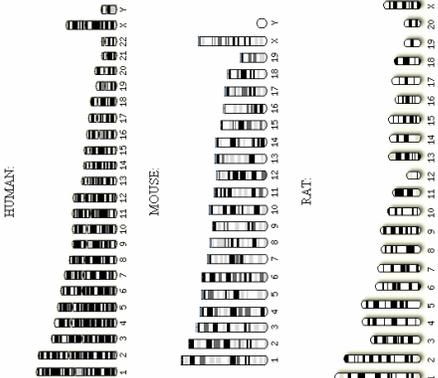
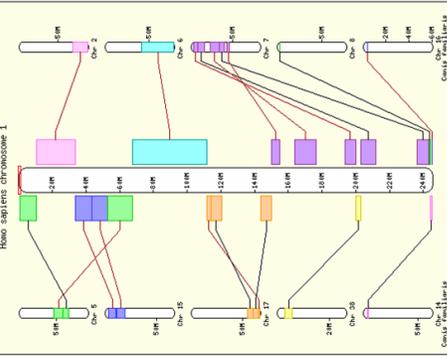
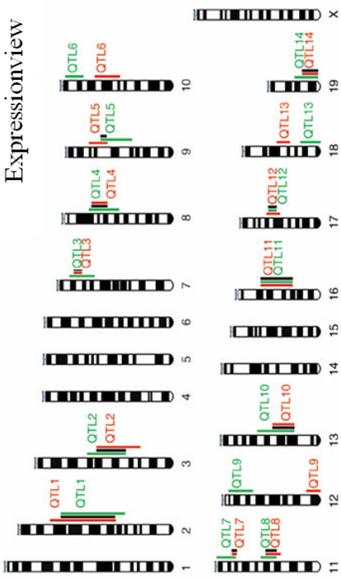
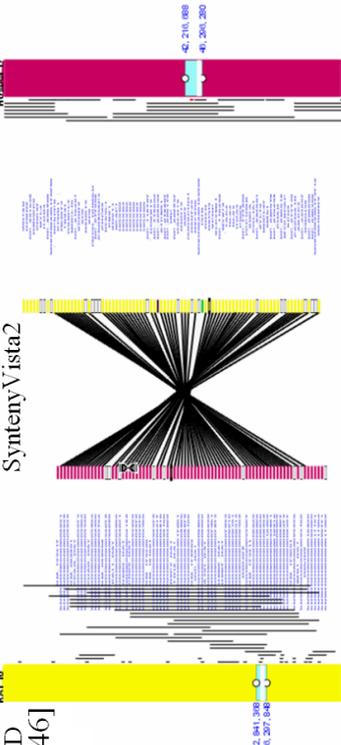
Figure 3.1: Genome browser classification schema.

(from blood samples gathered from patients), and subjected to further analysis which may uncover the biochemical causes of disease.

Biologists are faced with very large data sets. A QTL may contain around a hundred genes, or a few million letters of DNA code. On the other hand, single nucleotide polymorphisms (SNPs), which are individual DNA differences, are one letter long, and need to be shown along QTLs. Visualisation is the only viable way of making this data available, as close reading of thousands of letters is not a solution. That is why genetic databases visualise data in the form of maps which show a linear arrangement of genetic features. To our knowledge, the resulting visualisations have not been subjected to much scientific scrutiny, so far. They are used by thousands of scientists daily, but it is not clear how they should be best designed and how well they support scientific activity.

3.2 A Classification System

As shown in this and the next section, ‘Survey of Genome Browsers’, the world of genome browsers is very rich. We studied existing genome browsers and classified them according to three dimensions, see Figure 3.1. In the first dimension (number of species) we find that genome browsers represent between one and many species. Ensembl can be used to view one species at a time but other species’ information can be superimposed (see Figure 3.6 B). K-BROWSER [13] can show a number of species but the number is limited by the size of the web page (see Figure 3.17). Multiple alignment tools can show a number of aligned sequences from different species and those sequences can also be shown as a tree, see Figure 3.2 I

SPECIES	SINGLE REPRESENTATION	COMPARATIVE REPRESENTATION
<p data-bbox="500 1711 537 1854">SPECIES</p> 	 <p data-bbox="641 1596 714 1648">A [44]</p>	 <p data-bbox="641 892 706 945">B [44]</p>
<p data-bbox="844 1669 885 1900">KARYOTYPES</p>	 <p data-bbox="1047 976 1088 1060">Ensembl</p>	 <p data-bbox="1047 199 1079 294">Ensembl</p>
<p data-bbox="1266 1743 1307 1827">QTLs</p>	 <p data-bbox="1112 1596 1185 1648">C [31]</p>	 <p data-bbox="1112 892 1177 945">D [46]</p>

(alignment¹) and Figure 3.2 J (phylogeny²).

The second dimension represents the size of the objects shown. The smallest objects are one DNA letter long (SNPs). In the order of increasing size one can show gene promoters, exons and introns, and other constituent parts of genes. Genes are about 20-30 thousands of DNA letters long and QTLs may contain thousands of genes. QTLs may approximate chromosome bands in size. Finally, human chromosomes are between 50 and 300 million letters of DNA long.

From the point of view of representation complexity, we classify browsers into linear and graph representations. In graph representations we can distinguish trees, networks (e.g. pathways), and 3D structures (e.g. proteins). JMol [139], see Figure 3.2 H, is one of the viewers showing protein 3D structure, while Treeview, Figure 3.2 J, offers a tree representation of a phylogeny. BugView (Figure 3.2 F), Ensembl (Figure 3.2 B) and SyntenyVista (Figure 3.2 D) show genome comparisons as bipartite graphs.

We classified existing genome browsers according the three requirements (number of species, object size, and complexity), however, we found also other requirements which are very important from the point of view of our biological collaborators. We designed a list of requirements for a genome browser, which could be useful for biologists' work, as follows:

- We want a genome browser to present a number of different kinds of data, especially genes, markers, micro array probes, and QTLs which are very important during biologists' work, see Table 3.1 - only Ensembl and SyntenyVista present QTLs, but only for the rat chromosomes.
- We do not want to limit a genome browser to any species, however, from our users' point of view, it is important that it has data for the rat, the mouse, and the human chromosomes.
- Our collaborators conduct a number of experiments and they want to show their own data in a genome browser, which is also suitable for the future genome browser, see Table 3.1.
- The biologists want to compare their results with the existing data, see Table 3.1.
- Easy navigation tools are very important because they allow the biologists to do their work more effectively, see Table 3.1 - last column.
- Another important requirement is that a genome browser is commercially available, and for free. However, all studied genome browsers are free, therefore, we do not put the requirement in Table 3.1.

¹Alignment is the adjustment of an object in relation with other objects, or a static orientation of some object or set of objects in relation to others.

²The phylogeny of organisms is the history of organismal lineages as they change through time.

Genome browsers	OBJECT	QTL	SPECIES	INPUT	CR	NAV
AceDB	F	no	L	no	no	no
SGD	F	no	L	no	no	no
Ensembl	F	yes (partially)	RMH	no	yes	no
UCSC GenomeBrowser	F	no	RMH	no	no	no
SyntenyVista	L	yes (partially)	RMH	no	yes	yes
DerBrowser	F	no	RMH	yes	no	no
Apollo	F	no	RMH	no	no	no
Artemis	L	no	RMH	no	no	no
BugView	L	no	RMH	yes	yes	no
Sockeye	F	no	RMH	no	no	no
K-BROWSER	F	no	RMH	no	no	no
GBrowse	F	no	RMH	no	no	no
NCBI	F	no	RMH	no	yes	no
eQTL Explorer	L	yes	RMH	no	no	yes
Triple synteny Human-Mouse-Rat	L	no	RMH	no	no	no
VCMAP	F	no	RMH	no	yes	no
Cinteny	L	no	RMH	no	no	no
SynView	F	no	RMH	no	yes	no
BioViews	F	no	L	no	no	no
SyMAP	L	no	RMH	no	yes	no

Table 3.1: List of requirements for genome browsers. The abbreviations are: OBJECT - the number of different kinds of data or limited subset of data (L - limited, F - a number of different kinds of data), QTL - QTLs data is available (this data was very important for our collaborators), SPECIES - for our collaborators the most important are the rat, the mouse, and the human chromosomes (RMH), all other genome browsers from their point of view offer limited subset of animals (L), INPUT - to provide an **easy** way to input biological data, CR - to provide a view for comparative representation, which allows the users to see **details**, NAV - to provide smooth visualisation techniques such as smooth zooming and panning.

3.3 Survey of Genome Browsers

In the mid-1980s scientists believed that a better understanding of DNA would solve the secrets of life and bring a revolution in medical science. This motivated the start of the Human Genome Project (October

1990) [20] which aims to decipher the sequence of the human genome, however other species were also being sequenced. The main reason for the large-scale sequencing of the mouse and rat genomes were similarities in gene structure and function across species. The scientists believe that if a gene has a similar function in two species, it is similar in structure and sequence in the animals. The amount of data produced by genome mapping and sequencing motivated the creation of GenBank [131], the DNA sequence database, and accelerated the development of molecular biology and bioinformatics. Bioinformatics face the problem of creating tools which could be used for showing the whole genome, small details such as genes or SNPs, and also comparative genomics. The main problem in a genome browser (see Figure 3.3) is how to show large and small regions together, i.e. how to show an overview but also the details for each of the data categories.

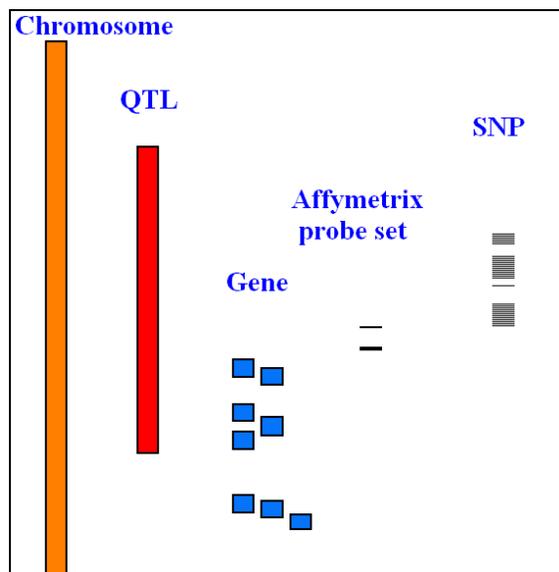


Figure 3.3: A genome browser represents chromosomes which are strings of letters: A, C, G and T, and contain between a few million and several hundred million letters (base pairs: bp). A QTL measures up to several million bp and an Affymetrix probe has 25 bp, while a single nucleotide polymorphism (SNP) is 1 bp long. Large differences in the size of the presented objects cause difficulties in the representation of data.

We present a survey of existing “classic” genome browsers used by our collaborators, see Table 3.2. We also introduce metabolic pathways and some other data analysis tools, which are used by medical researchers and have no complicated graphical interface, but also require user-friendly interaction techniques. A lot of them use basic interaction techniques such as zooming or panning. None of the presented genome browsers were tested in a formal user study. We used Ensembl in comparison with VisGenome in our quantitative user study, see Chapter 6.

Genome browsers	Basic Techniques	User Study
AceDB	zoom buttons, popup menus, double clicking, colours	no
SGD	zoom buttons, colours, clicking	no
Ensembl	zoom buttons, scrolling, popup menus, colours, clicking, overview	VG and Ens see Chapter 6
UCSC GenomeBrowser	zoom buttons, popup menus, clicking, overview	no
SyntenyVista	smooth zooming, panning, colours, overview	no
DerBrowser	zooming (a slider), scrolling, colours	no
Apollo	zoom buttons, scrolling, colours, overview	no
Artemis	zooming (a slider), scrolling, colours	no
BugView	zooming (a slider), scrolling, search, colours	no
Sockeye	zooming, semantic zooming, panning, rotating, colours	no
K-BROWSER	zoom buttons, clicking, colours	no
GBrowse	zooming, semantic zooming, scrolling, clicking, overview, colours	no
NCBI	zoom buttons, popup menus, scrolling	no
eQTL Explorer	zooming, popup menus, scrolling, colours	no
Triple synteny Human-Mouse-Rat	clicking, colours	no
VCMaP	zooming (a slider), clicking, colours	no
Cinteny	zooming by clicking, colours	no
SynView	zooming, popup menus, overview, colours	no
BioViews	semantic zooming, scrolling, colours	no
SyMAP	zoom buttons, scrolling	no

Table 3.2: User studies and basic techniques such as zooming and panning offered by classic genome browsers.

3.3.1 Classic Genome Browsers

A number of different genome browsers were created, but probably the earliest one was released in June 1991, with the development of **ACeDB** (*A Caenorhabditis elegans Database*) [26], see Figure 3.4. AceDB offers an object-oriented view of biological data, the ability to store very large objects, and rapid response time. The tool also provides a graphic representation which contains many objects in various colours. Colours help the researcher to identify the objects. For example, when a marker is coloured in yellow, it means that this marker has been cloned. The graphical user interface allows the users to view a genetic

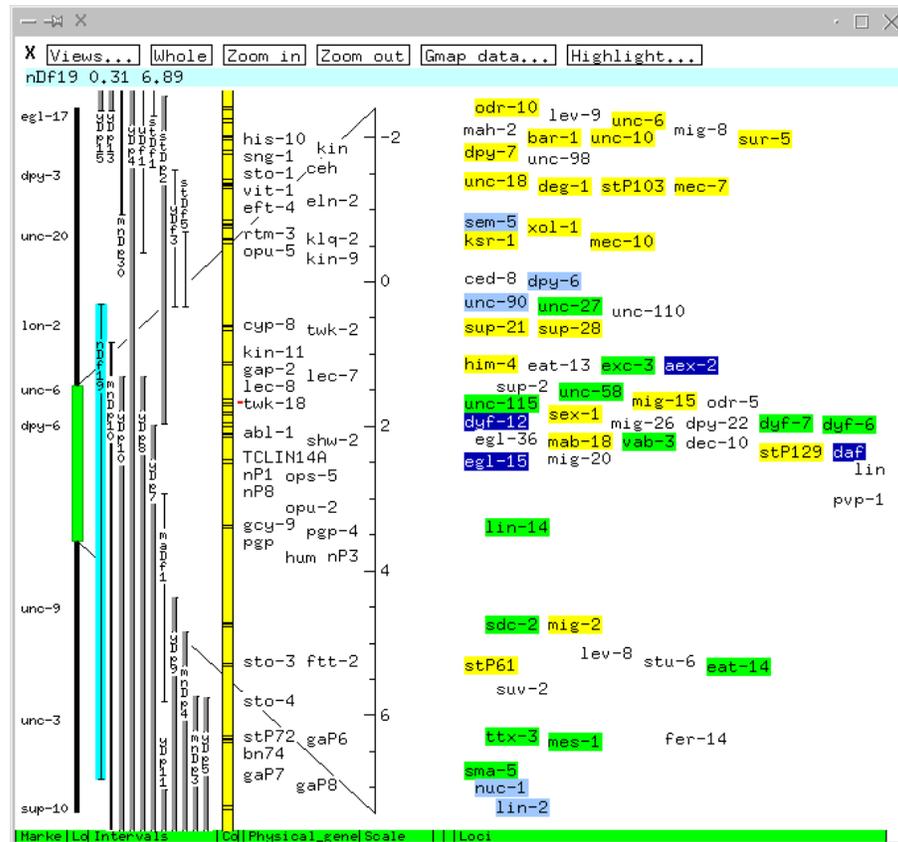


Figure 3.4: AceDB-representation of worm chromosome X. Figure adapted from [121].

map and display sequence annotation. The users can view textual details by double clicking on an object. AceDB offers simple zooming activated via zoom buttons. The viewer offers three types of sequence views: a genetic map, a physical map, and a sequence window which shows the DNA or AA (amino acid) sequences and is used for analysing and annotating them. All map views offer pop-up menus and present a series of columns which may differ in usage and position between databases. ACeDB is still being developed and potential users can get it from the web page <http://www.acedb.org/>.

To make genome browsers more accessible for users, a number of web accessible browsers have been developed. One of them was **SGD** (Saccharomyces Genome Database) [17]. SGD, see Figure 3.5, was designed to provide quick access to knowledge available for the budding yeast *Saccharomyces cerevisiae*. SGD focuses on the genome sequence and genes. It offers graphical interfaces which are geared towards biologists using the database. The application uses colours to code different types of genetic information. It also offers very limited zooming via buttons. The users see the description ">>Zoom In<<" when zooming is available, and after clicking on the description the picture becomes about 4 times larger.

Since the development of AceDB and SGD a number of genome browsers have been implemented. It is not possible to describe them all in detail. Therefore, we focus here on the genome browsers which

Combined Physical and Genetic Map

Chromosome IX : 260000 to 360000 bp, -37 to 7 cM

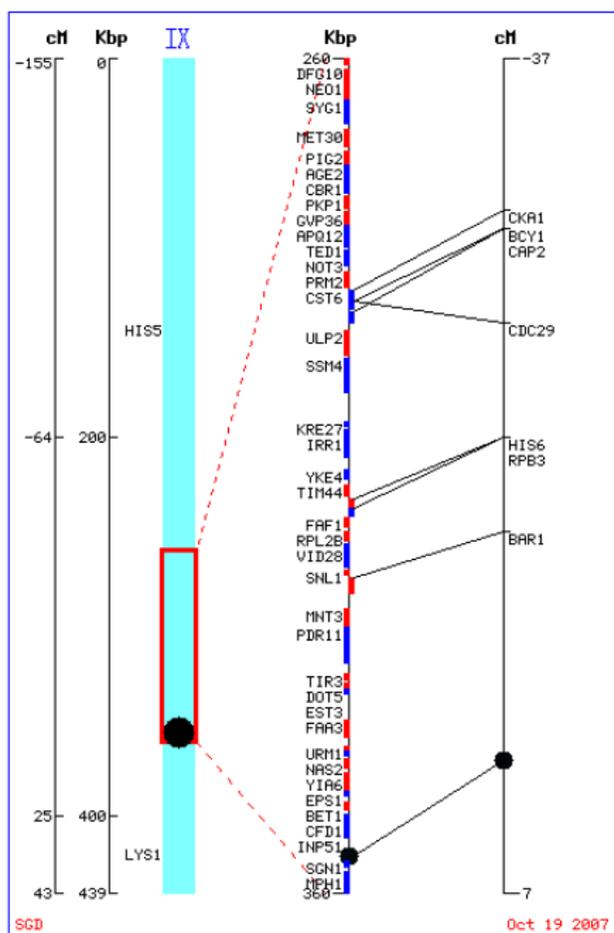


Figure 3.5: A clickable physical and genetic map in SGD. Data presented for chromosome IX of the budding yeast.

most influenced our research.

Ensembl [44], see Figure 3.6, is probably one of the most popular systems for genome analysis. The Ensembl database organises biological information around the sequences of large genomes. It is an interactive Web site, a set of downloadable flat files, and a complete, portable open source software system for handling genomes. The Ensembl browser displays assembled sequences, cross-species synteny, genes, transcripts, proteins, supporting evidence, dot-plots, protein domains and gene/protein families. The users can find 17 different views for data offered by Ensembl: AlignView, AnchorView, ChromoView, ContigView, CytoView, DomainView, ExonView, FastaView, GeneView, KaryoView, MapView, MarkerView, MultiContigView, ProteinView, SNPView, SyntenyView, and TransView. Different views are

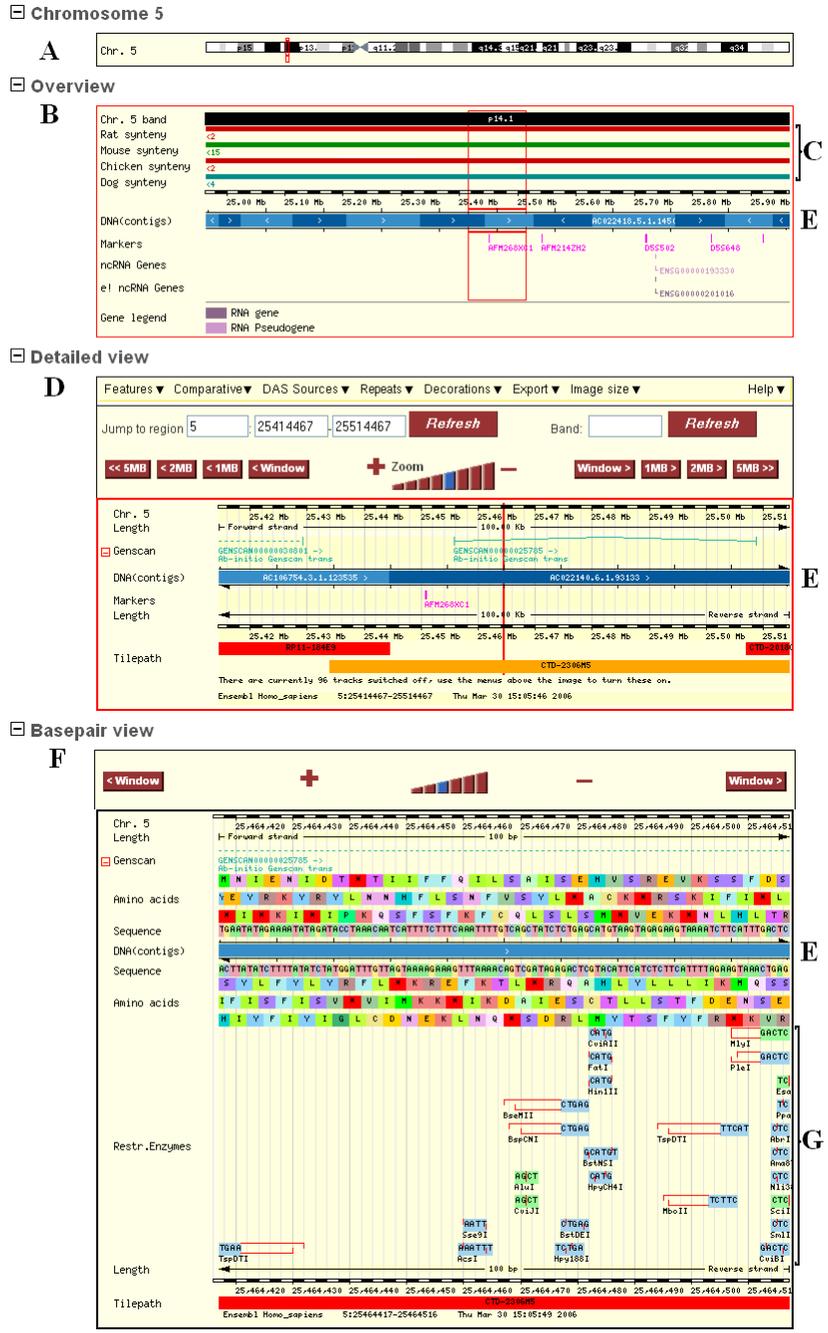


Figure 3.6: Ensembl - ContigView - human chromosome 5. ContigView provides a high level view of the contig sequences (E) that form the genome sequence assembly, and of genes and other features that have been placed on it. The figure shows the entire chromosome (human chromosome 5, see A), an ‘Overview’ (B) panel displaying a chromosome region of up to 1 Mb, the ‘Detailed View’ (D) panel showing genes and markers, and a ‘Basepair View’ (F) panel showing within a small assembly region of up to 500 bases the actual sequence, translations and restriction enzyme recognition sites (G). C shows syntenic chromosome fragments in other species.

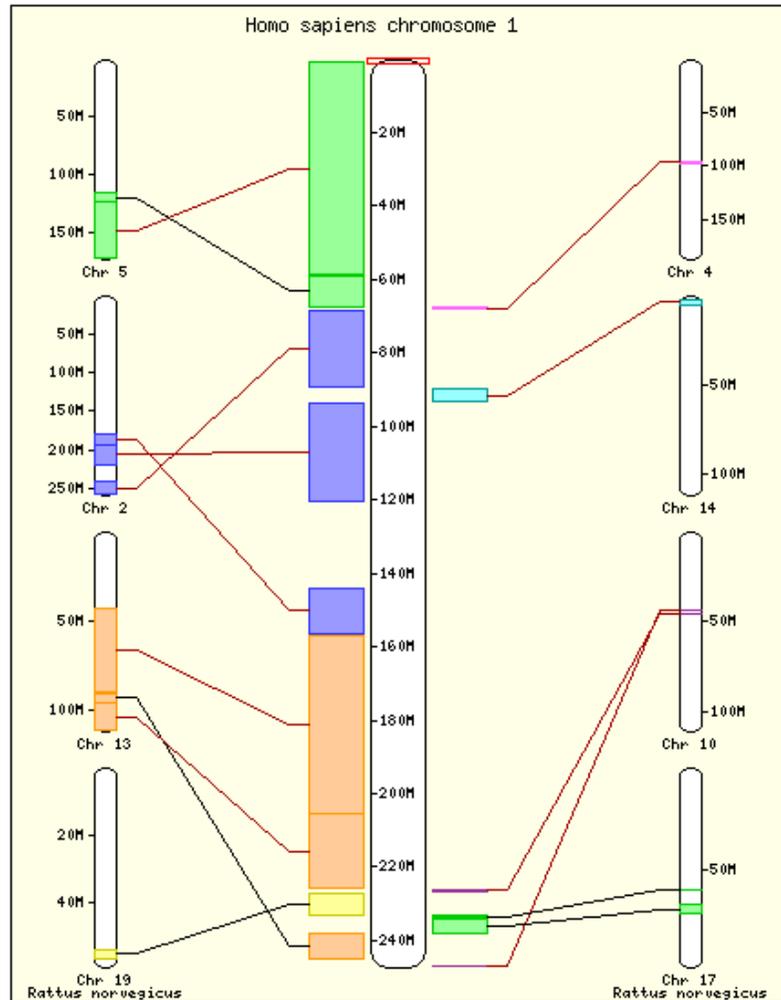


Figure 3.7: Ensembl - SyntenyView - human chromosome 1 and rat chromosomes. SyntenyView provides clickable pictures which allow one to see more information about synteny between chromosomes.

used to represent different kinds of data. In our first experiment (see Chapter 6 - ‘Initial Quantitative User Study’), a number of genomic data were represented by ContigView, MultiContigView and SyntenyView. In SyntenyView, see Figure 3.7, a diagram of chromosomes with blocks of conserved synteny and homology matches between individual genes with syntenic blocks shown. The SyntenyView is not satisfactory; it does not allow zooming and panning visible features. In ContigView, see Figure 3.6, a set of different views of a gene is shown, from broad chromosome context to fine nucleotide detail. These views are in separate horizontal frames, one below the other. The data presented in Ensembl is supported by labelling and searching. MultiContigView is an extension of ContigView. It supports the display of genome annotation for several species. We find that because of the size of the data set and the layout, it is difficult to show all requested details in one screen. The users need to use scrolling and very often get lost in the information space.

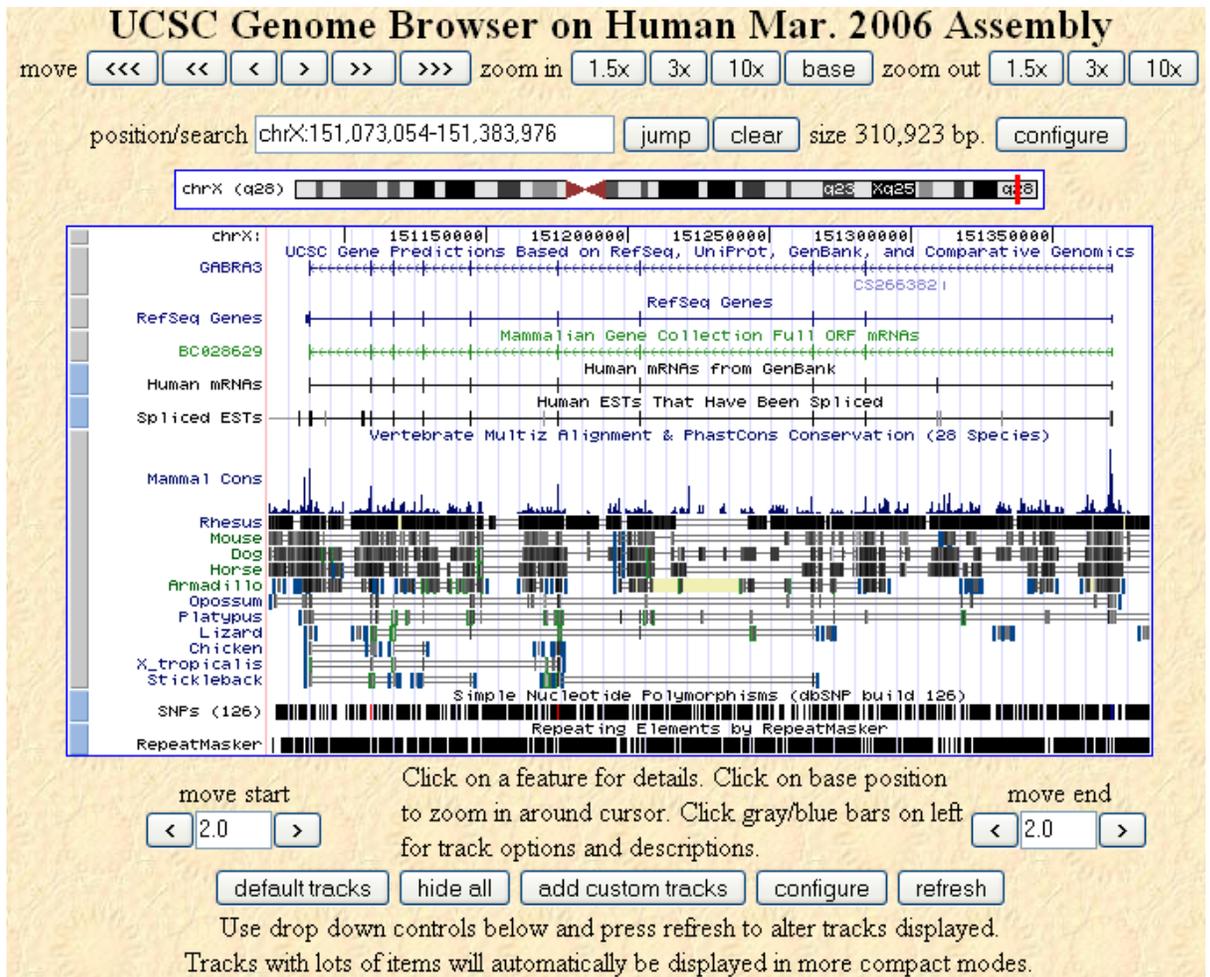


Figure 3.8: The University of California Santa Cruz (UCSC) Genome Browser visualises the human chromosome X.

The University of California Santa Cruz (UCSC) Genome Browser [56] is similar to the Ensembl tool, see Figure 3.8. It collects available genomic data from biological sources and stores it in a MySQL database. The application offers graphical and text-based views. The users can, similar to Ensembl users', zoom via buttons and specify a type of data they would like to see. After clicking on an element, the users see additional data, however it is strictly textual data, while Ensembl offers as well a graphical interface for selected elements. Ensembl and UCSC Genome Browser offer overviews, however, in Ensembl the users can choose any region to focus on by marking it by a red square. In UCSC Genome Browser the users can click on a band and detailed information is presented for the whole band. Both tools offer popup menus, however the UCSC Genome Browser popup menus are better designed as they disappear after use. Ensembl offers a number of different views, while the UCSC Genome Browser offers a variety of data and textual information, but only one type of graphical interface for single chromosomes.

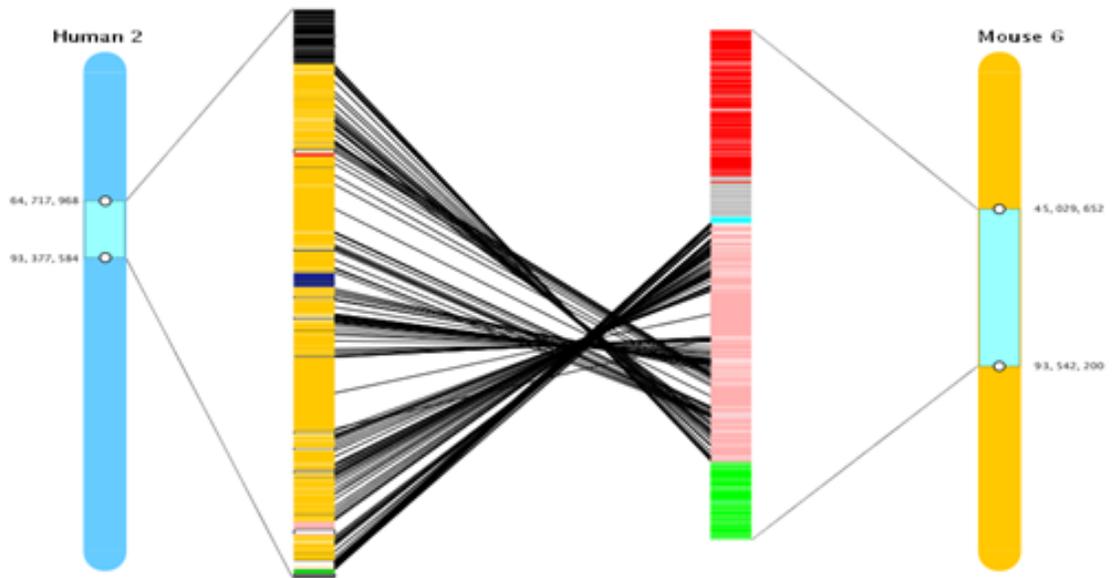


Figure 3.9: SyntenyVista - SV1. The display contains an overview of the human chromosome 2 on the left and of the mouse chromosome 6 on the right. In the centre a comparison of the selected parts of both chromosomes is shown. Black lines connect genes which are considered to be counterparts in the two species. A colouring scheme supports the user in relating genes to individual chromosomes.

SyntenyVista [46], see Figure 3.9, was the first interactive representation of synteny data designed for large genomes. It shows information about the human, rat and mouse genomes, and allows us to see the relationships between genes and chromosomes in two species at a time. SyntenyVista focuses on the visualisation of gene comparisons. The tool shows relationships between genes, syntenic groups, chromosomes and QTLs. It has features which make it more usable than other existing genome browsers. SyntenyVista shows the whole chromosome with detail and supports choosing the part which will be investigated. The view uses colour and chromosome numbering to support understanding at the starting point of the visualisation. The users can manipulate the view by using both mouse and keyboard interaction. An early version of the application (SV1) offers the option to invert the chromosomes, which was found to be useful. It offers smooth zooming which supports the visual exploration of the chromosome space. The user can keep an area of interest in focus during the zooming process. The tool also supports panning. The users can move the chromosome with the mouse on the gene panel, or drag a box enclosing the region of interest. The display of the genes can be scaled by using a mouse action. The second version of SyntenyVista (SV2), see Figure 3.10, has the cartoon scaling feature. The functionality visualises all genes as being equidistant and the same size.

On the other hand, SyntenyVista is poorly designed and the software has no documentation or comments. The code contains a lot of items (classes, methods) which probably were planned by the designer but never implemented. Because of this the users can see alerts or windows which appear and offer some

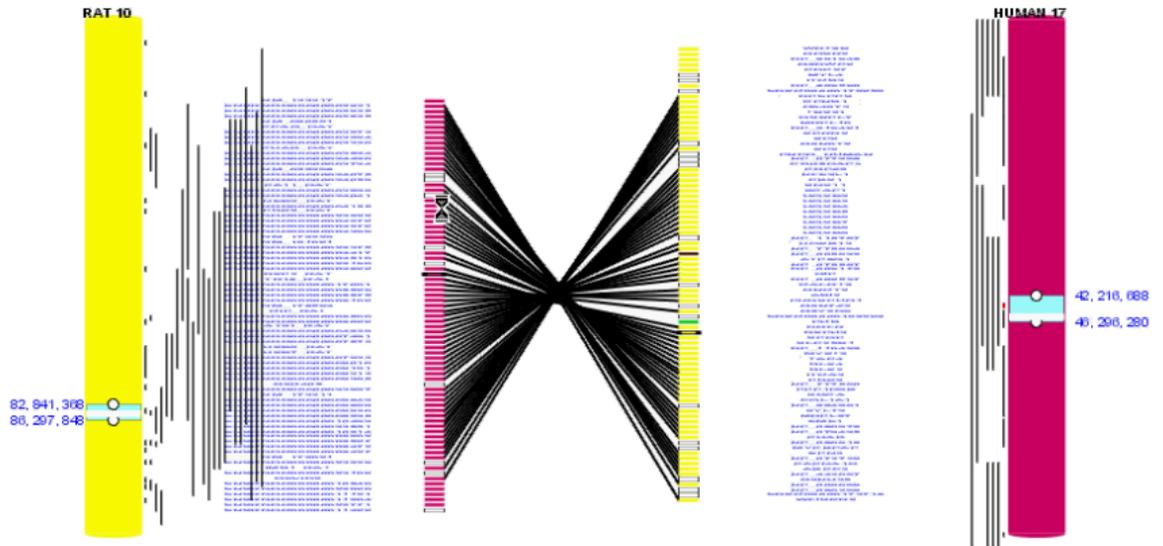


Figure 3.10: SyntenyVista - SV2. The view shows an overview of the rat chromosome 10 on the left and the human chromosome 17 on the right. QTLs (chromosomal regions associated with a disease) are shown as vertical black lines situated near the chromosomes.

functions but they can choose only one “path” through the application, and are distracted by pop-ups. For example, when we select a chromosome within the space of genomes, we are asked if this display should be offered in a 2D or a 3D version, but only the 2D version has been implemented. SyntenyVista has also a poorly designed database connectivity interface. At initialization the user can download data from Ensembl and he is presented with menu options to read the data from a file or to issue an SQL (structured query language) command. In fact the users do not have a choice; they can only download the data from Ensembl because the functionality which allows reading from a file does not work with the new version of SyntenyVista. On the other hand, the SQL command line is completely useless for people who do not know SQL, which is generally the case in the user population (biologists).

SyntenyVista includes also a top panel allowing additional user interaction and presenting extra information. The panel displays information on genes or QTLs in response to mouse movement in the gene area. QTLs are displayed as thin lines along the chromosome axes. The panel offers options to search for a gene name or a chromosome position. A gene is then highlighted on the whole chromosome image, and the gene and its counterpart in the other species blink for a few seconds.

One of the oldest but most frequently used tools - **DerBrowser** [61], see Figure 3.11 - was designed at the time of the human genome sequencing project. Since then new technologies have been developed, but not all of them can be added to this tool, because the software is out of date. It is a Java applet which supports interactive visualisation of one chromosome, or of a chromosome part. It connects to a local database to produce web pages showing all the information describing a given map object.

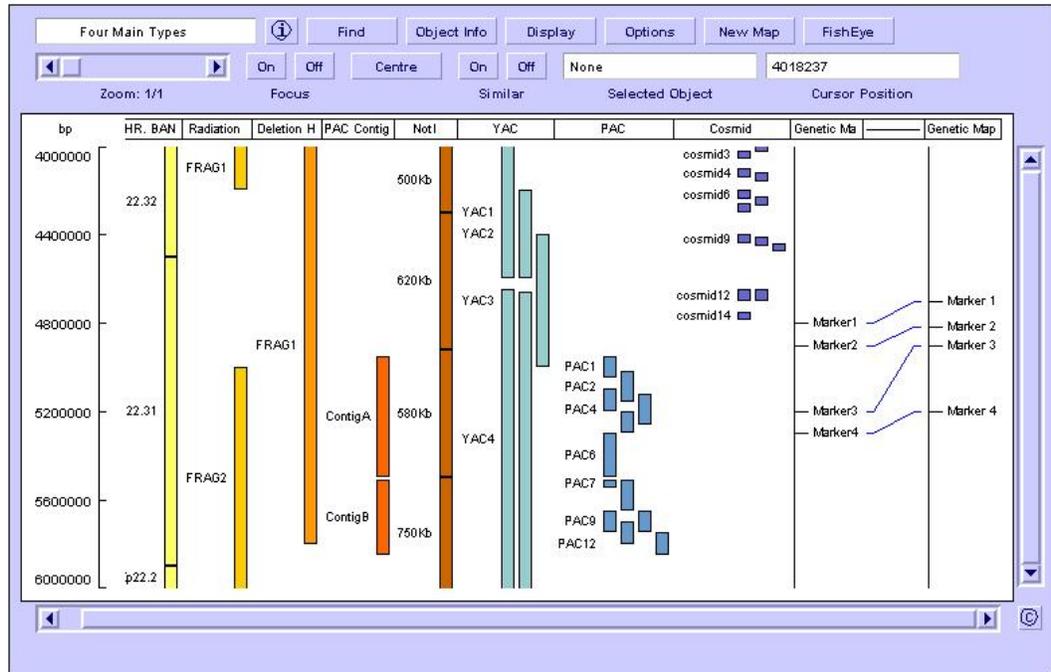


Figure 3.11: DerBrowser uses several colours and types of data presentation. There are columns with exemplary biological data presented in linear format (columns 2, 3, 4, 6, 7, and 8), simple overlap format (columns 1 and 5), and locus format (columns 9, 10, and 11) in different colours.

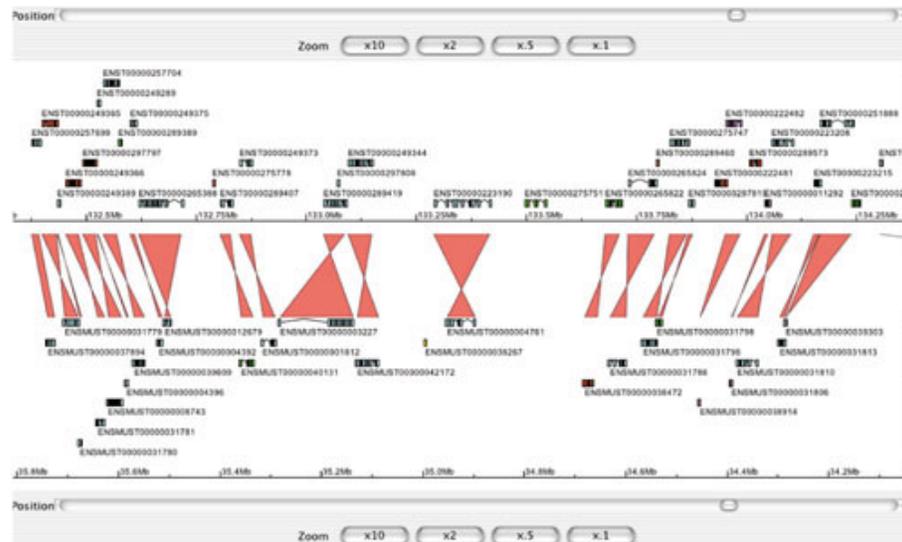


Figure 3.12: Chromosome comparison represented in Apollo. We see human chromosome 20 at the top and part of mouse chromosome 2 at the bottom.

It can be used to display genes, chromosome bands (chromosome parts coloured light or dark in the karyotype pictures), or markers, and can represent any object on a map [148]. It provides an illusion of smooth zooming (a slider), and supports the hiding of objects, based on object type. It can also perform search functions by using a “Find” button. The users still cannot see all the relevant areas in detail as the zooming is not powerful enough.

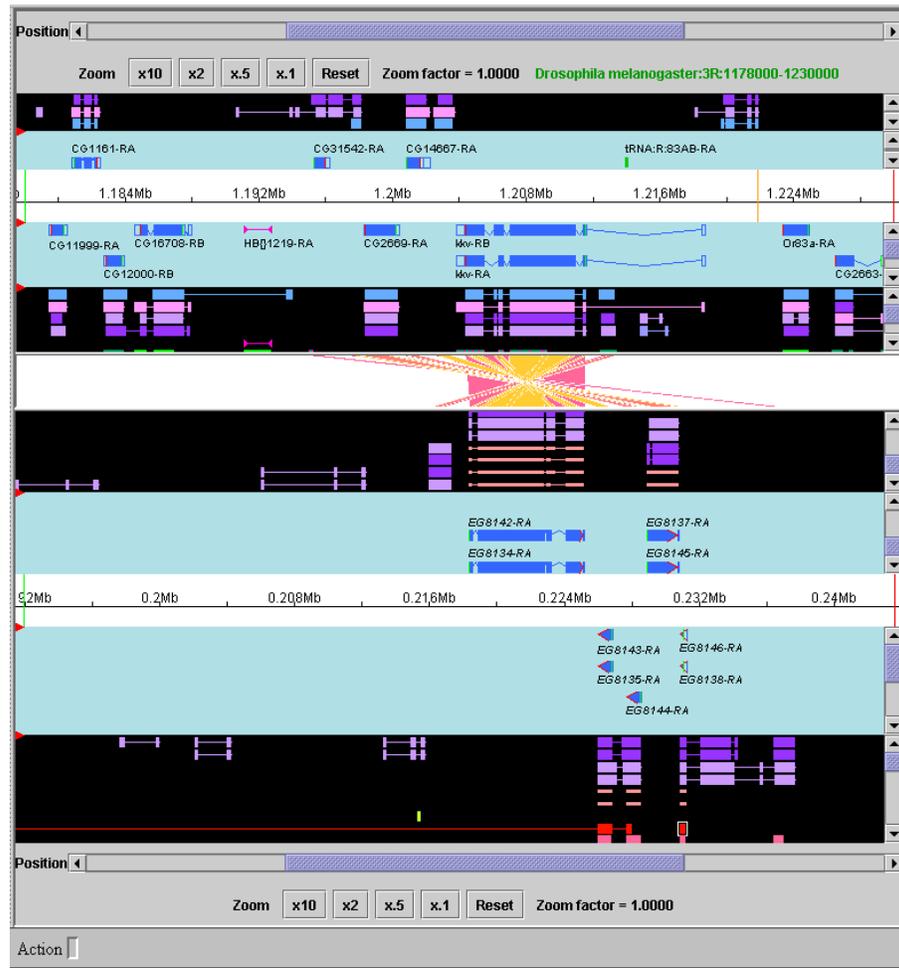


Figure 3.13: Apollo - gene visualisation. We see a comparison between two species *Drosophila pseudoobscura* and *Drosophila melanogaster*. Annotations from both organisms and orthologous regions between them are presented in the central part of the image, where many lines cross.

Apollo [63], see Figure 3.12 and Figure 3.13, is a sequence annotation viewer and editor. It allows the biologist to annotate the genomic feature descriptions derived from automated analyses and computational pipelines. It facilitates connecting to various databases and the comparison of existing annotations with other biological data. However, when the users want to choose the kind of connection they need and download the data, they see a lot of small windows, some of which offer scrolling, while some only inform about the chosen data after a mouse click. The tool offers researchers the ability to

probe, manipulate and alter the interpretation of the underlying data. Within the various views offered by the package, annotations can be created, deleted, merged, split, classified and commented upon. The tool allows the view to be scaled using zoom buttons and provides a degree of semantic zooming [5]. Some features are not displayed at low zoom levels and appear more precisely only when the user zooms in on them. The user can move to a specific position by specifying a coordinate, gene name, or short sequence string, or by using the horizontal scroll bar. Apollo can display features on two genomes at the same time. The view offers zooming and panning but it does not present the data clearly, and users cannot see all the relevant details. The pictures offered by Apollo show the connections between genes in the gene comparison view but they are unclear and lack sophistication. Because of the visual clutter the users can not see, or notice, all the interesting data details.

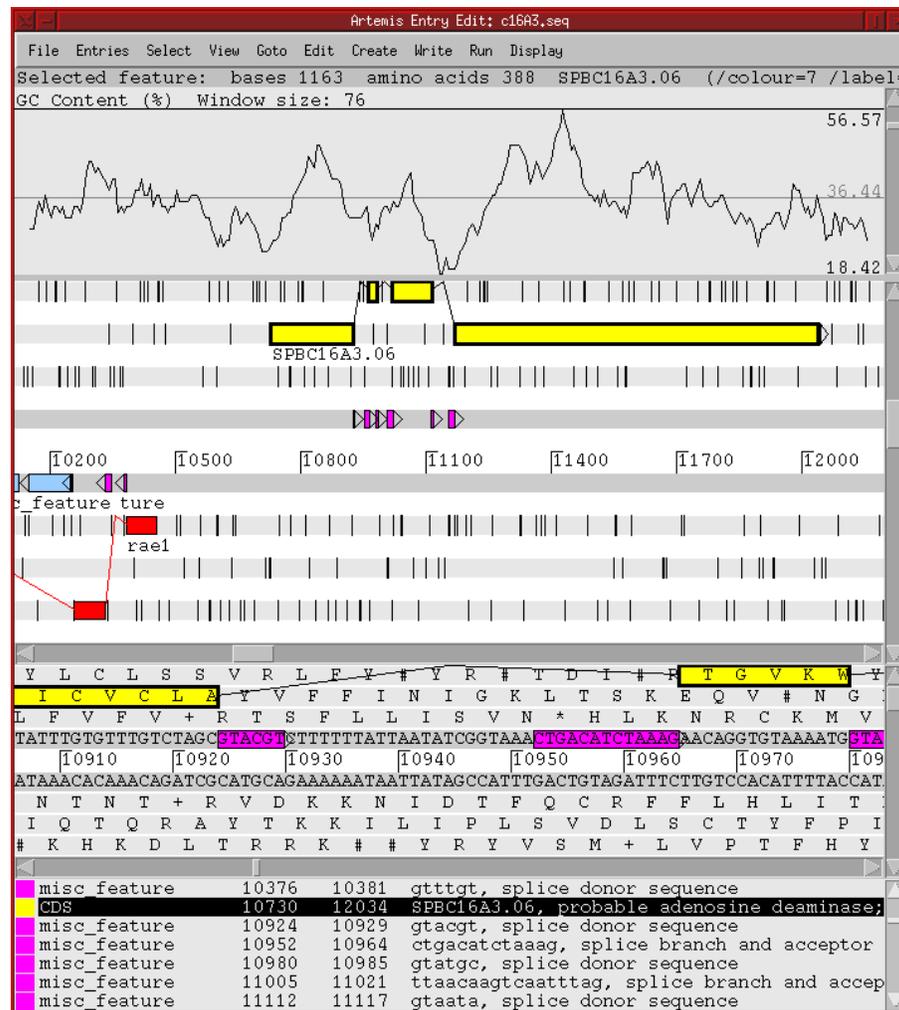


Figure 3.14: Artemis - gene presentation. The bottom window shows nucleotide sequences and translation of all three reading frames. In the middle window the zoomed version of the data from the bottom frame is presented and in the upper window the percentage of GC context is shown.

Artemis [98], see Figure 3.14, is a genome viewer and annotation tool that visualises sequence features and the results of analyses within the context of the sequence and its translation from DNA to protein. Artemis can be used as a sequence viewer and is suitable for smaller genomes. Properties of the sequence can be plotted. Each plot allows dynamic modification of the window size used for the calculation. The sequence and plots can be zoomed together into the single base level or out for the complete genome. Artemis provides two sequence windows to view the same sequence at different zoom levels simultaneously. The tool can be run as an applet within a web browser.

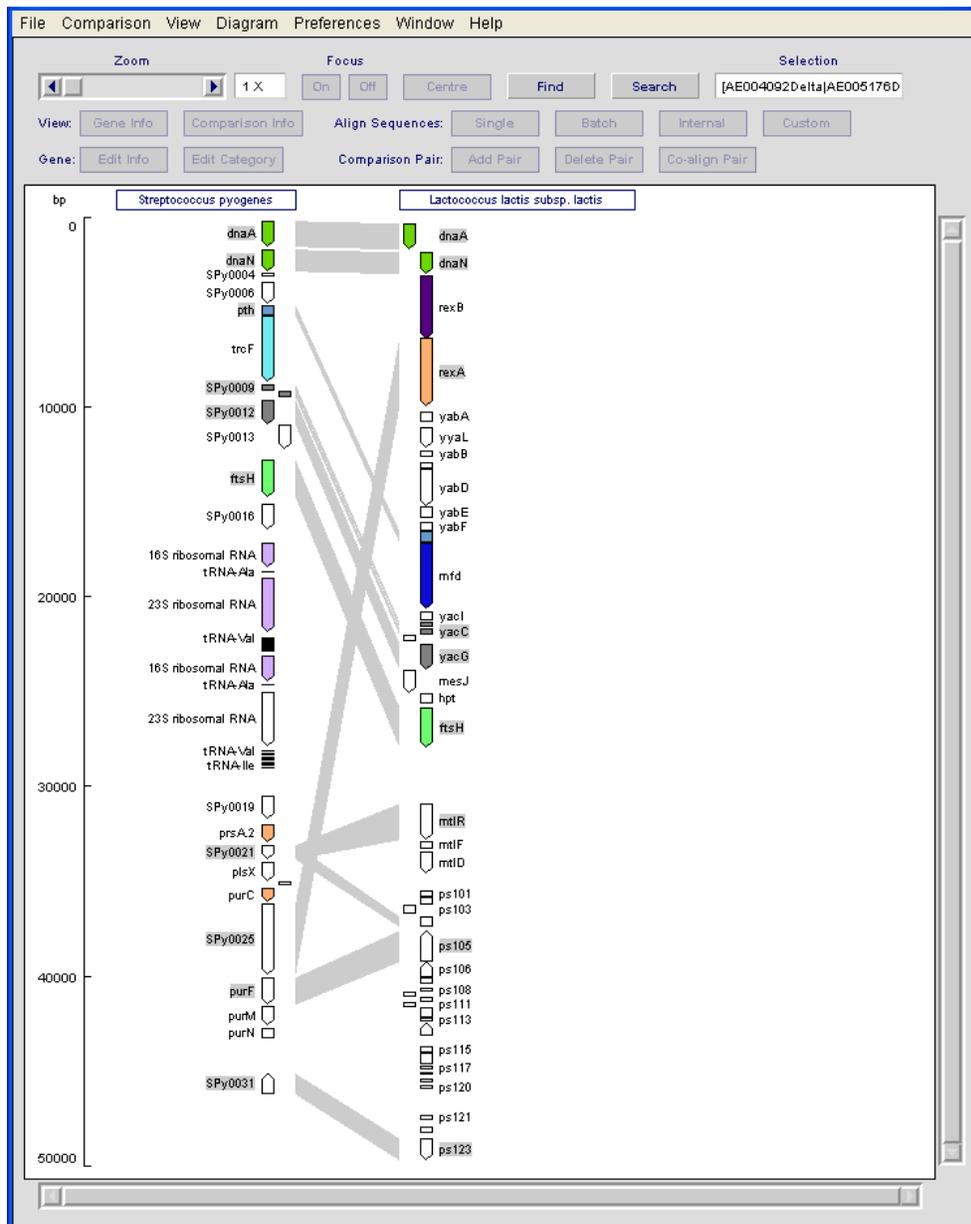


Figure 3.15: BugView - gene comparison.

BugView [60], see Figure 3.15, is a comparative genome browser. It allows one to compare the arrangement of genes in two genomes, and can also be used to view individual genomes. It was written to enable comparative study of bacteria, including the comparison of bacterial strains.

The view presented by BugView is restricted to genes, showing gene overlaps, and, where relevant, intron-exon structure, including alternative splicing. The users can scroll and zoom smoothly, and search for gene names. BugView includes support for the comparison of genes (sequence analysis) and analysis of gene alignments and other sequence features. For instance, one can filter sequence alignments by specifying percentage similarity.

BugView is a nice and simple tool but it does not offer any data connectivity at all. The users can only read the data from a prepared file in Genbank format. Its big advantage is that the users can choose the type of displayed diagrams such as linear, circular or comparison.

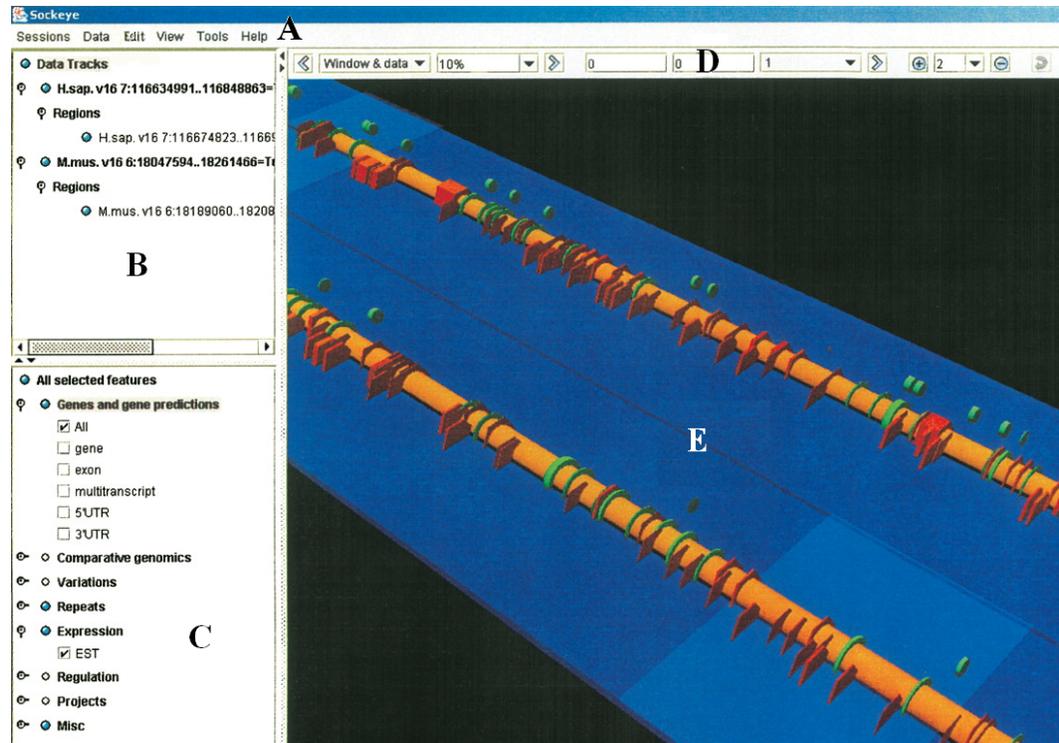


Figure 3.16: Sockeye chromosome visualisation in 3D. We see the menu (A), the sequence track selection tree (B), the feature selection tree (C), the navigation toolbar (D), and the 3D viewport (E). The application allows the users to show/hide and obtain detailed information for loaded sequence track annotation types. In 3D viewport the users can perform analysis and edit annotations. The figure adapted from [75].

Sockeye [75], see Figure 3.16, uses 3D graphics technology and data from the Ensembl database project. A user can also import custom sequences and annotation data. Large sets of functionally linked sequences containing genes that are coexpressed, and orthologous across multiple species can be

analysed in Sockeye. The application can also facilitate comparative analyses across sequences from any source. The difference between Sockeye and other existing browsers is in the 3D environment. View and annotation can exploit a third dimension. Each 3D model is specified in a user configurable XML format file. Sockeye integrates the process of obtaining sequence and annotation data. The application also allows a user to simultaneously visualise several different alignments and easily view alignment gaps. Montgomery et al. [75] stress that using a 3D environment has a lot of advantages and disadvantages and only few researchers decided to use it in their work. 3D visualisation is uncommon in genomics and for abstract data, moreover, in 3D graphics problems with occlusion and movement occur, because of this researchers may find it difficult to use. The developers find Sockeye to be user-friendly, but the users cannot easily see all interesting objects (there was no formal user study). The designers' goal is to provide the user with a system for locating and extracting targets from comparative genomics analyses for subsequent laboratory and computational study. Sockeye has been constructed as a software application capable of analysing and comparing the characteristics of several genome annotations simultaneously. The application creates simple annotation objects called TrackFeatures to meet the challenges of storing and managing the visualisation of genomic annotations. Sockeye's graphical user interface shows the sequence track selection tree, the feature selection tree, several navigation controls, and the 3D viewport with one sequence track. The users can also compare the information contained in multiple genomics sequences, and zoom, pan and rotate the position of the sequence track. Sockeye can adapt to the changing needs of its user community and provides its own error tracking and feature suggestion mechanism. The authors believe that an information-rich 3D environment will allow the user to see the underlying characteristics of multiple sequences within a single genome or from multiple genomes. However, during our user studies we had not met users who had used Sockeye.

K-BROWSER [13], see Figure 3.17 and Figure 3.18, is a comparative browser which visualises biological information at a higher level of resolution than is the case in most other tools. Its novelty is the representation of sequence similarity histograms along sequence features on several genomes. K-BROWSER was built on the foundation of the UCSC Genome Browser [56]. It can display a number of genomes overlaid with annotations and predictions, and shows the multiple alignments that describe global sequence relationships.

K-BROWSER takes as input a specific region in a genome and produces a set of images that succinctly represent the requested region and all orthologous regions in other genomes. The two critical components of the application are track realignment and image generation. Track realignment is responsible for the necessary scaling of DNA lengths in the comparative views to make it consistent with the multiple alignment. Image generation takes as input a genomic region query and produces an image for every corresponding region in the multiple alignment. The tool displays a sequence conservation plot above the tracks. It allows the users to select a track according to which the conservation plot is to be coloured. The K-BROWSER can compute the percentage similarity between the root sequence and the leaf sequence

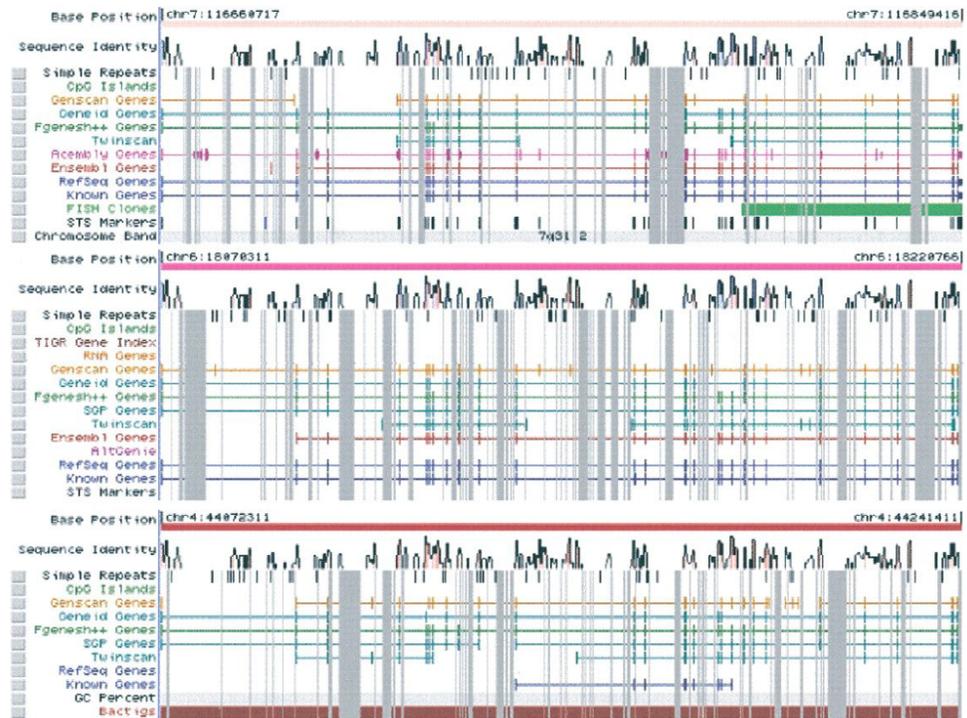


Figure 3.17: K-BROWSER showing the cystic fibrosis gene region (CFTR). Human, mouse and rat annotations are presented (from the top to the bottom). The grey bars indicate gaps (arising from insertions or deletions) in the sequences and the peaks in the histogram show regions of high sequence similarity. The user can navigate using zoom buttons, gene name searching, and position jumping.

in a window centred on a specified position. It allows one not only to determine if a genomic region is conserved within other genomes, but also to infer the rate at which it is evolving. **The Generic Genome Browser** [107], see Figure 3.19, is a combination of a database and interactive web pages for manipulating and displaying annotations on genomes. GBrowse can display an arbitrary set of features on a nucleotide or protein sequence, and can also accommodate genome-scale sequences. GBrowse provides most of the features available in other browsers, and was designed from the outset to be portable and extensible. This allows it to integrate well with other components of a model organism system database (MOD), and with planned components of the Genetic Model Organism Database (GMOD) project in particular. GBrowse provides multiple configurable levels of zoom and two scroll speeds, and also offers semantic zooming. The users can customise the view, the track and the width of the image. The application allows for adding annotations to the genome and publish those annotations by putting the feature file on an internet accessible web page or FTP (file transfer protocol) site. GBrowse implements a web based display. It is composed from several components such as a CGI (common gateway interface) script, the Bio::Graphic module responsible for rendering the genome images, the Bio::DB::GFF database and GadFly database. The Bio::DB::GFF schema has a few limitations. The main one is that it relies on a flat coordinate system

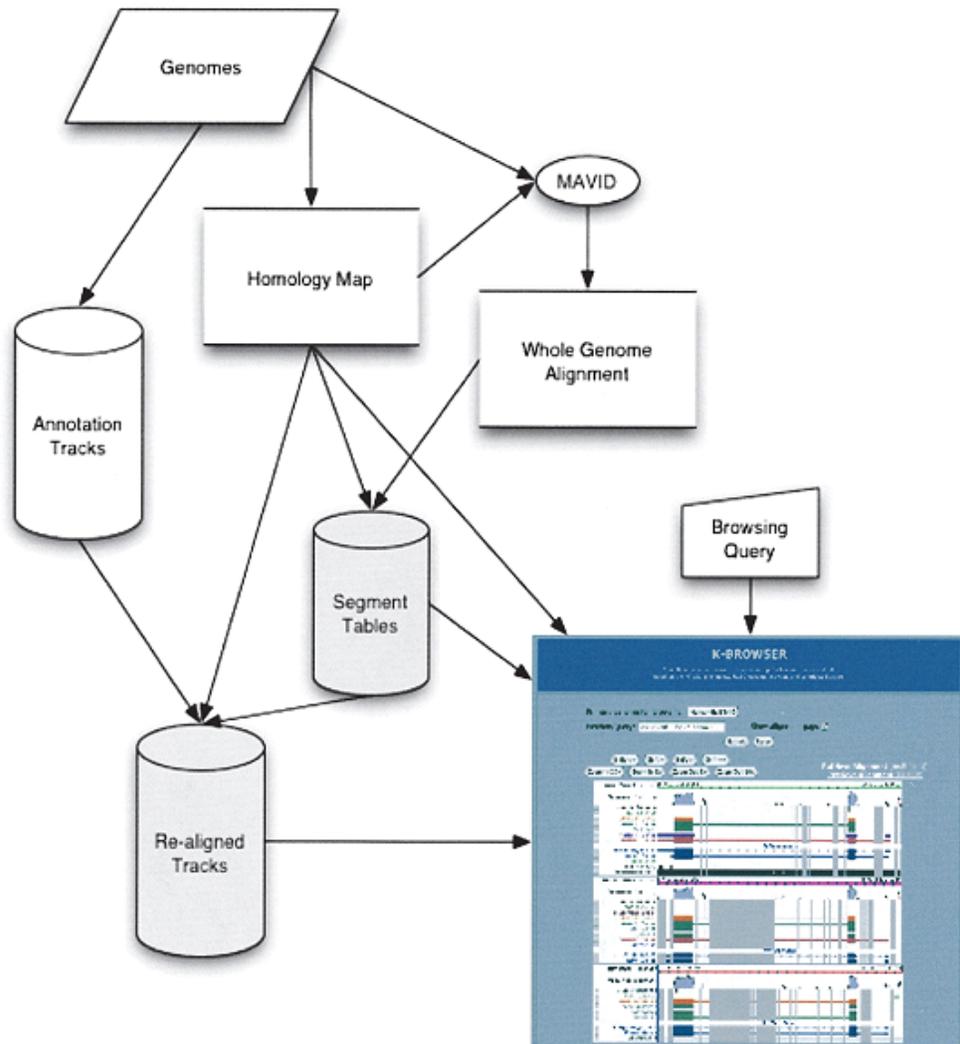


Figure 3.18: K-BROWSER - genome browser application, showing K-BROWSER infrastructure. Segment tables and re-aligned tracks are the two new databases built by the application. These databases are used in whole-genome alignment, homology map and annotation tracks. The users submit queries which are converted into a K-BROWSER figure.

to represent genomic features and can handle only one level of nesting of sequence features. GadFly was developed to solve these problems. GBrowse supports also a plug-in architecture that allows third-party modules to extend GBrowse's capabilities. Stein et al. [107] stress that the application was designed with extensibility in mind. Hence, there are multiple distinct layers at which software developers can add new code to extend the browser's capabilities. The developers stress that Web-based genome browsers are a class of applications that the bioinformatics research community seems to be doomed to reinvent time and again.

The National Center for Biotechnology Information (NCBI) Map Viewer [143], see



Figure 3.19: GBrowse. The users can type a landmark name into the text field at top. Landmarks can be gene names, clone names, accession numbers, or any other identifier configured by the administrator. Once a region is selected, it is displayed in a detailed view that summarises annotations and other genomic features.

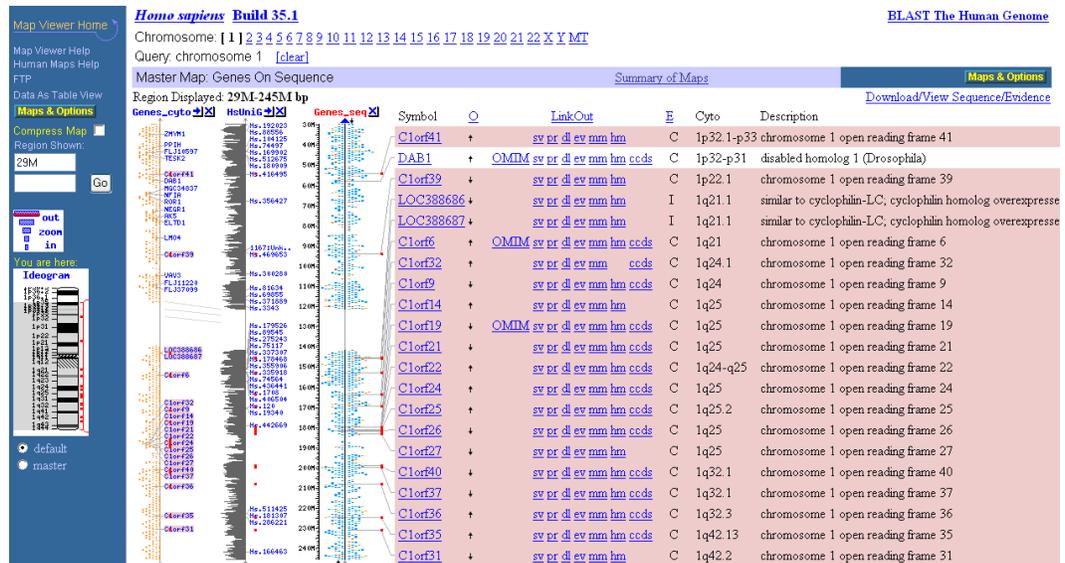


Figure 3.20: NCBI Map Viewer - genome browser application showing the human chromosome 1 with features as genes and micro array probes, and some additional information such as description, data sources, and links to other biological data bases. On the left hand we see an overview of the chromosome 1 with the selected part, which can be zoomed and presented in the main window.

Figure 3.20, is a Web interface used to view and search an organism's complete genome. The user can view maps of individual chromosomes and zoom into specific regions within chromosomes to explore the genome at the sequence level. They have access to several different types of maps for different organisms. Map Viewer allows the user to view these maps graphically or in a table format. NCBI Map Viewer's graphic display is limited to features related to gene identification, although there are text links to other pages. Zooming and other visualisation features are not as sophisticated, in our opinion, as those offered in Ensembl.

NCBI Map Viewer and Ensembl read data directly from a huge database and offer search functions. One of the drawbacks of connectivity via the internet is much slower response each time when the users try to perform an operation such as scroll or zoom. These sites have implemented their browsing facilities solely in HTML so that each change of view involves generation of a new web page. Such web pages cannot be displayed until the necessary bitmap graphic files have been transferred by the Internet from the remote server to the users' machine. On the other hand, they provide access to a wide range of information held in their databases. NCBI Map Viewer's graphic display is limited to features related to gene identification, although there are text links to other pages. Users familiar with the tool can add additional features.

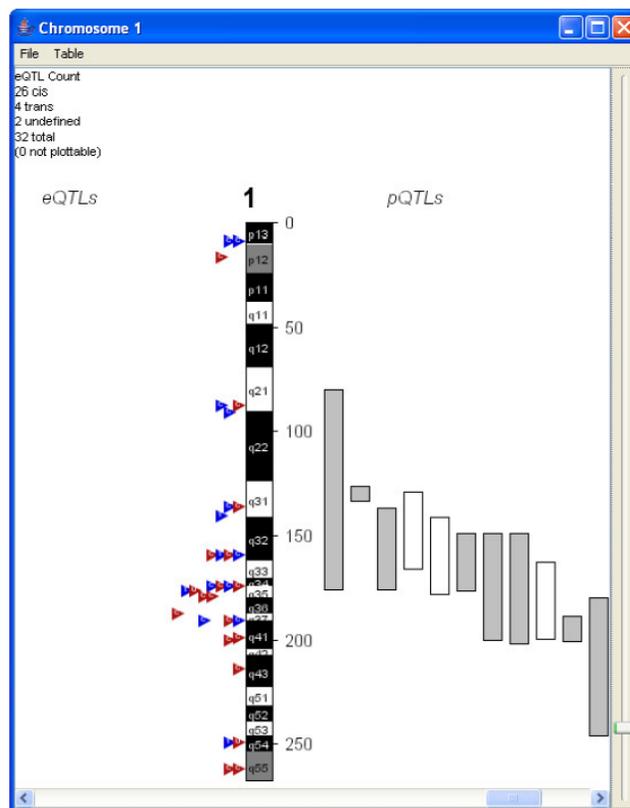


Figure 3.21: eQTL Explorer presents the rat chromosome 1, eQTLs marked by triangles to the left and pQTLs marked by rectangles to the right.

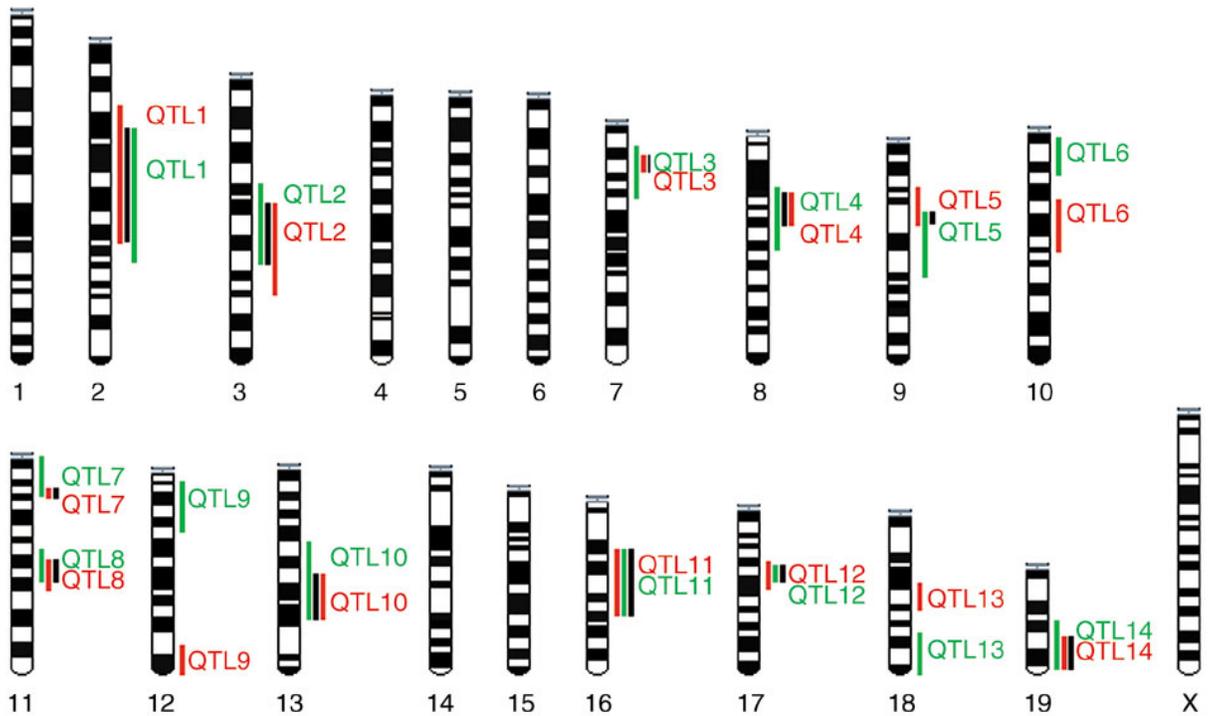


Figure 3.22: Expressionview displays mouse chromosomes and QTLs represented as vertical bars beside each chromosome.

eQTL (expression Quantitative Trait Locus) Explorer [77], see Figure 3.21, visualises QTL data [45] on the background of each chromosome. The chromosomes are drawn as vertical bars, and the QTLs are shown as coloured triangles. The application can display individual chromosomes in a separate view, with options to browse, zoom and export data. The tool has also a pop-up menu which provides access to annotations and cross-references to external data sources. The tool represents only a small subset of genome data.

Expressionview [31], see Figure 3.22, and eQTL Explorer, which are similar in appearance, are two applications designed specifically for the analysis of micro array experiments. Both applications show entire karyotypes and draw QTLs and genes identified in a micro array experiment alongside the chromosomes. Both applications are single-purpose, in that they do not show other biologically relevant information at the same time, for instance all the genes or SNPs.

A triple synteny Human-Mouse-Rat visualisation [135], see Figure 3.23, allows us to observe and compare genes and chromosomes in a similar way to SyntenyVista. Human-mouse-rat synteny server [135] provides information about 18,915 human genes mapped to mouse genome draft and 18,464 mouse genes mapped to human genome draft, among them 14,504 orthologous gene pairs. The authors claim that this is the most comprehensive data about homology between human, mouse and rat genomic regions.

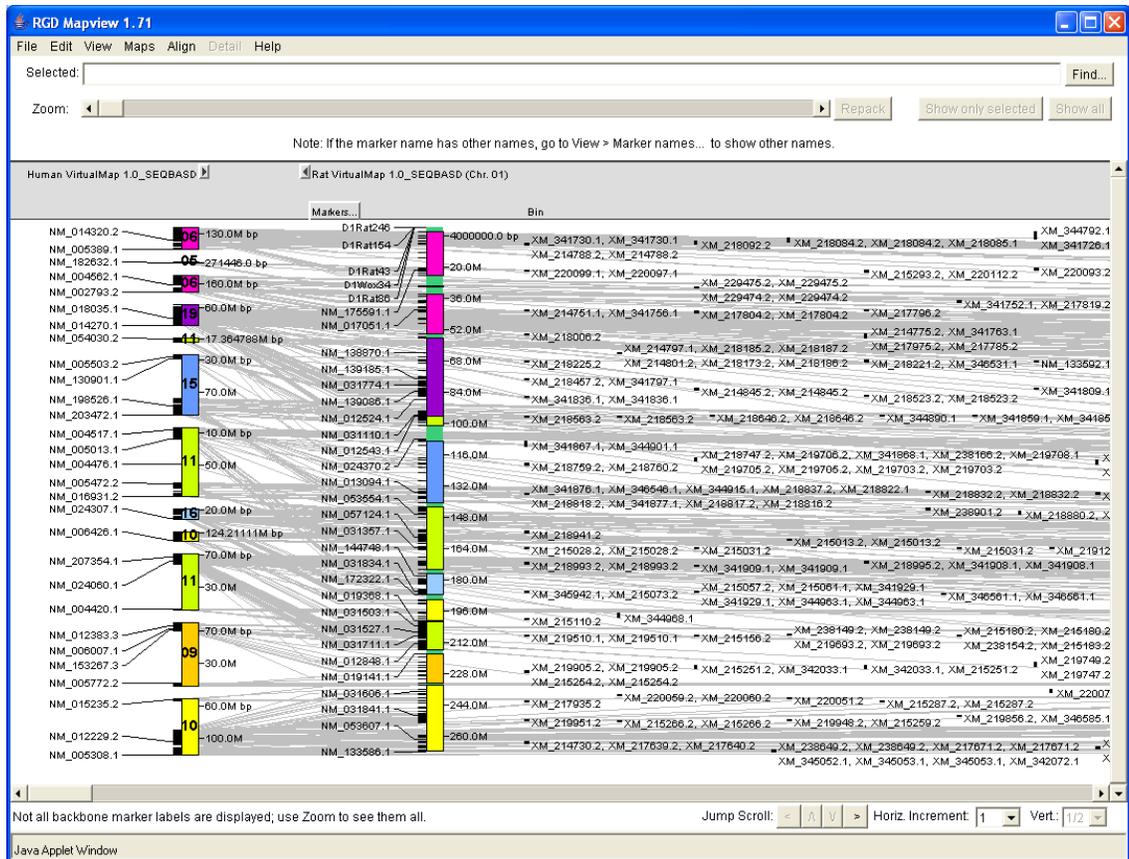


Figure 3.24: VCMa picture from the rat genome database. This is a ComparativeMap for the rat chromosome 1 and the human chromosome 1.

The user can find a good deal of textual biological information, however its visualisation is quite poorly designed. A triple synteny Human-Mouse-Rat visualisation offers colourful pictures for each rat, mouse, and human chromosomes, see Figure 3.23. However, it does not even support basic techniques such as zooming or panning. The user can click at a chromosome, see Figure 3.23 A, and then they see a comparison between three species, see Figure 3.23 B. Clicking is the only interaction offered by the tool.

A similar function to the Triple synteny Human-Mouse-Rat is presented in the **Virtual Comparative Mapping** (VCMa) from the rat genome database (RGD) [144]. The tool was developed to explore the relationships between rat, mouse and human. The comparative maps are extended by the prediction of locations for markers unmapped in a species based on their known locations in the syntenic region of another species. The application [58] gives a visually pleasing picture of the comparative maps but, because of the density of markers on the map, the tool does not display all available information. As can be seen in Figure 3.24 a lot of information is presented. The users can click on a bar of the map or explore a searched feature by clicking on it. The display is cluttered and the system does not support synteny analysis.

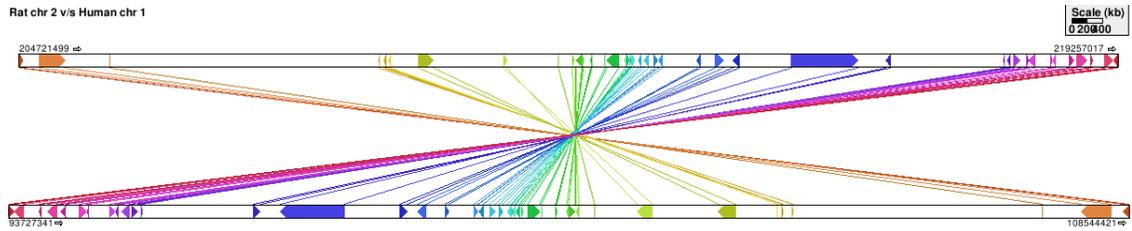


Figure 3.25: Cinteny represents a comparison between the rat chromosome 2 and the human chromosome 1.

Cinteny [103], see Figure 3.25, similarly to SyntenyVista, represents syntenic regions between species. The application represents markers and genes but its visualisation aspect is limited. It offers zooming, but it is available only by clicking on an element, and the user cannot zoom enough to see details. The advantage of Cinteny is that the graphical elements it presents are linked to external resources - here NCBI.

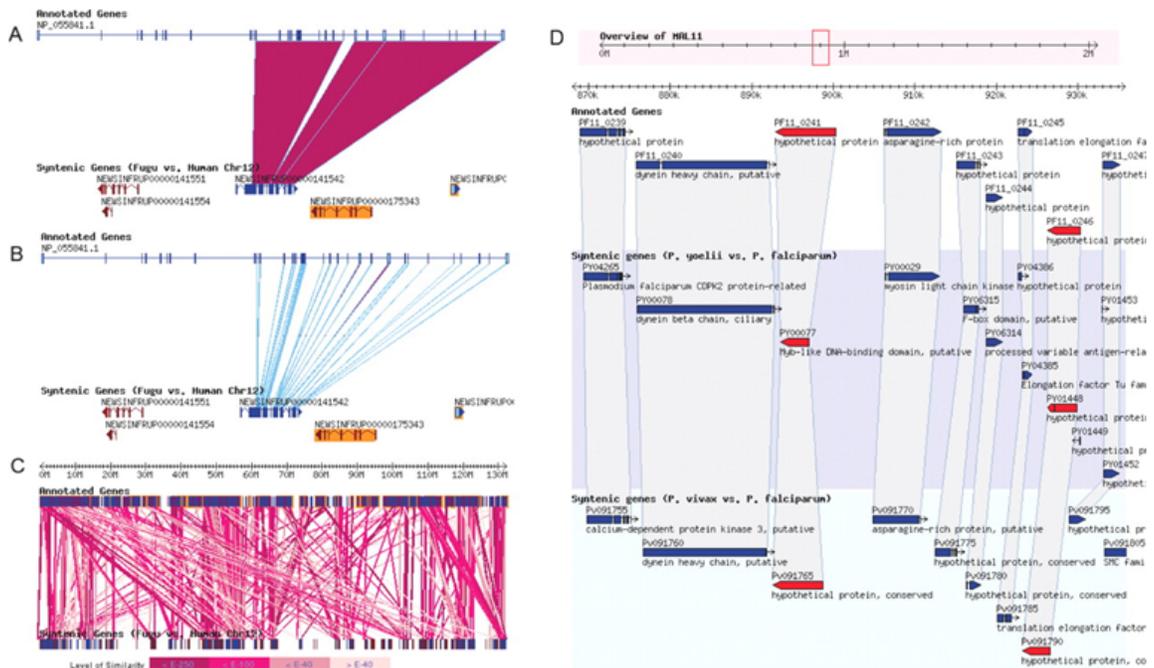


Figure 3.26: SynView - the figure taken from [114]. A displays similarity between orthologous genes, NP_055841.1 (human) and NEWSINFRUP00000141542 (fugu). B represents an exon level display for the same genes as in panel A. C shows a panoramic display of the human chromosome 12 (130 Mbp region) and the whole fugu genome. D is a multiple species display.

SynView [114], see Figure 3.26, also presents comparative genome analyses. The difference between SynView and other applications presented in the section is that SynView offers comparative genome

analysis for a number of species. On the other hand, the visualisation is not clear, and the user can easily get lost in a number of links representing the genome similarities. SynView is integrated with GBrowse.



Figure 3.27: BioViews - the figure taken from [39], presents a DNA view.

BioViews [39], see Figure 3.27, is yet another example of an older genome browser. The authors discuss the issue of displaying physical and sequence based maps³. The application also displays a genetic map which was common before the genome was sequenced. It offers user customization and graphical hyperlinks for GenBank records. It also supports semantic zooming.

SyMAP (Synteny Mapping and Analysis Program) [105], see Figure 3.28 visualises synteny blocks and presents the results as a dot plot or as a standard synteny block in chromosome regions. SyMAP provides its own algorithm to compute synteny. The user can set in the control panel the percentage of identity in synteny which allows them to see only the selected synteny blocks. SyMAP offers also a

³Physical maps show the physical length of DNA measured in base pairs. In molecular biology, two nucleotides on opposite complementary DNA or RNA strands that are connected via hydrogen bonds are called a base pair (often abbreviated bp). In the canonical Watson-Crick base pairing, adenine (A) forms a base pair with thymine (T), as does guanine (G) with cytosine (C) in DNA. Sequence-based maps improve with the scientific progress and are perfect when the genomic DNA sequencing of the species has been completed. In the meantime, it is worth to mention the genetic map, which was used when physical and sequence-based maps did not exist. A scale in a genetic map was measured in centimorgan (cM) or map unit (m.u.), which is a unit of recombinant frequency for measuring genetic linkage. It is often used to imply distance along a chromosome. The number of base-pairs it corresponds to varies widely across the genome (different regions of a chromosome have different propensities towards crossover), and is about 1 million base pairs in humans. The centimorgan is equal to a 1% chance that a marker at one genetic locus on a chromosome will be separated from a marker at a second locus due to crossing over in a single generation. A 50 cM distance means that the genes will reassort when an odd number of crossings happen, which happens 31.8% of the time.

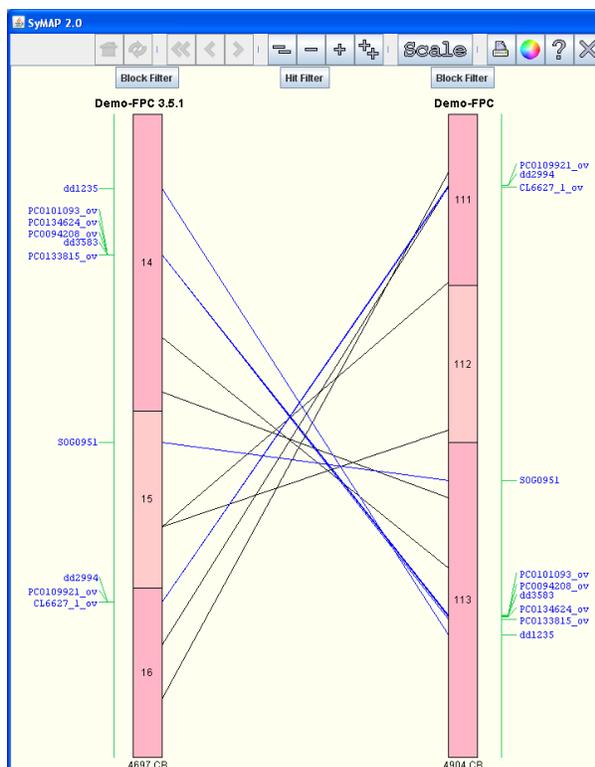


Figure 3.28: SyMAP - synteny blocks in chromosomes. Figure obtained from [147].

sequence filter which displays markers and genes. However, gene descriptions appear only when the user moves the mouse close to the gene. Markers can be displayed with all descriptions which makes the view neither aesthetically pleasing nor readable. SyMAP offers zooming limited by buttons and scrolling which appears when it is necessary.

3.3.2 Metabolic Pathways

In the previous section we introduced the most popular “classic” genome browsers. We now move to more complicated visualisation tools which visualise metabolic pathways, see Figure 3.30 and Figure 3.29. Metabolic pathways allow the users to draw and manipulate the representations of metabolism. Metabolic pathways (metabolic maps) are subnetworks of the metabolic networks. The structures represented by this kind of tools are much more complicated than those shown in classic genome browsers. Biological networks are studied to discover complex roles played by genes, gene products and the cellular environments in biological processes. In this kind of network the nodes represent genes or gene products and the edges represent specific interactions. One uses specific graph layouts and methods which enable the drawing of genome metabolic networks [10], biomolecular interaction networks [100] and other kinds of biological networks. It is very important that common activities such as selection or filtering are implemented in a

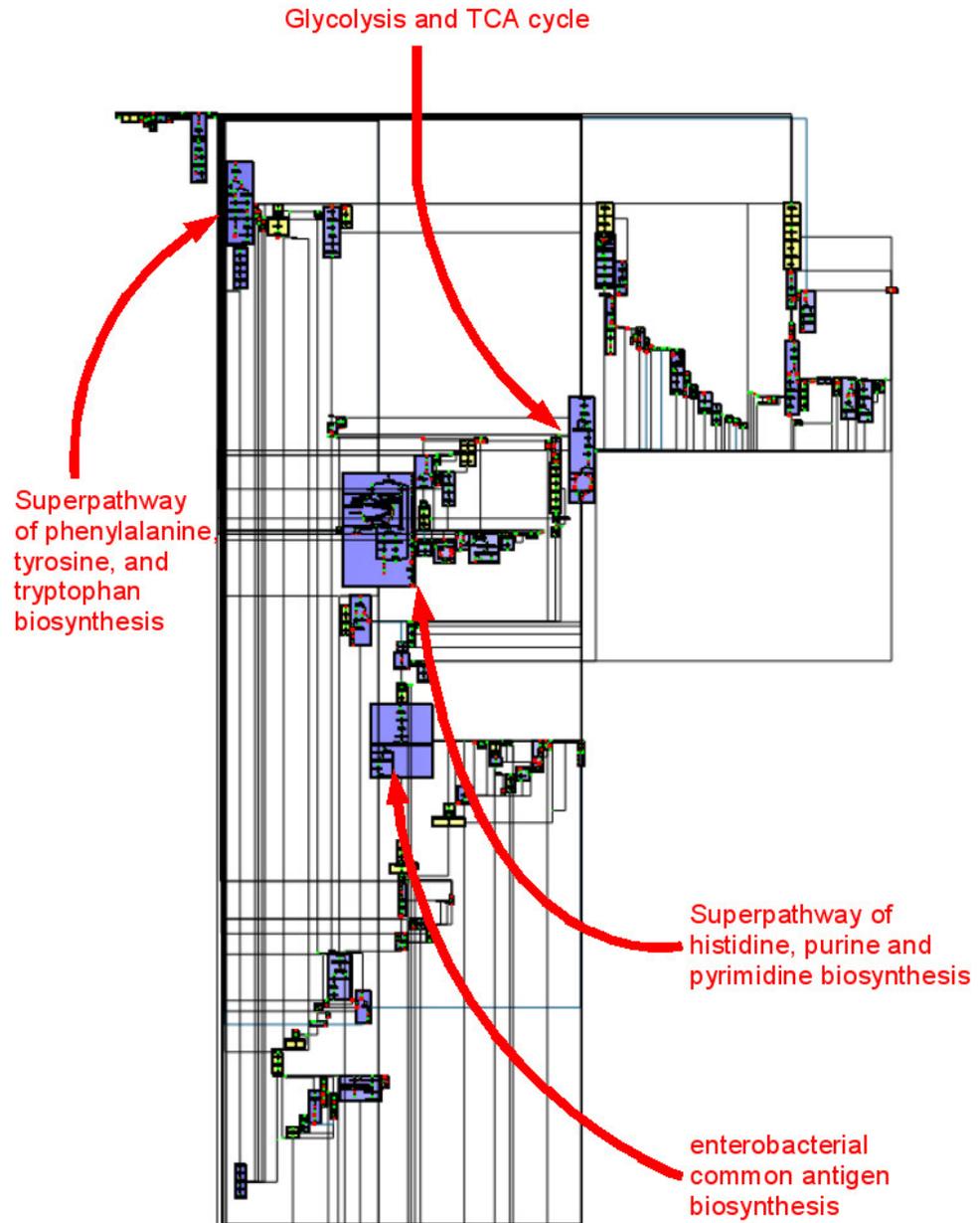


Figure 3.29: Whole metabolic network of *E. coli* drawn by MetaViz. The figure is taken from [10].

way that produces the results quickly. Cytospace [100] provides smooth zooming and overview and details techniques which allow the users to see where the part of network displayed in the main view is situated and quickly switch between different parts of the network.

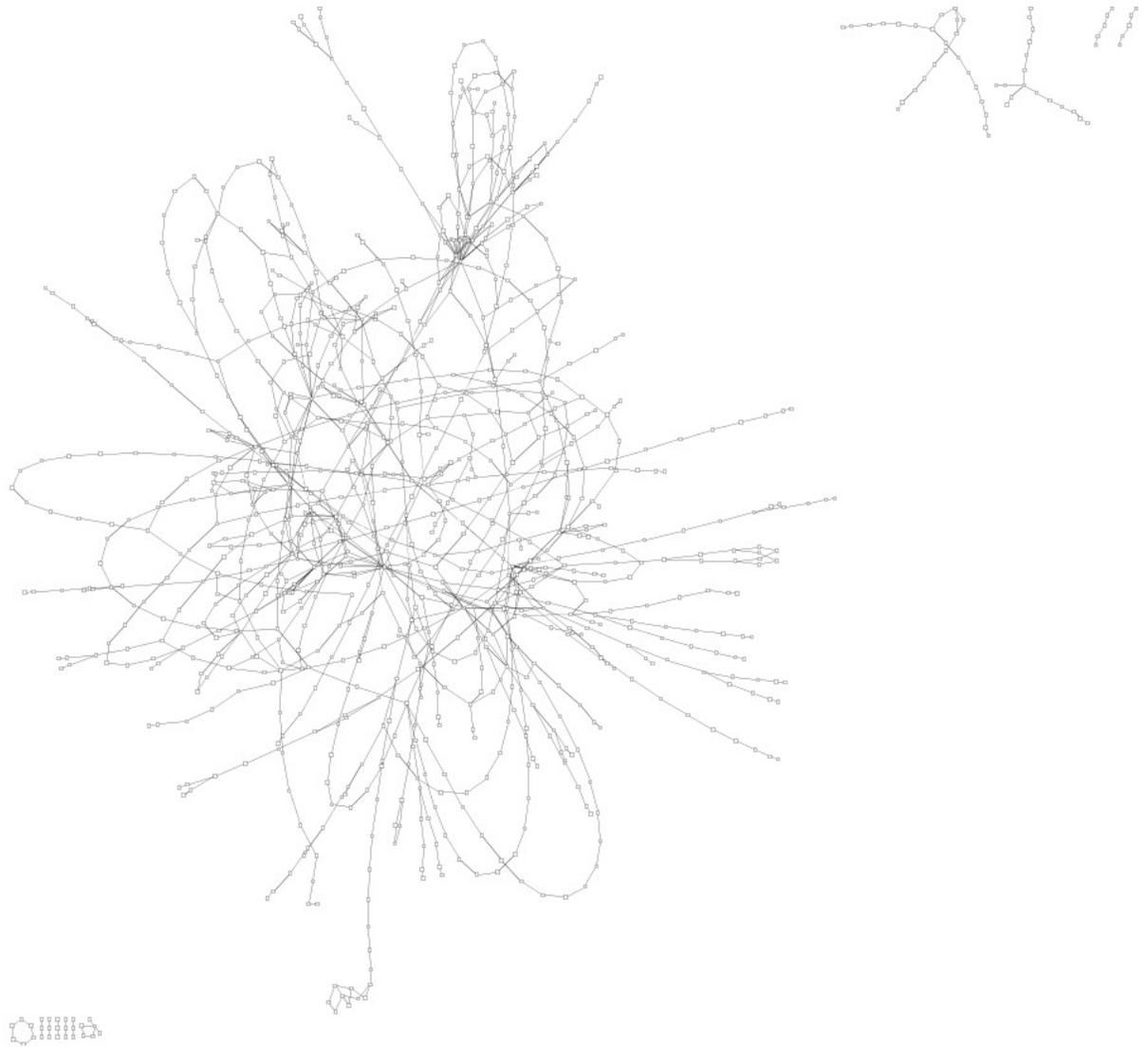


Figure 3.30: Cytoscape - whole metabolic network of *E. coli*. The figure is taken from [10].

3.3.3 Other Data Analysis Tools

As can be seen in Chapter 8 - ‘Mixed Paradigm User Study’ - a lot of biologists use Excel or Word, or even a simple text editor to store and manipulate their data experiments. Therefore, it is also worth mentioning such tools. They can present biological data and use a graphic interface (Excel and Word) but for biologists tables or simple text files which have no graphic representations are also important. Some tools in that domain were created specifically for biological data, such as Fisheye-Mage [118] which shows micro array data or FlyMine [130] which represents data in web page tables. Tools mentioned in this chapter, such as a triple synteny Human-Mouse-Rat visualisation [135] or Ensembl, specifically BioMart [122], also present data as text in table format. In those tools, techniques supporting filtering or searching

are very important.

The screenshot shows the MAGE-ML Viewer application. On the left is a sidebar with a list of data items, including 'MeasuredBioAssayData:FLYC:16' which is selected. The main window displays a table with the following data:

Gene	Parameters	Summary	med (Cy3)	med (Cy5)	st dev (Cy3)
Pu-FLYC-1:1.1.1			1970.00	1429.00	34371.36
Pu-FLYC-1:1.1.2			1776.00	1623.00	1156.89
Pu-FLYC-1:1.1.3			1230.00	1950.00	1787.78
Pu-FLYC-1:1.1.4			2633.00	2972.00	5254.49
Pu-FLYC-1:1.1.5			6804.00	9445.00	4891.03
Pu-FLYC-1:1.1.6			1016.00	1307.00	565.73
F:A-FLYC-1:1.1.1.7			12342.00	14128.00	7998.18
Pu-FLYC-1:1.1.8			1783.00	2224.00	923.98
Pu-FLYC-1:1.1.9			2215.00	2628.00	1881.54
Pu-FLYC-1:1.1.10			10311.00	13060.00	7572.81
Pu-FLYC-1:1.1.11			1546.00	2277.00	1764.27
Pu-FLYC-1:1.1.12			5066.00	6666.00	1061.19
Pu-FLYC-1:1.1.13			1111.00	1200.00	2571.41

Figure 3.31: Focus on a specific gene presented by Fisheye-Mage from [118].

Wu et al. [118], see Figure 3.31, reported on a table that uses fisheye distortion. The table showing gene expression data was a subject of a pilot user study. 5 people (not experts) who took part in the study were asked questions from the Questionnaire for User Interface Satisfaction (QUIS) [104]. Wu and colleagues described the user feedback as quite positive, with a mean overall reaction score of 8.2 (on a scale from zero to nine). They also stress that the application shows “real promise”. However, the authors did not precisely describe the experiment and mentioned nothing about data presented during it. We would rather call it “asking colleagues for advice” than user study.

In Fisheye-Mage, the authors do not present a special graphical user interface but only a simple table with fisheye effect which shows biological data.

FlyMine [130], see Figure 3.32, and BioMart [122], see Figure 3.33, are databases which present data, FlyMine for Drosophila and Anopheles genomics, and BioMart for selected mammals and other species. The two applications provide tables with textual data, however, under the text they hold hyperlinks to additional information - frequently also graphical representations.

As can be seen in Chapters 6 and 8 (our user studies), the textual information is very important for biologists and quite often they prefer to have the information in both versions, textual and graphical.

Trail: [Query](#) > [Results](#)

Results for template: Gene [D. melanogaster] --> Human homologue + disease associated with this homologue.
For a given Drosophila gene, show the homologous human gene (and the BlastMatch E-Value) and any associated disease, according to Homophila.

Drosophila gene identifier	Drosophila gene symbol	Human gene symbol	Omim disease > description	Omim disease omimId
CG6235	twc	PAFAH1B1	Lissencephaly-1, 607432 (3);Subcortical laminar heterotopia (3) ...	601545
CG6235	twc	PPP2R2B	Spinocerebellar ataxia 12, 604326 (3)	604325

Figure 3.32: FlyMine - figure adapted from [130].

Ensembl Gene ID	Ensembl Transcript ID	Reference ID	Gene End (bp)	Strand
ENSRNOG00000002495	ENSRNOT00000003371	ENSRNOSNP2258060	169128	-1
ENSRNOG00000002495	ENSRNOT00000003371	ENSRNOSNP2258062	169128	-1
ENSRNOG00000002495	ENSRNOT00000003371	ENSRNOSNP2258065	169128	-1
ENSRNOG00000002495	ENSRNOT00000003371	ENSRNOSNP2258071	169128	-1
ENSRNOG00000002495	ENSRNOT00000003371	ENSRNOSNP2259477	169128	-1
ENSRNOG00000002495	ENSRNOT00000003371	ENSRNOSNP2261106	169128	-1
ENSRNOG000000040320	ENSRNOT000000062063	ENSRNOSNP2268352	211973	-1
ENSRNOG000000024525	ENSRNOT00000003432	ENSRNOSNP2280029	377817	1
ENSRNOG000000024372	ENSRNOT000000030803	ENSRNOSNP2293422	482123	1
ENSRNOG000000024372	ENSRNOT000000030803	ENSRNOSNP2298783	482123	1

Figure 3.33: BioMart - figure presented data for rat genes.

3.4 Work Undertaken with Existing Genome Browsers

As can be seen from previous section, we studied a number of existing genome browsers but none of them fulfill the biologists requirements (see Table 3.1). Therefore, we decided to extend existing genome browsers and add additional features and visualisation techniques (fiseye) to make the tools more usable. We chosen two genome browsers SyntenyVista [46] and DerBrowser [61], because our collaborators use these tools during their work.

3.4.1 Software Re-engineering for Database Connectivity in SyntenyVista

We started the software engineering work by addressing the database connectivity issue. We introduced to SyntenyVista an XML (extensible markup language) configuration file which allows one to easily change

the database connection without changing the source code. Configuration is performed only using a text editor. Figure 3.34 shows some sample XML tags used in the configuration file. We also tried to use a web server on <http://penida.dcs.gla.ac.uk> to collect necessary data, especially about QTLs which were temporarily stored on a web server in the hospital (Western Infirmary). SyntenyVista displays QTL data, and we encountered problems with access to the hospital server which was outside our control, so we resolved to replicate this data locally.

```

1 <SyntenyVista>
2
3 <db_name>ensembl_mart_30_1</db_name>
4 <UserName>anonymous</UserName>
5 <Password>
6
7 </SyntenyVista>

```

Figure 3.34: Example XML configuration file.

3.4.2 Software Development of DerBrowser

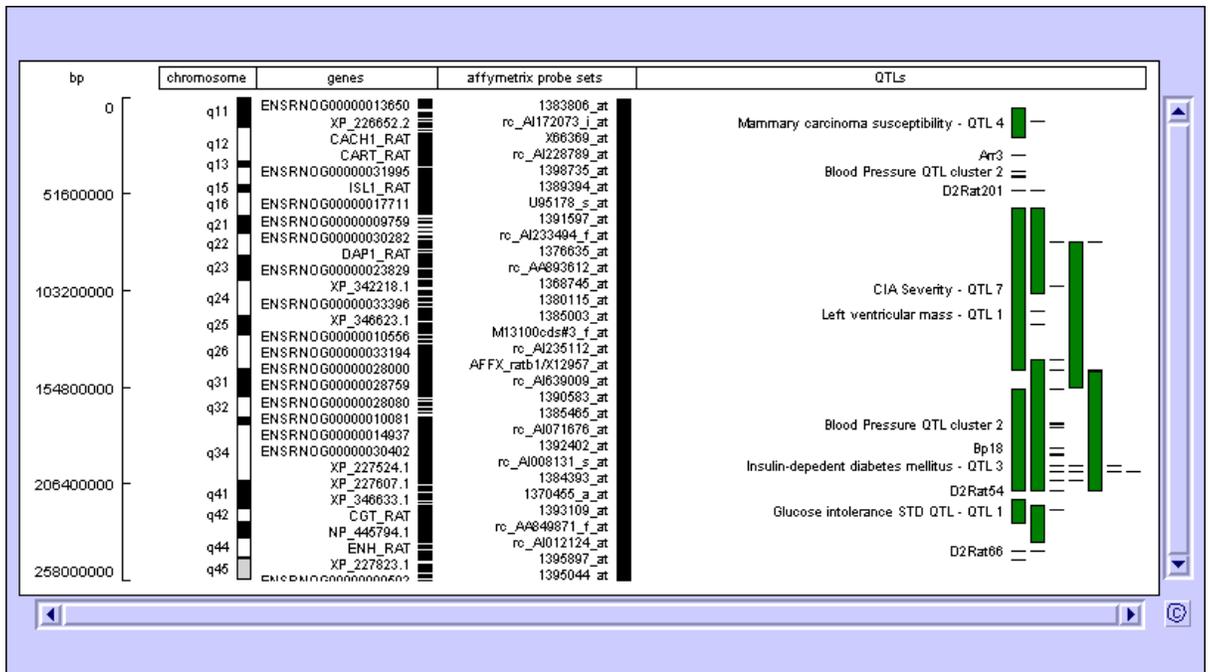


Figure 3.35: DerBrowser - rat chromosome 2. On the left hand side the chromosome scale is shown. The column called chromosome shows the chromosome bands which are visible under a microscope after a chromosome is stained. Genes and Affymetrix probe sets have been acquired from Ensembl. QTLs shown are areas on the genome associated with genetic defects and disease, some of which were first proposed in [45], and some downloaded from the rat database [144] or Ensembl.

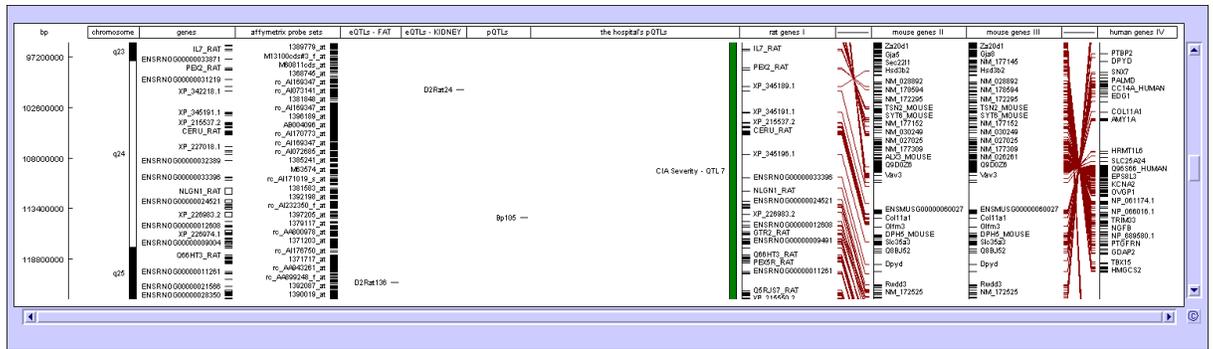


Figure 3.36: DerBrowser - chromosome comparison. On the left hand side only small part of the rat chromosome 2 is shown. We see bands, genes, Affymetrix probe sets and QTLs for this part of the chromosome. On the right hand side we show a comparison of the part of the rat chromosome 2 with a part of the mouse chromosome 3. For the same part of the mouse chromosome 3 we display a comparison with the part of the human chromosome 1.

As a part of our co-operation with the BHF Cardiovascular Research Centre and the authors of [45], we prepared a local cache of all rat chromosome data (including information downloaded from [45]) to show in DerBrowser, see Figure 3.35. The chromosome, genes, Affymetrix probe sets and QTLs were shown in the tool for each of the rat chromosomes.

We also tried to show in DerBrowser a comparison for rat, mouse and human chromosomes, see Figure 3.36. During the work we observed some difficulties that were possibly due to unresolved bugs. While in SyntenyVista the moving of all chromosomes to show genes and similarities between them in both species are well implemented, in DerBrowser this is impossible. A user can only add data with changed coordinates (so that both chromosomes are shown side by side), and this is not enough for a precise visualisation of the comparisons. We also observed a problem with zooming in that has a limited resolution. Zooming does not suffice to show precisely small parts of genes or Affymetrix probe sets. To support this work, we prepared 2 posters ‘System level visualization of eQTLs (expression QTLs) and pQTLs’ (physiological QTLs) which were presented in the Bioinformatics and System Biology Conference in Edinburgh on 14th and 15th of July 2005 and Cardiovascular Functional Genomics Meeting in Glasgow on 4th July 2005. We prepared two versions of the data displays presented in DerBrowser in order to find out what ways of displaying the data are better and show the information more precisely. The results are available at <http://www.dcs.gla.ac.uk/~asia/>. The difference between the two presentations we prepared was in the presentation of QTL data, which was in different columns, divided into fat, kidney, eQTL or pQTL in the 1st version, and in one column in the 2nd version, but with the use of colour to distinguish between different QTL types. A user assessment was not carried out, as we discovered that DerBrowser does not zoom in enough to show the required details.

Der Browser - eQTLs & pQTLs examples

RAT chromosome 7.

See the whole chromosome map
or
limit to chromosome interval:

from: 114000000
to: 120000000

see the chromosome map

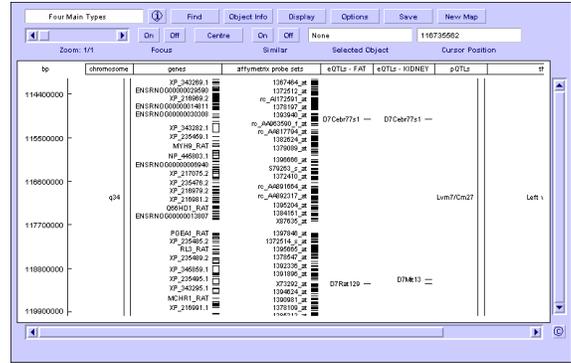


Figure 3.37: DerBrowser - choosing the coordinates for displaying only a specified part of the chromosome 7.

After consultation with researchers from the BHF Cardiovascular Research Centre, we decided to prepare also a version of DerBrowser combined with a CGI script, see Figure 3.37, which supports choosing the coordinates of the presented area. From the users' point of view the difference is that they can now select the coordinates of the part of the chromosome that they would like to see, and focus on an arbitrarily small fragment. The beginning and the end of the chosen chromosome part are set as defaults. In this solution we have two variables LEFTEND and RIGHTEND, and can draw the figure representing a chromosome fragment.

We also implemented a fisheye effect for DerBrowser, see Figure 3.38. First, we studied the fish-

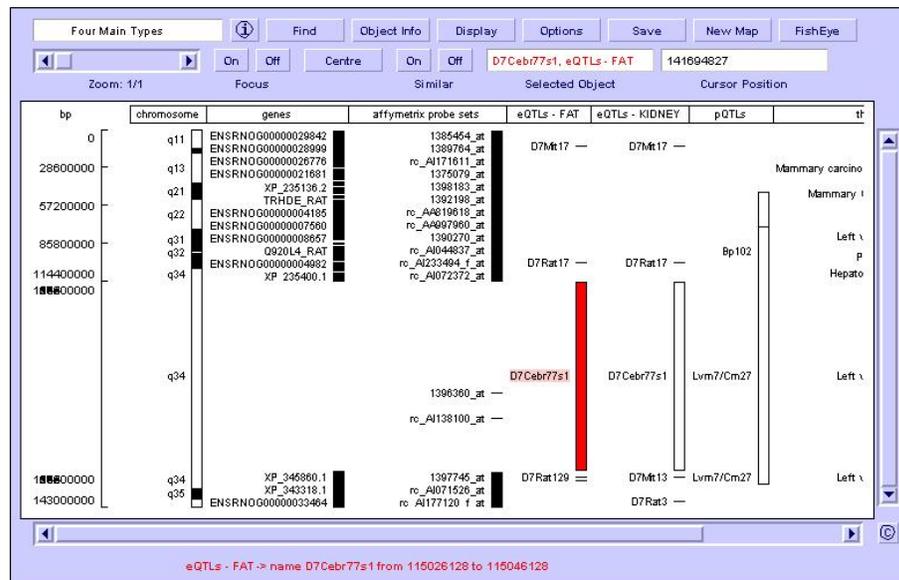


Figure 3.38: DerBrowser - fisheye effect.

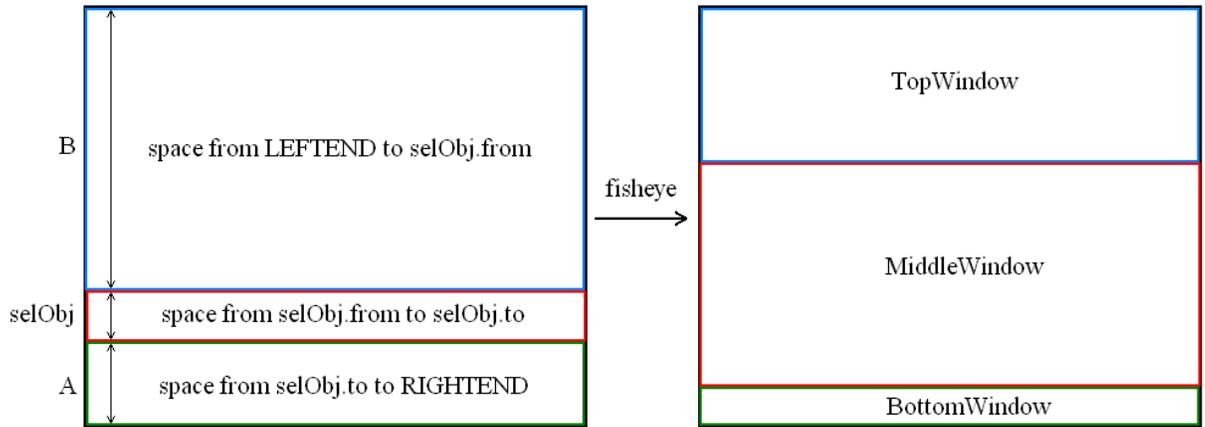


Figure 3.39: The three windows implemented in DerBrowser.

```

1 selObj = selected_element();
2 BigWindow(LEFTEND, RIGHTEND);
3 HalfWindow = BigWindow/2;
4
5 A = RIGHTEND - selObj.to;
6 B = selObj.from - LEFTEND;
7
8 if(A > B)
9     piece = HalfWindow / (A/B+1);
10    TopWindow = piece * HalfWindow;
11    BottomWindow = (A/B) * HalfWindow;
12 else
13    piece = HalfWindow / (B/A+1);
14    TopWindow = (B/A) * HalfWindow;
15    BottomWindow = piece * HalfWindow;
16
17 paint(selObj, MiddleWindow);
18 paint(B, TopWindow);
19 paint(A, BottomWindow);

```

Figure 3.40: An algorithm used during implementing fisheye effect for DerBrowser.

eye effect implemented in the Piccolo toolkit, and decided to implement a similar effect in the genome browser. It was more difficult to program in DerBrowser than in Piccolo because DerBrowser uses no Java Components or Containers. Therefore, the fisheye with semantic zooming effect was implemented from scratch, based on existing Java software. We divided the screen area into three windows where two of them change in size and the one in the middle changes in position. Conceptually, in the application there is a two level tree, where windows with zoomed elements are children of the window containing elements of the original size.

Before the work in DerBrowser, there was only one window responsible for drawing all data. We divided the screen into three windows, see Figure 3.39. The window in the middle (MiddleWindow) is always of the same size ($1/2$ of the big window) but it changes its position depending on the size of the two other windows (TopWindow and BottomWindow). When an element on the display is selected (mouse click), the size of both outside windows is computed as:

$A = \text{RIGHTEND} - \text{selObj.to}$

$B = \text{selObj.from-LEFTEND}$.

Then A is compared with B and the rest of the big window i.e. $1/2$ of the big window is divided into $(A/B+1)$ or $(B/A+1)$ pieces depending on whether A or B is larger. The smaller window receives 1 piece and the second one (B/A) or (A/B) pieces from the $1/2$ of the big window, see Figure 3.40.

We also implemented semantic zooming, with additional information appearing below the large window when an object is selected. This effect was very useful. The users can choose a small piece of data and see if there are any genes or Affymetrix probe sets in the area. It was impossible to see such data clearly in the previous version of DerBrowser. The source of DerBrowser is quite old and sometimes it is very tricky to find the piece of code responsible for some settings. There is a bug which should be corrected - the coordinates on the right hand side overlap.

3.5 Conclusion

Visualisation of genome comparisons is important to research in biology and medicine. We presented most of the available graphical tools used to represent biological data - genes and chromosomes in different species and recognised that none of them fulfill the biologists requirements. After the study of existing genome browsers, we decided to extend existing two browsers: DerBrowser and SyntenyVista. However, the browsers are using old technologies and it is not possible to show details for all relevant data our collaborators want to see in DerBrowser or SyntenyVista. Especially comparison between genes presented by DerBrowser does not show enough details required by the users. On the other hand, SyntenyVista does not allow the biologists to see their data. Therefore, we created VisGenome which is described in Chapter 5. According the requirements presented in Table 3.1, VisGenome ought to:

- present a number of different kinds of data, especially genes, markers, micro array probes, and QTLs which are very important for biologists work,
- present genome data for at least three species: mouse, human, and rat,
- provide an easy way for biologists to input their own data,
- show comparison between biologists' data and data available from other sources,
- provide visualisation techniques which allow the users easy navigate data,
- be accessible for all biologists.

In practice, all browsers should perform the same function, i.e. show the chromosomes of some species in detail. On the other hand, so many different visualisation tools have been developed to support the same user task and none of the existing tools shows all relevant data the biologist wishes to see. This

is due to the fact that a number of research groups or organisations are involved in genomics research, and all of them carry out their research and experiments independently. They all use the same types of data. All of the projects use a tool which focuses on a special part or aspect of the data, or the data annotation and interpretation process which is the most important for them. Each of the browsers has useful features (such as zooming, panning, or presenting relevant information) which we tried to adopt during the development of VisGenome. On the other hand, each of them has drawbacks (such as placing all information in one view or using pop up menus which do not disappear without clicking) which should be avoided.

This chapter presented a survey of genome browsers while focussing on visualisation techniques. We classified all genome browsers according to three dimensions. We also present our early work with existing genome browsers - DerBrowser and SyntenyVista, which are not enough to show all relevant data for biologists. The following chapter presents background from the field of human-computer interaction. As can be seen from Table 3.2, user studies are not popular in genome browsers development probably because the majority of the tools are created by biologists themselves for a special kind of data and the developers do not envisage that their application could be potentially used by other medical researchers. However, user studies accompanying the creation of genome browsers would help to improve the design. This could generate new design ideas and a better understanding of the real use of genome browsers, and save time needed for data analysis.

This chapter provided the context for genome browsing. In the following chapter we turn our attention to HCI techniques which provide a background to the user studies we conducted.

Chapter 4

HCI - Design and Evaluation

This chapter briefly introduces the main points from the Human-Computer Interaction (HCI) area and presents our users who took part in the two user studies detailed in Chapters 6 and 8 ('Initial Quantitative User Study' and 'Mixed Paradigm User Study'). In the next chapter we introduce VisGenome which was evaluated according to the HCI theory presented here and motivated by our genome browser survey presented in the previous chapter.

4.1 Introduction

"Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." The definition¹ presented in [41] (p.6.) clearly states that HCI study describes interaction between people and computers, which occurs at the user interface. We used in the thesis title **Biologist-Computer Interaction** to show the focus on that particular user community.

HCI is a multi-disciplinary subject, see Figure 4.1, and it is related to ergonomics and human factors, engineering, design, anthropology, sociology, philosophy, linguistics, artificial intelligence, computer science, cognitive psychology, and social and organizational psychology. However, HCI is especially important in computer science and systems design, because it involves the design, implementation and evaluation of interactive computer systems in the context of the users' task and work [23].

The main goal of HCI is to produce usable and safe systems, as well as functional systems. To

¹It is worth mentioning, that this was not the first definition of HCI. The term human-computer interaction was adopted in the mid-1980s and it acknowledged that HCI was concerned with all aspects which relate to the interaction between users and computers. One of the first HCI definitions was presented in [2] (p.40.) and it describes HCI as "[a] set of processes, dialogues, and actions through which a human user employs and interacts with a computer".

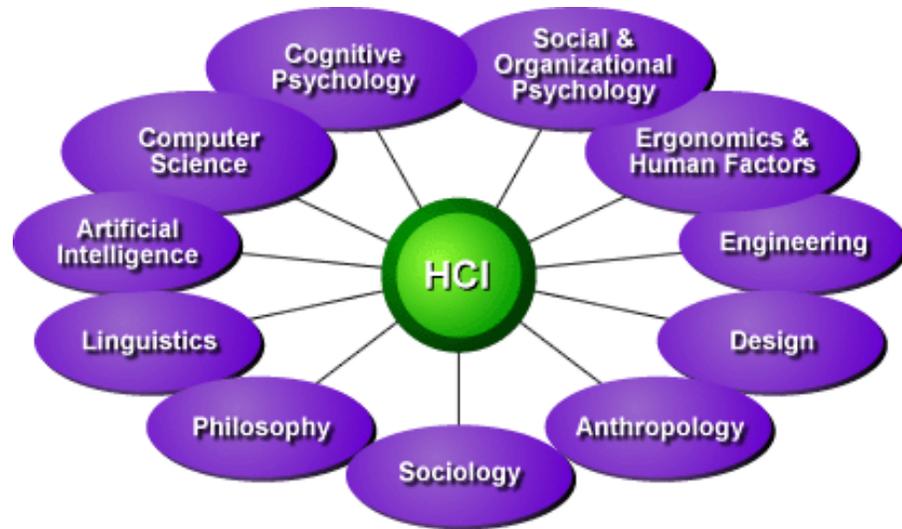


Figure 4.1: Disciplines related to HCI: ergonomics and human factors, engineering, design, anthropology, sociology, philosophy, linguistic, artificial intelligence, computer science, cognitive psychology, and social and organizational psychology. The figure is taken from [133].

produce computer systems with good usability, developers ought to:

- understand the factors that determine how people use technology
- develop tools and techniques to enable the building of suitable systems
- achieve efficient, effective, and safe interaction
- put people first

As present the definition of a “user interface” plays a very important role in HCI. Dix and colleagues [23] present common interface styles:

- command line interface
- menus
- natural language
- question/answer and query dialog
- form-fills and spreadsheets
- WIMP² (windowing system)

²WIMP is the style of graphical user interface that uses Windows, Icons, Menus and Pointers - common windowing widgets. It was invented at Xerox PARC, popularised by the Apple Macintosh and now available in other varieties such as the Microsoft Windows operating system, the X Window System, OSF/Motif, NeWS and RISC OS [137].

- point and click
- three-dimensional interfaces

The presented interface styles allow for communication between the users and computers. Each of the styles corresponds to interface types. We distinguish user interface types such as graphical, voice, multi-modal³, and other. During our study presented in next chapters we used a graphical user interface implemented in VisGenome. It is presented by a windowing system and the users can select items of interest mainly by clicking and dragging.

4.2 Design

Dix and colleagues [23] propose a simple definition of design: “achieving goals with constraints”. First, the designer has to decide what the purpose of the design is and who it is for. In our situation it was a user-friendly genome browser (VisGenome) which could be used by biologists and medical researchers. The tool aimed to allow them to see their experimental data and compare those with the results from other sources (biological experiments). Second, the designer should know constraints - what materials he could use, how much it costs, or how much time he is going to spend on it. The authors [23] stress that not all goals are always achieved within the constraints. Therefore, “trade-off” plays also an important role in designing - “choosing which goals or constraints can be relaxed so that others can be met”.

Dix and colleagues present “understanding materials” as “the golden rule for design”. In HCI it is understanding humans and computers, and how interfaces affect the users. Cognition⁴ plays an important role during these processes. As can be seen in the following chapters, during our second user study some biologists forgot about the options offered by VisGenome. Because of this, they did not use some from the functions offered by the tool.

The process of interaction design consists of four basic software development phases (identifying needs and establishing requirements, developing alternative designs, building interactive versions of the designs, and evaluating designs), [89]. The activities could be related to one another in different ways. Depending on how they are related, we distinguish different “lifecycle models”. The best known life cycle models are:

³Multi-modal interfaces attempt to address the problems associated with purely auditory and purely visual interfaces by providing a more immersive environment for human-computer interaction. A multi-modal interactive system is one that relies on the use of multiple human communication channels to manipulate the computer. These communication channels translate to a computer’s input and output devices. A genuine multi-modal system relies on simultaneous use of multiple communication channels for both input and output, which more closely resembles the way in which humans process information, [23].

⁴Cognition is what goes in our heads when we carry out our everyday activities. It involves cognitive processes, like thinking, remembering, learning, daydreaming, decision making, seeking, reading, writing, and talking, [89].

must be completed before the next one begins. The V-shaped model stresses testing more than the waterfall model. Testing procedures are developed before any coding is done, during each of the activities preceding implementation.

- incremental model - The incremental model is an intuitive approach to the waterfall model. It has multiple development cycles, which making the model a “multi-waterfall” cycle. Cycles are divided up into smaller, more easily managed iterations. Each iteration passes through the requirements, design, implementation and testing phases.
- linear model - The linear design model presents different design strategies that are done once, in a fixed order.

During our research we used the spiral model, see Figure 4.2. The “lifecycle model” has a number of advantages such as:

- high amount of risk analysis
- good for large and mission-critical projects
- software is produced early in the software life cycle

On the other hand, it also has weaknesses, such as:

- can be costly to use
- risk analysis requires highly specific expertise
- project success is highly dependent on the risk analysis phase
- does not work well for smaller projects

The spiral model focuses on prototyping. The developer builds a simplified version of the proposed system and presents it to the user for consultation as part of the development process. As can be seen in the next chapters, it exactly corresponds to our research work. We developed a prototype of VisGenome and consulted with biologists to see what should be changed in the next prototype version. We received feedback and went back to the system requirements. We wanted to create a genome browser for medical researchers according to their requirements, and this type of “lifecycle model” was the most appropriate in our situation.

4.3 Evaluation

We can briefly define evaluation such as “collecting data about the usability of a design or product by a specified group of users for a particular activity within a specified environment or work context” [23]. A

number of methods could be used for evaluation, however, in each of them it is important to consider:

- the characteristics of the users
- the types of activities that the users will do
- the environment of the study, which may range from a controlled laboratory situation to a natural work setting
- the nature of the artefact being evaluated, for example a paper prototype, a working software prototype or the finished product

Monk and colleagues [74] stress that in late 1980s and early 1990s there was a trend to move from evaluator-controlled forms of evaluation to more informal techniques, some of which are derived from anthropology and sociology. They listed the following evaluation forms: interpretative evaluation⁵, contextual inquiry⁶, cooperative and participative evaluation⁷, predictive evaluation⁸, and usage simulations⁹. Molich and Nielsen [73] initialised a method known as heuristic evaluation. According to the authors, this kind of evaluation is “cheap”. Such cost effective methods could be used by small companies who could not afford or did not have the facilities, time or expertise necessary to do usability engineering. In heuristic evaluation reviewers examine the system or prototype as in a general review or usage simulation, but their inspection is guided by a set of high-level heuristics [80] which guide them to focus on key usability issues. Nielsen [79, 87] also developed discount usability evaluation. The idea behind this technique is to enable developers with few resources - in terms of time, money or expertise - to benefit from usability testing during product design and development. A number of researchers use a walkthrough in HCI design [74], to detect problems very early on so that they may be removed. Walkthroughs involve constructing carefully defined tasks from a system specification or screen mock-ups¹⁰.

A number of HCI researchers use also (mentioned before) ethnography as a method of collecting data about a real work situation. They stress that the scientific hypothesis-testing has a number of

⁵Preece and colleagues [88] summarised the interpretative evaluation as “spending time with users”. They stress that lab conditions are not real world conditions, and only by observing the users in a natural environment can the developer detect the presence of these factors. Preece and colleagues classified the ethnographic investigation as interpretative evaluation.

⁶The users and the researchers work together to understand and identify usability problems. They work in the users’ natural environment.

⁷“Co-operative evaluation is a technique to improve a user interface specification by detecting the possible usability problems in an early prototype or partial simulation. It sets down procedures by which a designer can work with the sort of people who will ultimately use the software in their daily work, so that together they can identify potential problems and their solutions.”[74]. Participative evaluation is more open than cooperative evaluation and subject to greater control by users.

⁸The developers try to predict problems before user testing.

⁹Usage simulations involve reviewing the application to find usability problems. These reviews are usually done by experts who simulate the behaviour of less experienced users and try to anticipate the usability problems that they will encounter.

¹⁰A mock-up is a scale model of a structure or device, usually used for teaching, demonstration, or testing a design.

limitations [74]. In ethnography, different kinds of data could be collected by video recording, annotations in notebooks, snapshots, etc. Preece and colleagues [88] define also ethnomethodology as a method that analyses behaviour by observing events in their natural environment. Ethnomethodology refers to the analysis of commonsense methods that the users exploit while carrying out everyday actions.

During both our studies we tried to use all available evaluation methods. We cooperated with biologists from the BHF Cardiovascular Research Group from the Western Infirmary (Glasgow) and on a number of occasions we observed their work either in the laboratory or in the office (**direct observation**). We also used video and voice recording (**indirect observation**) and log file **monitoring** during our second user study. During the observation our users knew that they were observed and this affected the way they performed, especially during indirect observation. We also **collected medical researchers' opinions** during interviews. In a **structured interview** we asked them specific questions in a given order, and in a **flexible interview** they were free to express their opinions about visualisation techniques and genome browser. We also used **questionnaires** both **closed** and **open questions**. We conducted two **experiments**, where during the initial quantitative user study we applied **benchmarking**, to compare the results achieved by the developer and by the users. We knew that visualisation techniques are required by biological researchers in their work, and are used in genome browsers which summarise experimental data. During our second experiment we used **interpretive evaluation** which enables us to understand better how the biologists use genome browsers in their natural environments, and how their use of VisGenome integrated with other activities that they performed. We also used **predictive evaluation** where we tried to predict problems that biologists may encounter before testing VisGenome with users. We used this method before creating the first application prototype. In particular, we used **feature inspection** - we thought about interaction techniques which could be used for typical tasks carried out by biologists.

We used a quantitative user study during our first experiment because we wanted to compare two tools: VisGenome and Ensembl, and this kind of evaluation allows us to collect data which was used during statistical calculations. According to suggestions from specialists in the HCI field we decided that our second user study was conducted in a biologists' workplace, which allowed us to observe the participants in their natural environment. We used different techniques mainly because we wanted to find an answer to the question: which visualisation techniques are useful for biologists in their work? We recognised that the same question asked during different evaluations could give different answers. Therefore, we decided to gather data via different kinds of observation, monitoring, interviews, questionnaires, and video and voice recordings. We used recording to gather data for future detailed analysis. We planned to combine the results from the different approaches, however, it was not always possible. For example, during mixed paradigm user study we used video and voice recording, monitoring and observations. We were not able to record biologists during their work with animals, and what quite frequently happened was that we collected data from video recording, observations and monitoring, but the data was not compatible and could not be combined. We observed work with animals, monitored VisGenome usage, and recorded work

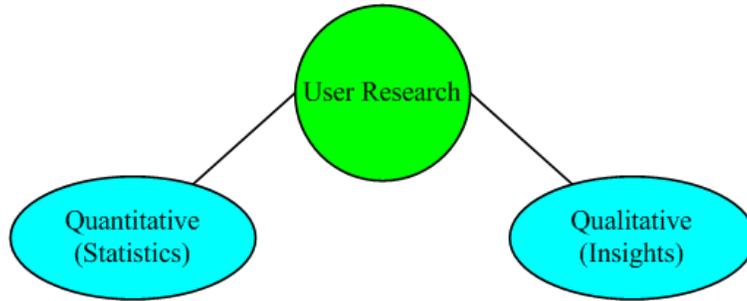


Figure 4.3: User research types.

in laboratories. We learned that each kind of evaluation could be useful and give positive results when used properly. We used interpretive valuation to better understand biologists’ work, predictive evaluation to predict problems that may be encountered before VisGenome testing, and feature inspection to see which techniques could be used by the users during their work.

We can divide usability research into two main types: quantitative¹¹ and qualitative¹² [81], see Figure 4.3. We used statistics during initial quantitative user study. We also used insight-based observational research during our second user study.

On the other hand, Christensen [21] begins his discussion of experimental methodology from causation¹³

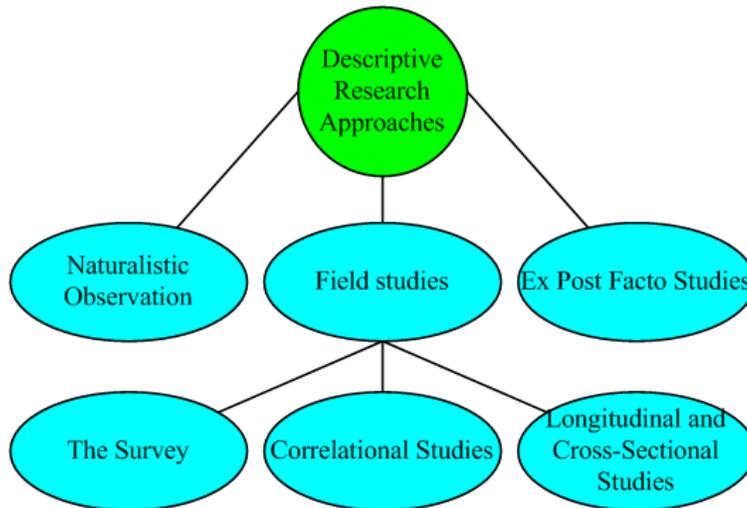


Figure 4.4: Approaches to descriptive research [21].

¹¹Quantitative approaches are strictly concerned with numerical measurements (quantities). They are typical of the mainstream scientific approach in psychology. Quantitative approaches aim to test hypotheses, and usually identify numerical differences between groups.

¹²Qualitative approaches refer to how people understand their experiences (qualities).

¹³John Stuart Mill [72] found four canons (the methods of agreement, difference, concomitant variation, and the joint methods of agreement in difference) which could be useful during identifying causation. The author presented the methods

which should be identified during the experimental methods; he also presents descriptive approaches, see Figure 4.4. Christensen stresses that descriptive approaches (naturalistic observation¹⁴, field studies¹⁵, and ex post facto studies¹⁶) are different from experimental approaches. Descriptive approaches have to describe or “paint a picture” of a particular phenomenon, whereas the experimental approach attempts to identify cause-and-effect relationships.

An experiment is an “objective observation of phenomena which are made to occur in a strictly controlled situation in which one or more factors are varied and the others are kept constant” [120]. It means that a tester should avoid taking steps what could influence the outcome of the experiment. An experimental research approach gives the tester a number of benefits such as:

- control
- ability to precisely manipulate one or more variables of the experimenter’s choosing
- use of the experimental approach can produced results that are valid over time
- has suggested new studies and solutions to practical problems

The main disadvantage of an experimental approach according to Zimney [120] (the psychological experiment) is that laboratory findings are obtained in an artificial environment which “precludes any generalization to a real life situation”. However, the experimental approach is used in both laboratory and field environments. The experiment presented in Chapter 8 (‘Mixed Paradigm User Study’) was conducted in the biologists’ labs - a natural settings where they engage in daily activities. Field experiments such as ours do not have to worry about the artificiality problem that arises in with laboratory experiments - our by examples. The method of agreement is illustrated by an example of man who wanted to find out why he got drunk. One day he drank rye and water, second day he drank scotch and water, and third day he drank bourbon and water, and always he was getting drunk. He decided that the water was the cause of his getting drunk because it was an element he drank each time. The method of difference is used during psychological experiments. Drugs and a placebo are given to the subjects and the reaction time is measured. The joint method of agreement and difference, and the method of concomitant variation qualify that a variable is either a cause or an effect or “is connected through some factor of causation in variation in the variable results in a parallel variation in another variable” [21]. According to Christensen [21], finding the cause of an effect requires discovering the necessary and sufficient conditions for the occurrence of an event.

¹⁴Naturalistic observation enables the investigator to collect data on naturally occurring behaviour. As an example we present Ebbesen and Haney [27] study. They used observer instead taking part in their experiment. The observer kept distance in order to record natural users’ behaviour. If the users knew they were being observed, their behaviour could not be the same. They are not behaving in an environment removed from real life.

¹⁵Field studies are conducted in the “real world” but the developer intervenes in the data collection. The observations of interest are focused on a more specific aspect of behaviour than are those in a naturalistic observation study. “Field studies use a variety of diverse approaches whereas naturalistic observation uses one general procedure.” ([21]) Sampling error, time constraints, and constraints in the length of the survey are the disadvantages of field studies.

¹⁶Ex post facto studies are studies in which the variables of interest to the developer are not subject to direct manipulation but have to be chosen “after the fact”.

initial quantitative user study is a laboratory experiment. The difficulty with field experiments is that variables cannot be controlled as well as in a laboratory experiments [21].

The next sections present examples on how evaluations and user studies are carried out in visualisation and bioinformatics.

4.3.1 User Studies in Visualisation

User studies are very popular in the field of visualisation. Kosara and colleagues [57] present four main reasons why user studies should be conducted in this area. The reasons are:

- evaluation of the strengths and weaknesses of visualisation techniques,
- showing that a new visualisation technique is useful in a practical sense,
- showing why a particular technique is effective,
- showing that an abstract theory applies under certain practical conditions.

We conducted two user studies for the following reasons. We wanted to show advantages and disadvantages of visualisation techniques used in genome browsers. We developed a new scaling algorithm and wanted to see how the medical researchers use it in practice and if it is effective. We hypothesised that the biologists required a good implementation of basic interaction techniques such as zooming or panning, which could help them navigate their data.

Tory and Moller [110] find that formal user studies are essential but they could be inappropriate when “clear objectives and variables may not be defined”. They focus on the superiority of qualitative with regard to quantitative user study. The authors suggest it is better to ask experts about their opinion of how the interface could be used than “a few friends” whose opinion could be not sufficient. Preece and colleagues [89] also stress that “five usability experts typically find 75% of a system’s usability problems” and up to 50 users in laboratory user study could have the same results.

Ellis and Dix [29] propose explorative evaluation as a method of studying visualisation techniques. They find that a number of researchers report in their papers on future work “undertaken through user evaluation”. On the other hand, 60% of researchers do not think that evaluation is worth mentioning. Ellis and Dix present their personal opinions on 65 papers they selected. They focus on difficulties which could occur during evaluation, such as finding suitable datasets and users. They find a study successful if “they effectively demonstrate the potential benefits of some application through user evaluation”. As a “good” user study the authors single out LifeLines [85]. A number of “real” users from a wide range of areas took part in the study. On the other hand, they all had good knowledge about the presented domain data. The users were asked to use the tested application with typical data and could comment,

and discuss advantages and problems. Ellis and Dix present also a “bad” sample of experiments. They stress that Fekete and Plaisant [30] ignored the time to zoom in the map in their studies where they asked the users to find information on a map. The participants could use popup labels or zoom in to read the labels. They revealed that zooming was often disorientating, and the developers could offer them other techniques to show small text. Moreover, activating popup labels required more precision than zooming manipulation.

Alphaslider [1], discussed in Chapter 2, was also tested by users in a controlled environment. Ahlberg and Shneiderman present four different alphaslider interfaces and ask the users to locate an item in a list of 10,000 movie titles. They observed that the users used different buttons depending on the application version. The authors found that the alphaslider allows the users to rapidly select items without a keyboard, using minimal screen space.

Graham and colleagues [35] presented a case study of a visualisation which represents relationships between multiple hierarchical structures. They reviewed different HCI concepts, and developed a system to support taxonomists who classify organisms and generate a classification hierarchy depicting relationships between the phenomena. Their work belongs in two fields: visualisation and bioinformatics. Graham and colleagues adapted known HCI methods, and iteratively tried and selected two most appropriate approaches. They wanted to find visualisation techniques that show multiple hierarchies and facilitate the exploration of relationships between different classifications. The authors developed a methodology which builds a number of prototypes in an interactive cycle of design and testing. They divide the work into 6 phases. In the 1st phase (initial requirements) they classified the data, which helped them to extract the visualisation requirements, and also gleaned some clues as to how to construct the visualisation interface. Graham and colleagues defined a number of tasks that a proposed visualisation should be able carry out or support. In the 2nd phase they sketched a number of diagrams that visually demonstrate the activities that the taxonomists are concerned with. The 3rd phase included the first user test. Representative users tried out the prototypes using representative tasks, and conducted a usability test to check if the prototype visualisations could easily communicate the outcomes of the functions on an example data set. The authors were looking to decide which prototype to proceed with, based on a pair of visualisations using different metaphors. The users were also asked questions on how they felt about each prototype. In the 4th phase the user test was rerun to confirm the proper functionality and to find usability issues. The users’ actions were recorded during this experiment. The 5th phase was dedicated to usability issues. This validated the removal of the major usability flaws found in the second test. The final 6th phase used a more statistically rigorous approach. Graham and colleagues presented a multi-step design approach for deriving and testing a new IV-oriented interface. At each stage appropriate techniques were used to test different aspects of the interface.

4.3.2 User Studies in Bioinformatics

A number of user studies were conducted in visualisation, while only a small number of user studies have been published in bioinformatics.

Stevens and colleagues [108] surveyed bioinformatics tasks undertaken by biologists. They investigated the biological nature and syntactic structure of queries and tasks. The authors used a questionnaire to assess the bioinformatics knowledge and tool usage in the community. They report on a number of new requirements which could stimulate the development of future bioinformatics applications. The paper does not involve a user study and focuses on task classification.

Wu and colleagues [118] report on an electronic table that uses fisheye distortion. The table shows gene expression data and is a subject of a pilot user study. Five researchers took part and a Questionnaire for User Interface Satisfaction (QUIS) [104] was used. User impressions were positive.

Yang and colleagues [119] studied biologists interacting with MetNet3D. Both students and researchers used the software to analyse experimental data, however, a formal study has not taken place. They used a six-wall surround-screen projection viewed through stereo glasses, and a six degree-of-freedom head tracker and joystick. The authors reported that selection of an object was easy for users, and visualising pathways gave them more realistic feeling and more natural interaction. However, they did not present any user studies supporting their conclusions.



Figure 4.5: A photo of a medical researcher from the BHF Glasgow Cardiovascular Research Centre.

4.4 Our Users

During our user studies we consulted medical researchers from the BHF Glasgow Cardiovascular Research Centre, see Figure 4.5. The biologists from the BRC (Bioinformatics Research Centre at the Department of Computing Science at University of Glasgow) also helped during our initial quantitative user study. They worked or still are working with a genome browser - from the developer point of view. They did not take part in our second mixed paradigm user study. Therefore, in this section, we focus on medical researchers who took part in our research work.

The BHF Glasgow Cardiovascular Research Centre is led by Professor Anna Dominiczak. The group investigates hypertension¹⁷. Essential hypertension is a disease influenced by genetics, environment, and the interaction of genes and the environment [141]. The BHF Glasgow Cardiovascular Research Centre researchers carry out a range of activities and use a number of biological techniques [24]. We divide their activities into three areas: studies of humans, work with animals, and office activities. In all areas they collect data which should be visualised and compared with data from other sources. They also use a number of computer applications which should allow them to do their work efficiently.

4.5 Medical Researchers' Activities

This section presents the work of medical researchers from the BHF Glasgow Cardiovascular Research Centre, which we observed during this PhD.

There are about 50 people who are divided into small groups focussing on projects. They all meet once a week on Thursdays and exchange their discoveries and work problems in a few words. They present graphs or figures, and one person reviews a new biological paper - every week a different person. Therefore, everybody knows, more or less, what people in the group are doing and what people from the same area in other centres around the world have discovered and published. Moreover, every Thursday there is also an one hour meeting centred on a grant or a project, led by the group leader. Frequently, people from the same group share the same office, so they can continuously exchange their knowledge. Of course, in small groups supervisors coordinate the research. Depending on qualifications - biologists or medical doctors - they carry out activities with people or rats. Everybody has a work space in a laboratory and a desk in an office. Office space is very limited.

¹⁷Hypertension, most commonly referred to as “high blood pressure”, HTN or HPN, is a medical condition in which the blood pressure is chronically elevated [140].

4.5.1 Human Studies

Some of the medical researchers work as doctors. They take care of patients with hypertension and heart diseases. Very often they study a few generations of families to identify genes associated with hypertension. They work as doctors and as a physiotherapists, i.e. some of them treat patients in the hospital and some try to design exercises which could them help. They also develop new forms of gene therapy [90, 117] to help people affected by disease. The medical researchers collect a lot of blood samples from their patients, which are investigated and compared with blood samples from rats.

4.5.2 Animal Work

The researchers study rats to identify hypertension genes. It is not easy to identify genes responsible for hypertension. It is possible to localise genes influenced by hypertension, but this is not the same as identifying the genes that cause it. Therefore, the researchers use rats which provide models for human hypertension.

Their work involves:

- rat feeding
- radio-telemetry probe preparation (disinfection, testing, telemetry surgery)
- blood pressure measurements
- using computer tools for visualising blood pressure from the measurements and the radio-telemetry.

The biologists from the Centre feed the animals with special high calorie food. It allows them to simulate overfeeding in a short time. Each animal is under one researcher's care. Therefore, it is not possible to share animals. Each rat has its unique tag which allows the biologist to identify it quickly.

To acquire accurate blood pressure readings, the researchers use radio-telemetry probes, see Figure 4.6. They implant the equipment into an animal and observe it for about 3-4 weeks. The probes measure rat activity and blood pressure in the aorta. A radio-telemetry probe could be used a few times, but it has to be sealed and disinfected, and its battery has to be recharged.

An animal with a radio-telemetry probe is monitored and observed all the time. The biologists use for this the DSI Acquisition tool, see Figure 4.7. The application collects a number of data points, and supports zooming and graph drawing. However, the biologists do not analyse the data in detail. They use it only for drawing and analysing significant changes during the animal life.

Every two days the researchers take the animal blood pressure. They use special equipment, see Figure 4.8, which is coordinated by an application collecting the data.

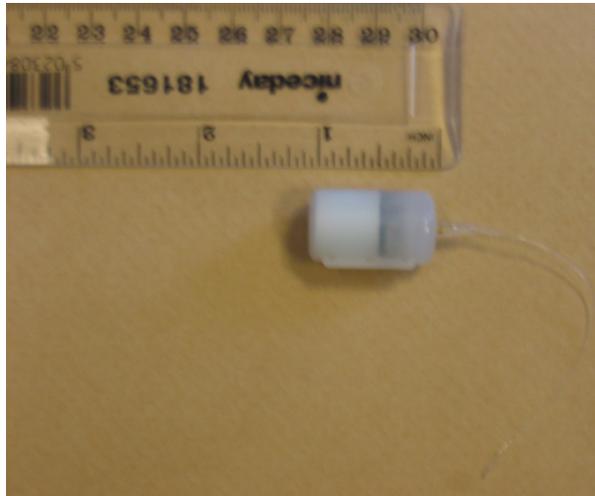


Figure 4.6: A radio-telemetry probe used for measuring blood pressure and activity levels in rats.

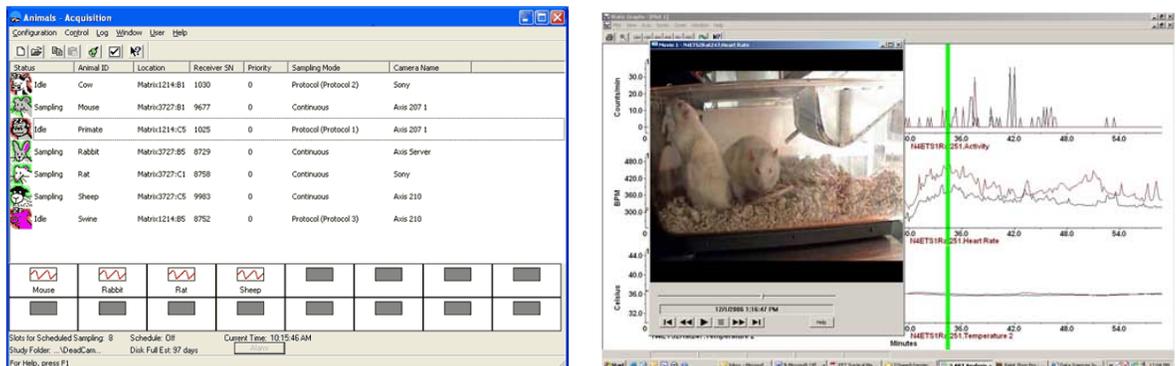


Figure 4.7: Data Sciences International (DSI) Acquisition [127] system for monitoring animals with radio-telemetry probes.

The researchers also conduct animal breeding, see Figure 4.9. Breeding allows them to find a gene which causes disease, by eliminating the contribution of other genes (minimising the size of the area under investigation).

Another very important activity allowing the researchers from the Centre to find genes responsible for hypertension are micro array experiments (analysing the activity of genes, including faulty genes), see Figure 4.10. GeneChips, like the one presented in Figure 4.10, are widely used and are very useful during the biological experiments [37]. However, new modern equipment, called Illumina, see Figure 4.11, is being introduced. An Illumina Gene Expression Chip contains more micro arrays than the Gene Chip used by a photo lithography machine (Illumina's bead-array technology provides competition to Affymetrix's 500K gene chip) and needs less genetic material for producing results.



Figure 4.8: Blood pressure measured for rats.

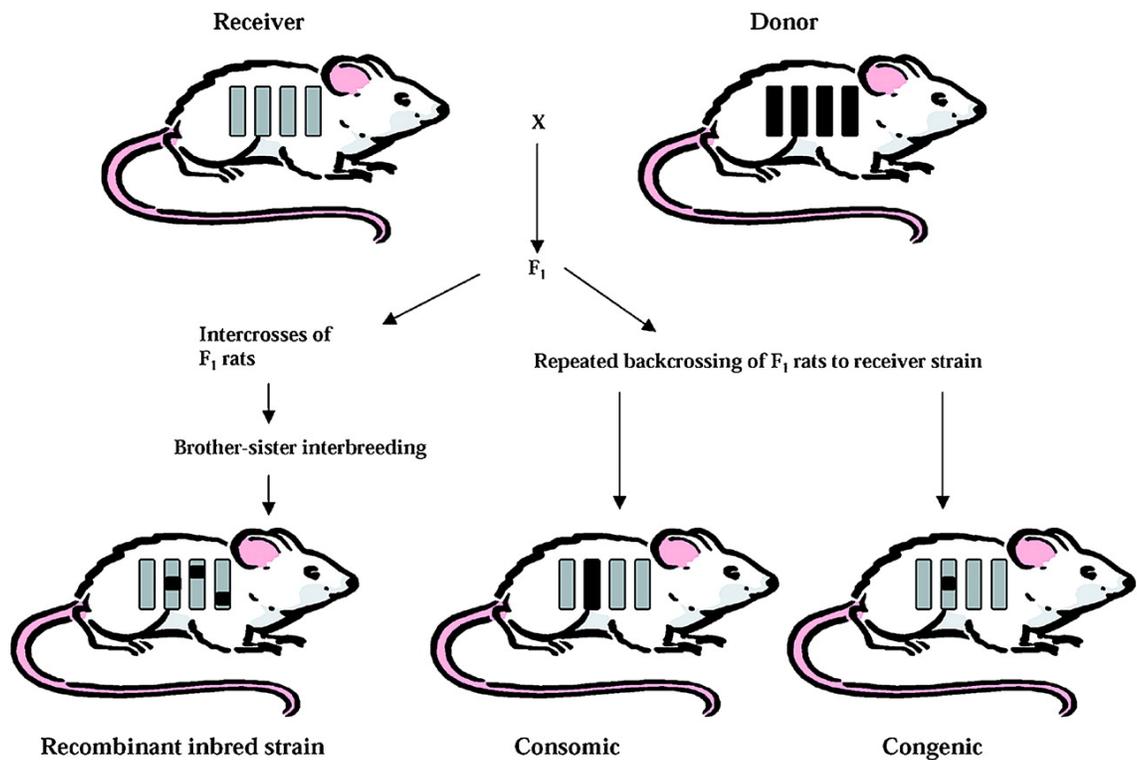


Figure 4.9: Breeding recombination, taken from [66]. It clearly presents that a number of studies is expected to find genes responsible for hypertension. Where an animal in the next generation has a disease and receives from a donor only a part of one chromosome, all genes in the chromosome could be responsible for the disease.

When the biological researchers obtain all available data from human and animal blood samples, they use comparative genomics to study relationships between the genomes of different species. They compare all genomics data at different levels of detail and they need a user-friendly genome browser to support their work.

4.5.3 Office Work

Office activities include work with students and other collaborators, meetings and conferences, work with computer applications, learning by reading papers and writing articles for conferences. The researchers spend a few hours per week sitting in their offices, looking for the newest published papers, and using computer applications. They mainly use Microsoft Office tools such as Word, Excel, Access, Power Point, Outlook, and Graphpad Prism. They use the tools to write papers, collect their experimental data, or make presentations and graphs for meetings. Some applications, such as DSI Acquisition, or rat tail blood pressure determination, are not used in the offices but in the laboratories where the animals are held, because the tools are interfaced with the equipment used for animal breeding. Other tools, such as Ingenuity Pathway Analysis (IPA), are used for searching and exploring experimental data. The tool allows them to identify the pathways, molecular mechanisms and biological processes in their experimental data, based on micro array results.

4.6 Conclusion

Nowadays not only specialists use computers but people in all areas. Therefore, it is important that they can interact with the machines in a useful and beneficial way. The HCI challenge is to design computer systems that users can gain maximum benefit from.

The chapter presented background information from the human-computer interaction field. We introduced design and evaluation methods with examples from our user research work. Examples of literature from system evaluation in bioinformatics and in visualisation were also briefly presented. In the second part of this chapter, we present users and their activities which allow them to collect medical and genomics data. We introduced our users and what they do in their work. We also showed why the medical researchers collect the genomics data and what they want to find in it. From the presented work, it is clear that the researchers require user-friendly tools to compare their results with other known findings. In the next chapter, we introduce VisGenome, which was developed according to the medical researchers' suggestions and the feedback from our user studies. The following chapters illustrate how we used the theory presented in this chapter and in the chapters about 'Visualisation Background' and 'Genome Browsers' (Chapters 6 and 8) in practice.

Chapter 5

VisGenome

This chapter presents the first version of VisGenome which was used during our initial quantitative user study (presented in the next chapter). VisGenome together with the CartoonPlus algorithm presented in Chapter 7 are our main contributions, and are still used by biologists from Western Infirmary. We briefly describe the features offered by the application and present Java Web Start and stand alone versions of VisGenome. The work presented in this chapter was published as an application note in Bioinformatics [51]. More information about how to use VisGenome is available in our user manual (see Appendix G) at the web page <http://www.dcs.gla.ac.uk/~asia/VisGenome/>, which also offers the application source code.

5.1 Introduction

VisGenome visualises single and comparative representations for the rat, the mouse, and the human chromosomes at different levels of detail. The tool offers smooth zooming and panning which is more flexible than that seen in other browsers. It presents information available in Ensembl for single chromosomes, as well as homologies (orthologue predictions including `ortholog one2one`, `apparent ortholog one2one`, `ortholog many2many`) for any two chromosomes from different species. The application can query supporting data from Ensembl by invoking a link in a browser.

According to the requirements presented in Chapter 3, we designed VisGenome to:

- present a number of different kinds of data, especially genes, markers, micro array probes, and QTLs which are very important to biologists' work,
- present genome data for at least three species: mouse, human, and rat,
- provide an easy way for biologists to input their own data,

- show comparison between biologists' data and data available from other sources,
- provide visualisation techniques which allow the users to easily navigate data,
- be accessible to all biologists.

VisGenome was designed to match the visualisation needs of the BHF Cardiovascular Research Centre at the University of Glasgow, which uses a rat model of cardiovascular disease. The user can analyse genomic data at different levels of detail, to dissect rat and human QTLs [67] in the search for candidate disease genes. QTLs are shown in three species: the rat, the human and the mouse, and the work uses genotyping, and micro array and proteomics techniques. VisGenome supports QTL analysis by showing QTLs and the genes within each QTL in two species in one display, along with the supporting experimental data. It also shows data for a single chromosome in one display and supports zooming at an arbitrary level of detail. The first version of the software release connects to Ensembl and can be used as an alternative viewer for a subset of that data. As can be seen, we used all the requirements without providing an easy way for biologists to input their data. We decided to apply this requirement later, and for first prototype of VisGenome we downloaded data instead of leaving this task to biologists. We designed VisGenome mainly because our studies showed that existing genome browsers (see Chapter 3) do not fulfil our collaborators' requirements. However, not only did existing genome browsers influence our application, so did the existing visualisation techniques which made it easy to navigate (zooming and panning, see Chapter 2) and the evaluation techniques (see Chapter 4) which allow us to check what our users expect and how they interact with the new tool.

5.2 Features

VisGenome loads QTLs, genes, micro array probes, bands, and markers, and displays pairs of homologies from Ensembl. It welcomes the user with a view of all rat, mouse and human chromosomes, see Figure 5.1. In the single chromosome representation, after choosing a chromosome of interest by clicking, a new view with detailed data about the chromosome is created. In the comparative representation the user clicks on two chromosomes from different species and a new view representing homologies between the chosen chromosomes is created. After choosing a chromosome, the users can manipulate the view by mouse and keyboard interaction.

5.2.1 Navigation

VisGenome offers “overview and detail” views which are manipulated by mouse and keyboard interaction. At the beginning the users see an overview of all chromosomes and can choose the one they would like

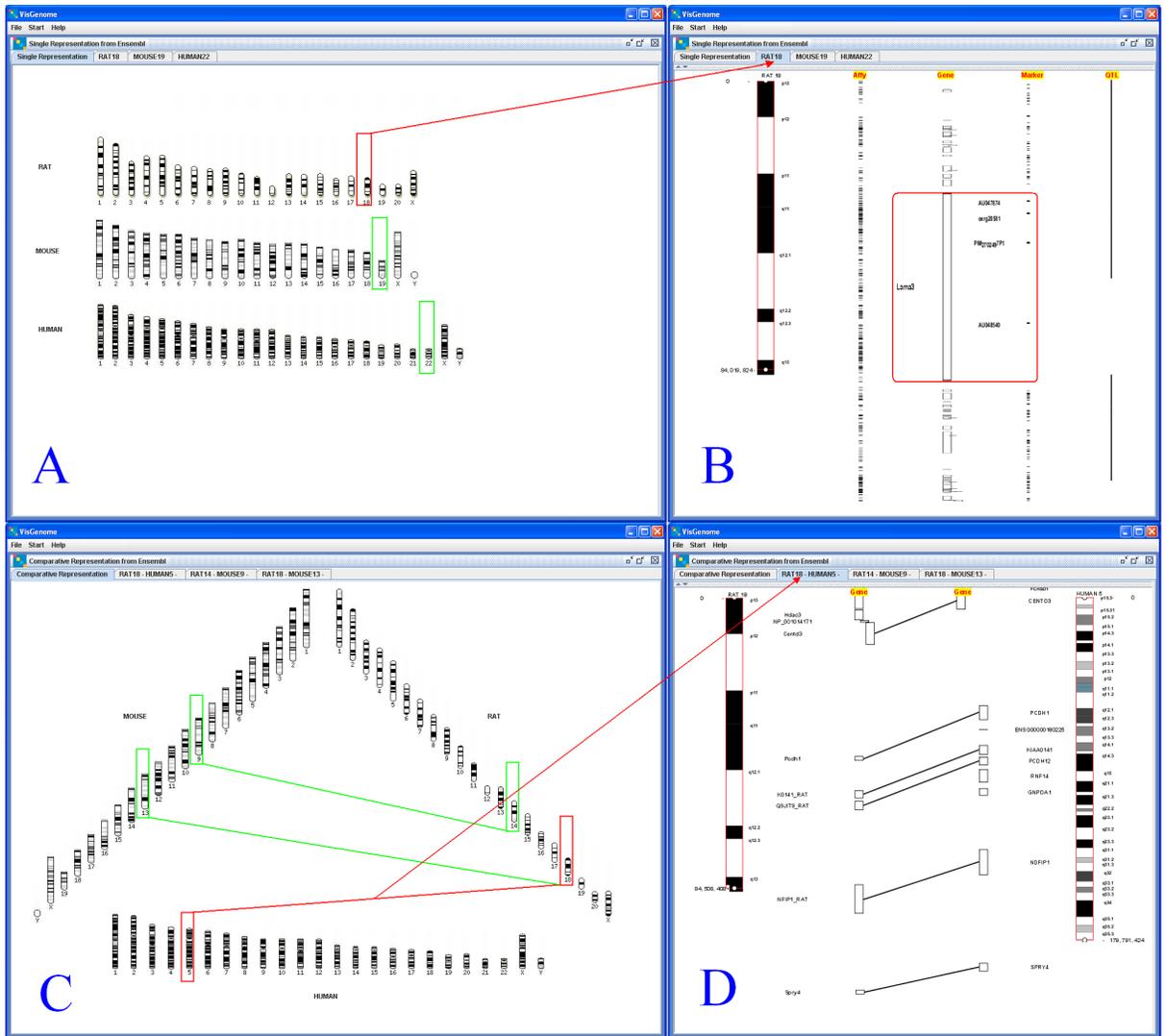


Figure 5.1: A and B - VisGenome, single chromosome view. C and D - comparative view. A and C: chromosomes from the mouse, the rat and the human. B: an overview and detail for the rat chromosome 18. D: an overview and detail for the rat chromosome 18 and the human chromosome 5. The arrows connecting views A and B and C and D are not part of the application but show the relationships between the two views.

to see in detailed view. They choose the representation by selecting a menu item or by typing Ctrl+S (single) or Ctrl+C (comparative). When they see all data (micro array probes, genes, markers, QTLs, and bands) for a selected chromosome, the tool gives them the possibility to see an overview of all data, but also details for each part of the data. A choice is offered for marking the chromosome region of interest and interacting only with the selected part. The users can drag or enlarge the box enclosing the region marked on the chromosome or enter the region coordinates in the top panel. To make the view clear,

instead of presenting all information in one view, we use an info panel (top panel) which shows additional information for the selected elements on mouse-over.

5.2.2 Zooming and Panning

VisGenome offers smooth zooming which supports the visual exploration of the chromosome space, based on Piccolo [4]. This provides efficient repainting of the screen, bounds management, event handling and dispatch, picking, animation, layout, and other features. The zooming technique allows the users to keep an area of interest in focus during interaction with the data. Zooming is manipulated by the right mouse button by moving it to the right (zoom in) or to the left (zoom out). Panning uses the left mouse button. Both interactions are easy to use and the users quickly become familiar with them, as confirmed by our study (presented in the next chapter).

5.2.3 Marking a Region of Interest

The users can choose the chromosome region of interest and focus the view only on the region. This functionality is offered by both single and comparative representations. The red rectangle marking the region on a chromosome can be moved along the chromosome and each of its boundaries can be adjusted independently. The main view shows only the data for the marked region and the users manipulate the data in the selected region. This means that when the user zooms or pans in the main view, all or some of the data from the red rectangle is available. Data outside the coordinates marked by the rectangle is not shown. Users who work with a particular part of a chromosome and do not need to download all data for the chromosome found this functionality especially useful (see user studies presented in the following chapters).

5.2.4 Additional Information

Many genome browsers display all detail for all data in one view, which makes the data difficult to read. We display selected additional information in an ‘info panel’, see Figure 5.2. In a comparative representation we show two types of information. We display Ensembl id, coordinates and a description for each element which is pointed to by a mouse. In a comparative representation when the user points to an element from one chromosome which has a homology with an element from the other chromosome, the additional information is displayed for both genes, see Figure 5.2. Display Options Tab allows the users some data manipulation, like choosing the range of the chromosome region displayed. In our solution we do not attempt to display all information in the main view and this improves usability.

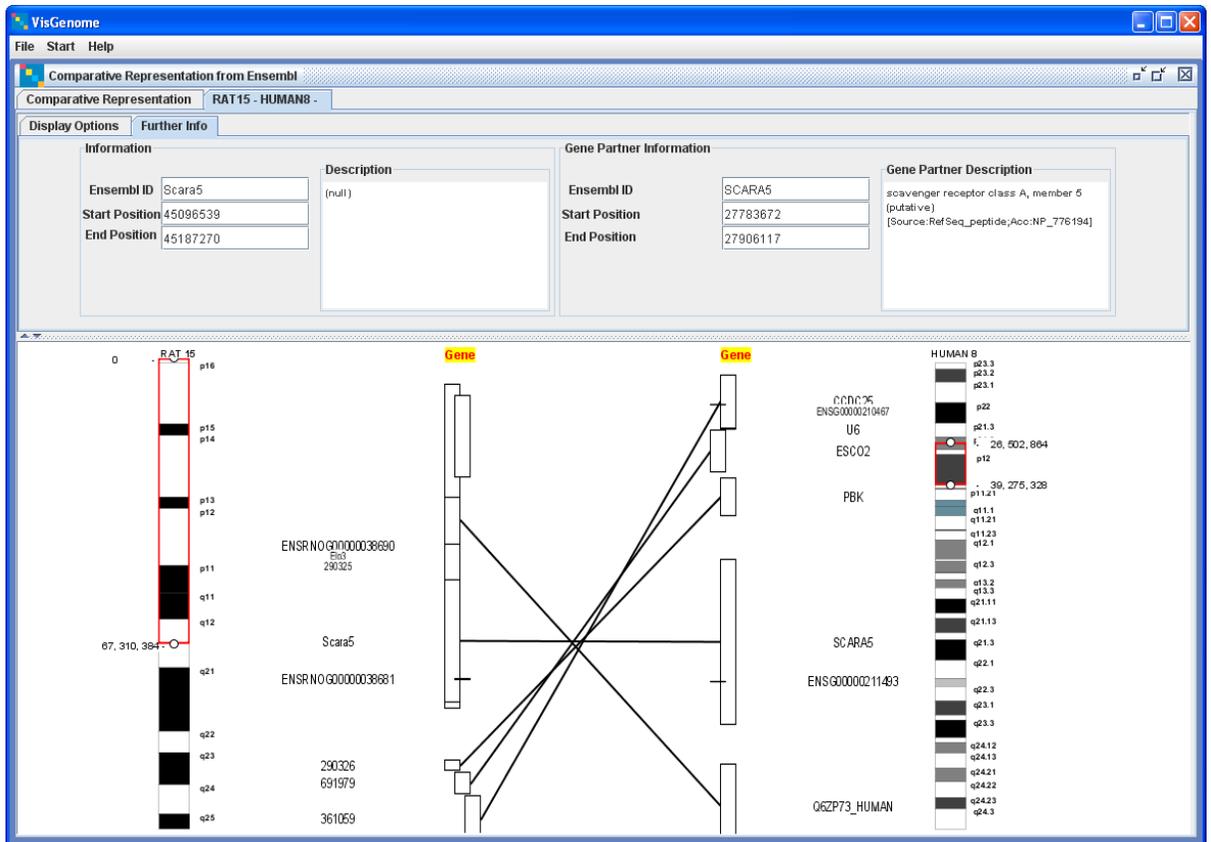


Figure 5.2: Additional information is presented for gene SCARA5 in the rat chromosome 15 and the human chromosome 8 in the Info Panel.

5.2.5 Supporting Data

Access to Ensembl is provided via clicking (right, left, or middle mouse button - this is changed in the second version of VisGenome where a user has to press **Shift** and click) on a feature of interest, which invokes Ensembl web pages in the user's browser. The first version of VisGenome implements this feature for genes only.

5.2.6 Implementation

VisGenome was written and tested under Windows XP with Java 1.5, on a 2.39 GHz Pentium 4 with 512 MB RAM. VisGenome connects via JDBC to Ensembl mart (<http://www.ensembl.org>) and the databases for the rat, the mouse, and the human. During the tests I visualised chromosome bands, micro array probe sets, QTLs and genes. Eclipse, www.eclipse.org, was used as a development environment. The application is packaged as a jar file. The user has to install Java prior to invoking the code. 1280 x 1024 screen resolution was used for testing.

5.2.7 Availability

VisGenome is available at the web page <http://www.dcs.gla.ac.uk/~asia/VisGenome> for Java 1.4 and Java 1.5 as database and file version, and as a Java Web Start Server. The web site also offers access to the source code.

5.3 Caching

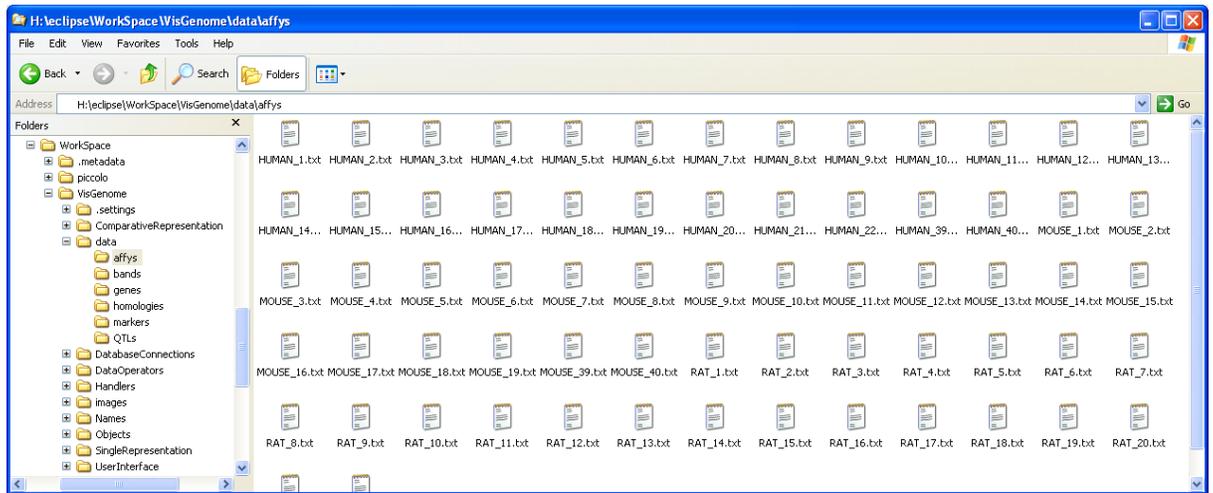


Figure 5.3: The tree hierarchy of cached data for VisGenome.

VisGenome is available in a file version which contains data downloaded from Ensembl. The data for bands, genes, markers, QTLs, micro array probes and homologies are in separate directories. Each chromosome for each species has its own file containing the data, see Figure 5.3.

The file version of VisGenome was developed in response to the needs of the users who took part in a user study comparing VisGenome and Ensembl, as the Ensembl database does not perform consistently and it is sometimes difficult to obtain data from. When the users used VisGenome, Ensembl was very slow, and therefore the file version of VisGenome was needed. The data takes up 173 MB and is available as a zip file, see <http://www.dcs.gla.ac.uk/~asia/VisGenome>. Data retrieval time presented in Table 5.1 (in ms) shows that the file version is faster than the database version. Presenting a view with data from a file is around 94 times faster for the file version than for the database in the comparative representation (see H1 - M1 in the Table 5.1). For data in the single representation the difference is not so large, but it is still at least 3 (H1) to around 13 (R18) times faster to see the view when data is taken from a locally cached file.

Chromosome	VisGenome - File Version	VisGenome - Database Version
H1	7,996	30,190
R18	1,169	15,711
M7	2,089	19,451
R18 - H5	857	72,117
H1 - M1	1,324	124,642
RX - MX	874	70,105

Table 5.1: Time in milliseconds for obtaining data from a file and from Ensembl database in VisGenome, based on one execution. The data is for selected rat, mouse, or human chromosomes. For example, H1 means that the data is for the human chromosome 1 (in the single representation), R18 - H5 means that the data is for a comparison between the rat chromosome 18 and the human chromosome 5 (in the comparative representation). H1 - M1 comparison involves two largest chromosomes in the human and the mouse, and Ensembl database access took over two minutes (124,642 ms).

5.4 Java Web Start

Java Web Start allows standalone Java software applications to be deployed with a single click over the network. It is a framework developed by Sun Microsystems that allows application software for the Java Platform to be started directly from a web browser. Unlike Java applets, Java Web Start applications do not run inside the browser, and the sandbox in which they run does not have to be as restricted, although this can be configured. One chief advantage of Java Web Start over applets is that it overcomes many compatibility problems with browsers' Java plugins and different JVM (Java Virtual Machine) versions. Java Web Start also provides a series of classes in the `javax.jnlp` package which provide various services to the application. Most of these services are designed around the idea of allowing carefully controlled access to resources while restricting the application to authorised operations.

We used this technology with VisGenome. First, we prepared a database and a file version of the application and created a key which we used to sign it. Second, we created a `.jnlp` file, see Figure 5.4, which gives access to VisGenome situated at a Java Web Start Server. The solution allows the users to start the application without the need to explicitly download and install it, as they simply click on the link on the developer's web page.

5.5 Package Structure

VisGenome contains six main packages, see Figure 5.5, among which `UserInterface` package controls everything drawn on screen. It uses data and structures from `SingleRepresentation` and `ComparativeRepre-`

```

<?xml version="1.0" encoding="utf-8"?>
<jnlp spec="1.0"
  codebase="http://compbio.dcs.gla.ac.uk/software/VisGenome"
  href="http://compbio.dcs.gla.ac.uk/software/VisGenome/VisGenome_DatabaseVersion.jnlp">
<information>
  <title>VisGenome</title>
  <vendor>Genome Visualisation</vendor>
  <offline-allowed/>
</information>
<resources>
  <jar href="VisGenome\_DatabaseVersion.jar"/>
  <jar href="mysql-connector-java-5.0.5-bin.jar"/>
  <j2se version="1.3+" initial-heap-size="256M" max-heap-size="256M"/
  href="http://java.sun.com/products/autodl/j2se"/>
</resources>
<application-desc main-class="UserInterface.VisGenome"/>
<security>
<all-permissions/>
</security>
</jnlp>

```

Figure 5.4: The .jnlp file which allows the user to connect with Java Web Start Server and run VisGenome.

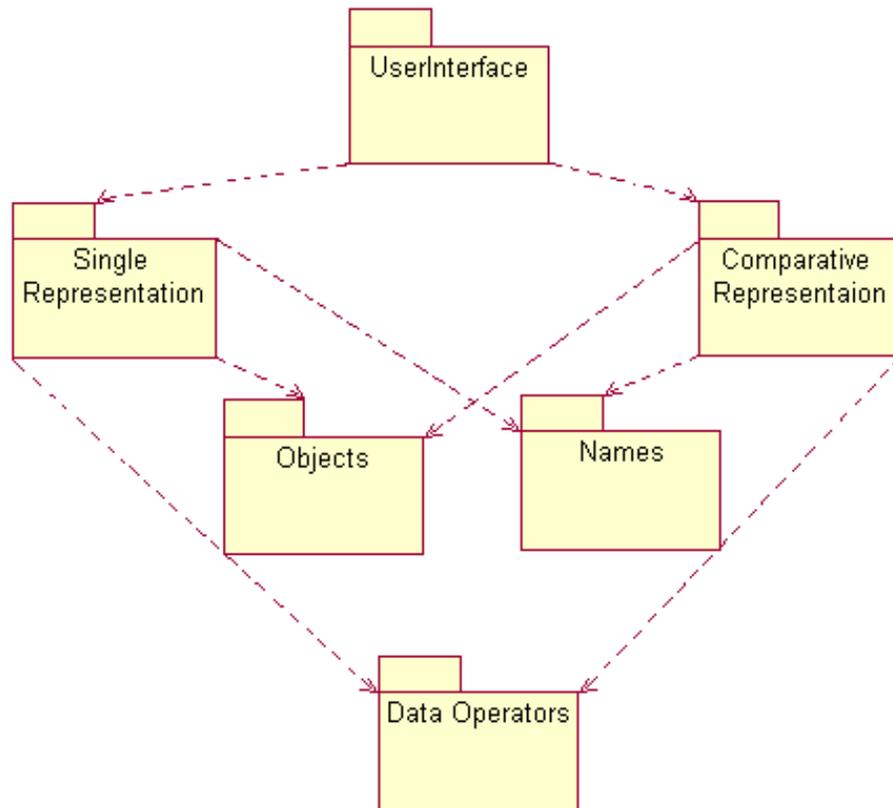


Figure 5.5: Overall package structure of VisGenome.

sentation packages. We split data structures into SingleRepresentation and ComparativeRepresentation because of the use of different layouts. In a comparative representation we show more data, including the connections between two chromosomes, and additional data about homologies. However, in both representations we use the same objects (package Objects) and the same object names (package Names). We also use a package DataOperators in the database version of VisGenome. The package contains classes

responsible for database connection and query execution.

5.6 Conclusion

The chapter outlined the main features of our first version of VisGenome. VisGenome extends SyntenyVista [46] with a comparative genome display and presents homologies alongside QTLs and micro array probes. We introduced features offered by the application and briefly described Java Web Start, and the database, and file versions of the application.

In Chapter 3 we presented requirements for a genome browser, which could be useful in biologists' work, see Table 3.1. We also showed that none of the existing genome browsers fulfils the requirements. In this chapter we introduced VisGenome, which presents a number of different kinds of data, especially genes, markers, micro array probes, and QTLs for the rat, the mouse, and the human chromosomes. It also allows the users to show their own data and compare it with the existing data from other sources. Moreover, VisGenome implements easy navigation, which will be shown in Chapters 6 and 8. It is important because this allows the biologists to do their work more effectively. These features make our genome browser better than all the other existing systems which do not conform to these requirements.

Next chapter presents the use of VisGenome in our initial quantitative user study. We test VisGenome and Ensembl to identify the best features in the two genome browsers. We improved VisGenome after the first user study and present its second version in the chapter 'VisGenome - Extension' (Chapter 7).

Chapter 6

Initial Quantitative User Study

This chapter presents our first user study with the first version of VisGenome, introduced in the previous chapter. We briefly describe the methodology and the results from the initial quantitative user study. The work presented in this chapter was published as a technical report [53] and then as a paper at Data Integration in the Life Sciences (DILS) [52]. During the user study we received feedback from the medical researchers which we fed into the second version of VisGenome presented in the next chapter.

6.1 Introduction

With the growth of biological data volumes it is becoming difficult to find the correct biological information and put it in the right context. A number of genome browsers show similar data differently, but, to our knowledge, their development was not accompanied by usability studies, see Chapters 3 and 4 (‘Genome Browsers’ and ‘HCI - Design and Evaluation’). During the study we compared VisGenome to Ensembl which is the most popular browser in the BHF Glasgow Cardiovascular Research Centre. An ethics application for this study, submitted in July 2006, can be found in Appendix H.

6.2 A User Study

We studied the usability of Ensembl and VisGenome. Although the tools offer similar functions, Ensembl shows more data types than VisGenome, as VisGenome does not show sequence level data or gene structure. In contrast, VisGenome shows comparative representations of genes and gene expression results.

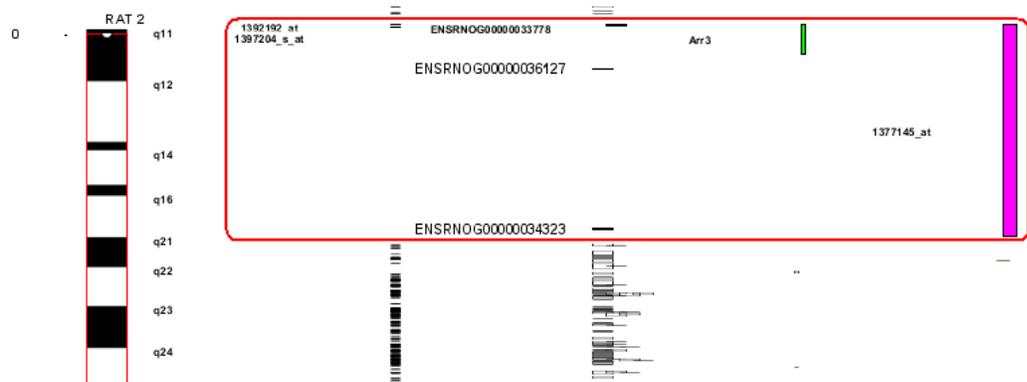


Figure 6.1: VisGenome Single Representation for rat chromosome 2. From left to right: chromosome overview, micro array probes, genes, eQTLs and pQTLs [45], and micro array probe sets from a user’s experiment are shown in one view within which the user can move smoothly and quickly.

6.2.1 Participants

We first carried out a pilot experiment with two subjects from the Bioinformatics Research Centre (BRC) and five from the BHF Glasgow Cardiovascular Research Centre - Western Infirmary (WI) in Glasgow. In the full experiment we had 15 participants from the WI and the BRC. Six of them use Ensembl often (Ensembl Experts: Ex). Nine of them use different tools, such as BugView [60], UCSC GenomeBrowser [56] or AtIDB [83]. Some were from BRC and do not use genome browsers but know them from presentations (NonExperts: NEx). Three of the participants (Ex) previously took part in a one day Ensembl course.

6.2.2 Methods

None of the biologists had used VisGenome before the experiment. I gave a short presentation of VisGenome to all subjects. I explained how Single and Comparative Representations work, how the users can select a chromosome, zoom, pan or mark an interesting region of a chromosome. I showed the additional info panel. Several researchers asked us to remind them first how Ensembl works and where to find information (three participants - NEx). I gave them a short introduction to Ensembl. As I am not an Ensembl expert, I showed them how to zoom, scroll, search for a gene or a chromosome in Ensembl, and how to mark a region under interest which was all that was needed to carry out the test. I presented SyntenyView, MultiContigView, CytoView, and ContigView. I demonstrated how to obtain information from BioMart or RGD using Ensembl. Before the experiment we offered the subjects the opportunity to carry out an experimental task in VisGenome (for NEx also in Ens). We did not randomise task order and the VisGenome task came first. The order in which the tools were attempted is thus a confounding

factor; although a positive effect on the performance for the second attempted tool (Ensembl) is the most likely consequence of this, Ensembl performance was not better than VisGenome.

In order to make the experiment realistic, two WI subjects had asked to see their own experimental data in addition to the shared experimental data set. To that end, we created one version of VisGenome for the majority of participants and two specific versions with private data. In those versions micro array probes were coloured in both Single and Comparative Representations, see Figure 6.1. The aim was to receive more feedback from those subjects.

The experiment was divided into two parts (Ensembl and VisGenome). We explained to the participants what we understand by Single and Comparative Representation (data for a single chromosome representation or for a comparison between two chromosomes) and that VisGenome offers Single and Comparative Representations, but in Ensembl the subjects have to decide if they would like to use MultiContigView or SyntenyView as Comparative Representation, and ContigView or any other Views as Single Representation. Some of the participants asked us if they can use BioMart [122] or RGD [144] (2 users) during the execution of Ensembl task. They could use all tools available from Ensembl pages. During the experiment the participants could give up if they thought that it was not possible to complete the task. The majority of subjects attempted the tasks and only one person gave up and abandoned tasks T2 and T3, as described below.

6.2.3 Search Tasks

Rather than choose our own tasks, which might have created a bias in favour of VisGenome, we asked our biological collaborators to recommend some common search tasks. Our collaborators use comparative genomics to find syntenic regions on chromosomes from different species (mouse chromosome 3 and human chromosome 1 in [65]). Therefore, one of the tasks is to find homologies between genes (T2). The biologists also analyse QTLs and make studies based on comparisons between QTLs from rats [109] - see T3. They look for genes responsible for hypertension in a chromosome or in specified chromosome regions [34] - see T1. Our biological collaborators defined three tasks, as follows.

- T1.** Single Representation. Choose one of the rat, mouse or human chromosomes. Mark the whole chromosome and show all available data. Then choose the region between 100bp and 10,000,000bp and note the name of the first gene and the last Affymetrix probe inside the region.
- T2.** Comparative Representation. Choose rat chromosome 18 and human chromosome 5. Zoom in and out to find any homologies between genes. Then choose one of the homologies and read out the gene names of the homologous genes.
- T3.** Single Representation. Choose one of the rat chromosomes. Find the longest QTL. Then zoom in on it and write down the names of the genes which are the closest to the beginning and the end of

the QTL.

We captured screen usage as videos, recorded the time used for each task in minutes (STi, search time), and counted the number of mouse clicks (NoMc) for all tasks in VisGenome and Ensembl. On finishing the tasks, the subjects filled in a questionnaire and participated in an interview.

6.2.4 Questionnaire

We offered all subjects a questionnaire in order to study their perception of the mental and physical demands of the tasks. The scale used had 21 points, from -10 (low/poor) to +10 (high/good). All 15 subjects answered the following questions, once with regard to Ensembl and then with reference to VisGenome.

- Q1.** How much mental and visual activity was required?
- Q2.** How psychologically demanding did you find this experiment?
- Q3.** How much time pressure did you feel because of the rate at which things occurred or the time limit imposed on the task?
- Q4.** How hard did you work?
- Q5.** How successful do you think you were in doing the task set by the experimenter? How satisfied were you with your performance?
- Q6.** How much frustration did you experience?
- Q7.** How annoying did you find the mouse manipulations used in the experiment?
- Q8.** Which of the systems do you prefer?

6.2.5 Interview Questions

During an interview after the experiment we asked the participants to answer the following questions.

- Q9.** How often do you use a computer during your work?
- Q10.** How often do you use a genome browser during your daily work?
- Q11.** If you use a genome browser, please give the name of the one you use the most frequently.
- Q12.** What do you like/dislike about Ensembl?
- Q13.** How often do you use Ensembl in your daily work?

- Q14.** What do you like/dislike about VisGenome?
- Q15.** Do you think the fisheye visualisation technique is useful? (Do you like it?)
- Q16.** Do you think excentric labeling is useful? (Do you like it?)
- Q17.** Do you use panning? (Do you like it?)
- Q18.** Do you use zooming? (Do you like it?)
- Q19.** Is zooming via buttons, for example in Ensembl, better than via mouse action?
- Q20.** Do you use scroll bars, for example in Artemis or UCSC Browser?
- Q21.** Which other visualisation techniques in VisGenome and Ensembl seem to be helpful?
- Q22.** Are the colours in the visualisation meaningful for you?
- Q23.** If you use the colours at all in the visualisation, please say how you use them.
- Q24.** Which of the representations of chromosomes do you prefer? (Karyotypes in three rows, karyotypes in a triangle, coloured histogram, see Figure 6.4.)
- Q25.** Is it important for you to have any additional information about the genes such as presented in VisGenome in Panel Info? What would you like to see on it?

Questions Q12 and Q13 refer to Ensembl, question Q14 refers to VisGenome, all other questions from the interview refer to common visualisation techniques and usage habits.

6.2.6 Task Benchmark

We first carried out a test, benchmarking the tasks. We were interested what results are obtained by the developer (thesis author), if she succeeded or not, and how fast the tasks were executed by the developer and by the participants. For VisGenome we observed: for T1, 30 NoMC, 2min 22s; for T2, 16 NoMC, 1min; and for T3: 38 NoMC, 2min 40s. On our first attempt for T2 in Ensembl we used 6 NoMC, 1min, for T1 the browser crashed a few times, and then during T3 in Ensembl we abandoned the test, as Ensembl became unavailable. On another day, we carried out T1 and T3 in Ensembl successfully. T1 took 3min 40s and 30 NoMC, and T3 took 15min 10s and 51 NoMC. We summarise this benchmark test in Table 6.1. Each of the tasks could be carried out using different views - Ensembl, and techniques - such as marking region of interest by red square or by entering coordinates in VisGenome. A sample task execution scenario for VisGenome and Ensembl is presented in [53]. We estimated that a biologist would need around an hour to complete the experiment, including the time needed for the introduction and the questionnaire.

	VisGenome		Ensembl	
	Time	NoMC	Time	NoMC
T1	00:02:22	30	00:03:40	30
T2	00:01:00	16	00:01:00	6
T3	00:02:40	38	00:15:10	51

Table 6.1: Benchmark time and NoMC for T1, T2, and T3 in VisGenome and Ensembl.

T1 in VisGenome

We expected each participant to choose the Single Representation from the menu and then drag one of the chromosomes into the lower panel. After clicking on the selected chromosome in the lower panel, the user has to mark the whole chromosome by stretching the red box to the top and to the bottom of the karyotype image. Subsequently, the user has to zoom out to see all genes, Affymetrix Probe Sets and QTLs. Then, in the top panel, the user chooses the region between 100 bp and 10,000,000 bp and notes the name of the first gene, and the last Affymetrix probe inside the region.

T2 in VisGenome

In T2 we expect the users to choose the Comparative Representation and select the rat chromosome 18 and the human chromosome 5. They zoom in and out to find any homologies between genes, and then choose one of the homologies and show the names of the homologous genes.

T3 in VisGenome

In T3, the participant selects the Single Representation and chooses one of the rat chromosomes. To find the longest QTL, we marked the whole chromosome and zoomed out. Then we zoomed in to the beginning of the longest QTL and showed the name of the gene which was the closest. We zoomed out to see the end of the chromosome and finally we zoomed on the end of the QTL and showed the name of the gene which was the closest to the end of the longest QTL.

T1 in Ensembl

All the tasks in Ensembl start from the main Ensembl web page. In T1 we chose a chromosome by clicking on it and then we chose the region between 100bp and 10,000,000bp and show the name of the first gene and the last Affymetrix probe inside the region, which in Ensembl is marked as Oligo (we explained this to the users before they begin of the task). We used in T1 MapView, CytoView and FeatureView offered

by Ensembl.

T2 in Ensembl

In T2 we chose the rat chromosome 18 and went to the SyntenyView for human chromosomes. Ensembl shows us two data columns to the right of the view. The first column shows all rat genes and the second column their human homologues. In the experiment it was enough to find one homology, but some users became interested in the data and showed a few homologues.

T3 in Ensembl

In T3 the participants choose one of the rat chromosomes. Then, to find the longest QTL, we mark the whole chromosome by entering the region between 1 and 59,218,465bp. We can easily find the longest QTL in CytoView. Because Ensembl does not show detailed data for large regions, we marked a small region close to the beginning of the longest QTL, and showed the name of the gene which was the closest to the beginning. Then we go back one step and zoom into the end of the longest QTL and show the name of the gene closest to the end of the QTL.

6.3 Experimental Results

The results are quite surprising. Even the researchers who use Ensembl frequently are often unsuccessful in task execution. The experts encounter no problems when focusing on one chromosome fragment. However, when they examine similar data in a different part of the chromosome, they encounter problems. We also found that some of the zooming mechanisms in VisGenome were hard to use and that the subjects prefer mouse clicking to dragging. The researchers want to see large amounts of data, but when they are looking for a particular object, they prefer to see only a small part of the data under investigation.

During the analysis of the data we used a number of statistical methods, presented in Appendix O. Because of the small number of users, data types (normally distributed or not normally distributed), and conditions required for some tests, we could not analyse the data, always as we had hoped. We wanted, for example, to use McNemar's test to analyse mouse clicks, but because of the test condition we could not do this, see Appendix O. Therefore, after consultation with the statistician, we chose statistical tests that could be applied to our data and could show us any significant results from the statistical point of view.

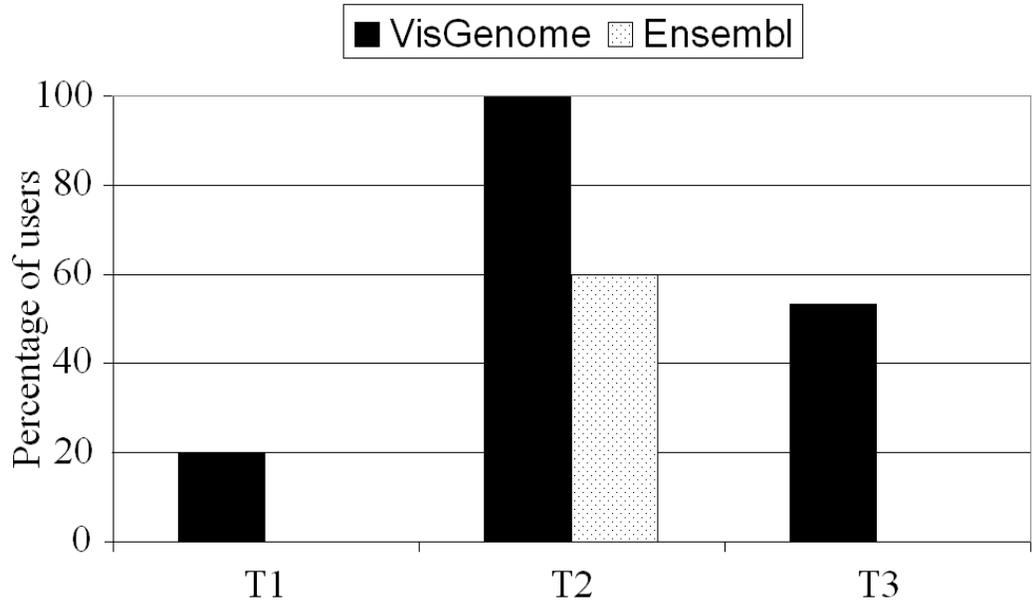


Figure 6.2: Percentage of subjects (out of 15) who completed each task (succeeded).

6.3.1 Accuracy and Task Completion

Figure 6.2 shows that T2, the only task involving comparative genome representation, was more successful with VisGenome (100%) than with Ensembl (60%, 9 subjects). In T3 53% of attempts were successful in VisGenome (8 subjects), while in Ensembl the success rate was 0. In T1 we note 20% success rate in VisGenome and 0% in Ensembl. Using the two-sided sign test (where 0=both/neither successful; 1=VisGenome success but Ensembl not; -1=Ensembl success but VisGenome not) as an alternative to McNemar’s test [8], the success rate for VisGenome was significantly greater for both T2 ($P=0.0313$) and T3 ($P=0.0078$), but not for T1 ($P=0.25$). The null hypothesis for these tests was that the proportion of successes was the same for both VisGenome and Ensembl, and the alternative was that they were not. Completion rates were higher in VisGenome than in Ensembl for all tasks, particularly for T2 and T3. This may be due to the fact that Ensembl is a much richer interface, with many more options and controls and represents more data. Possibly, the subjects were not able to find out how to generate comparative genome views, or were getting lost while learning to use Ensembl (NEx and Ex).

6.3.2 Time to Finish

Time was measured in minutes. The biologists who completed the tasks had means of $\text{Mean}(T1)=5.69'$ ($\text{StDev}=1.39'$), $\text{Mean}(T2)=3.58'$ ($\text{StDev}=1.17'$), and $\text{Mean}(T3)=5.29'$ ($\text{StDev}=0.97'$) in VisGenome and $\text{Mean}(T2)=2.83'$ ($\text{StDev}=1.76'$) in Ensembl. As no one completed T1 and T3 in Ensembl, statistics

were calculated only for T2. In T2 in Ensembl and VisGenome, 9 researchers correctly completed both tasks. As the differences in times were not normally distributed, the Wilcoxon signed rank test was used ($P=0.554$). We realised that Ex used both tools differently than NEx. Ex usually wanted to see more information, got interested in the data, while NEx subjects just wanted to complete the task. Ex tried to find and show all possible answers they knew, explore while doing the task, and to be sure they had the best answers. If there were several ways of doing the tasks in Ensembl they wanted to show all the solutions. In T2, for example, it was enough to show two genes in VisGenome and Ensembl, and most NEx did that and finished quickly. Most Ex performed T2 and then explored MultiContigView to see more information about homologous genes, which took more time. Users behaved similarly in T3, however nobody succeeded in Ensembl (see 7.4 The Frequent Errors). NEx showed Affymetrix probes in ContigView, while Ex used FeatureView and looked at the detail. There were also slight differences in server response times for Ensembl which might have influenced the speed of data analysis. Overall, in T2 there was little difference in task execution time between Ensembl and VisGenome.

6.3.3 Mouse Clicks

Those who completed the tasks had the means for T1 of 53 (StDev=9.54), for T2 of 51.07 (StDev=26.65), and for T3 of 74.38 (StDev=13.38) NoMC in VisGenome, and the mean for T2 of 23 (StDev=18.93) NoMC in Ensembl. Only T2 mouse clicks were analysed, due to non-completion in Ensembl for T1 and T3. 9 subjects completed T2 with both VisGenome and Ensembl, and despite the mean number of clicks being larger in VisGenome than in Ensembl, there was no significant difference in NoMC, possibly due to the small sample size. One Ex had a very large NoMC (138) for VisGenome, and only 19 for Ensembl. This shows that mouse manipulation in VisGenome needs getting used to, as panning and zooming require keeping the left/right mouse button down and moving the mouse at the same time left/right or up/down, and the left/right movement is not offered by many similar applications where clicking on zoom bars is used instead, and smooth zooming is not widely used. This is a potential problem, however most subjects learned how to use the mouse quickly. On the other hand, Ex often clicked to see additional information and some NEx clicked because they wanted to find the solution and they were not sure where they had to look for it. This contributed to a large NoMC in some Ex as well as NEx.

6.3.4 The User Questionnaire Results

The results of the user questionnaire are summarised in Figure 6.3 (note that means are given for Ensembl and VisGenome separately, ignoring any pairing). All 15 subjects answered questions Q1 - Q7, once with regard to Ensembl and then with reference to VisGenome.

Paired t-tests on the pairwise differences between Ensembl and VisGenome gave significant results

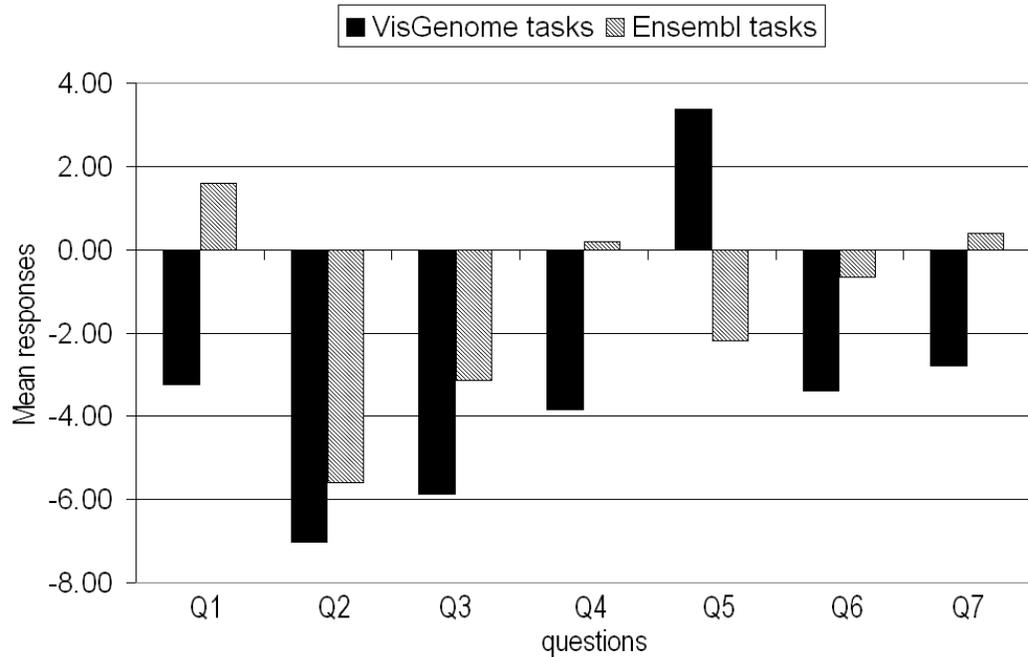


Figure 6.3: Mean response values for questions Q1-Q7 for 15 subjects. The subjects felt higher mental and physical demand during Ensembl tasks. They thought they worked harder and under higher time pressure in Ensembl. Subjectively, VisGenome tasks were more successful and less frustrating. Mouse manipulation was more annoying in Ensembl tasks.

for Q1, Q3, Q4, Q5, and Q7. In Ensembl the subjects reported more mental and visual activity than in VisGenome (Q1). Population mean difference lies between -7.7 and -1.9 with probability 95% ($P=0.003$). Answers to Q3 indicate that the subjects felt more rushed in Ensembl trial ($P=0.010$, 95% CI: (-4.7,-0.7)). Ensembl was perceived as being significantly harder in Q4 ($P=0.011$, 95% CI: (-6.9,-1.1)). In Q5 the subjects thought they were less successful in Ensembl than in VisGenome. Population mean difference for Q5 lies between 3.3 and 7.7 with probability 95% ($P=0.000$). In Q6 the subjects felt on average more frustrated with Ensembl than with VisGenome, but this was not statistically significant. We have an additional observation here, gathered directly, that subjects were frustrated by the need to learn how to use the mouse in VisGenome, but in Ensembl they were equally frustrated by the pop-up menus which suddenly appear and obscure the view. Those menus may be one of the annoyance factors in the system (and do not appear in newer versions of Ensembl). Population mean difference for Q7 lies between -6.1 and -0.2 with probability 95% ($P=0.036$). Additionally, the subjects were asked to state which of the two applications they prefer (Question 8, Q8). Significantly more subjects preferred VisGenome to Ensembl ($P=0.013$; 1-proportion test). One subject preferred Ensembl, one did not answer Q8, two said that both tools were equal, while 12 preferred VisGenome.

6.3.5 Additional Interview Questions

At the end of the experiment we briefly interviewed all the participants about visualisation techniques they know and like in genome browsers. 12 subjects know Fisheye [32] and see it as useful and 11 users like it (Q15). In response to (Q16) 11 participants said that excentric labeling [30] is useful and 10 like the technique. Zooming and panning (Q17 and Q18) are common in genome browsers and the users like them, while 14 subjects use zooming and 13 like it, and 14 users use panning and 14 like it. Only two persons preferred zooming via buttons to mouse action (Q19).

We asked the users about the use of colour and if it has any meaning for them (Q22 and Q23). This is an interesting issue, as Ensembl offers a lot of colours and VisGenome has a monochromatic display for the karyotypes, and all other data is coloured white in the current version of the application, with the exception of two participants where private gene expression data were coloured red and green. Only 8 persons answered that colours were meaningful. They would like to have the option to change the colour to mark interesting data. The subjects stressed as well that for them Ensembl colours have no meaning (only for one participant was Ensembl colouring meaningful). The subjects believe that colour only shows the grouping of data items, but if Ensembl offered horizontal lines instead of colours, this would be also a good solution. Some of the participants from BRC, touched upon the problem of colour blindness where some colours may have no meaning at all.

In (Q24) we showed the users three different representations of karyotypes (see Figure 6.4) and asked them which representation they prefer. 6 participants preferred karyotypes in a triangle, 7 liked karyotypes in three rows better (2 under the condition that they can click on the chromosomes and not drag them) and 1 person was not sure if he prefers karyotypes in three rows instead of a coloured histogram as in SyntenyVista [46]. The one user who liked the view with coloured squares motivated the choice by saying that it takes less screen space and in the future can allow the developer to add more species.

The experimental version of VisGenome used dragging in the single representation and clicking in the comparative representation. We wanted to check what was preferable. During the experiment we observed that the participants prefer clicking on a chromosome to dragging it into a display. Only one user preferred dragging because in VisGenome it allowed him to see that data is being downloaded. The majority of the participants clicked on a chromosome in the single representation in VisGenome and waited, and when nothing happened, the subjects recognised that they ought to drag instead of click. The participants liked the info panel (8 users strongly recommended it - Q25) instead of keeping everything in the main view.

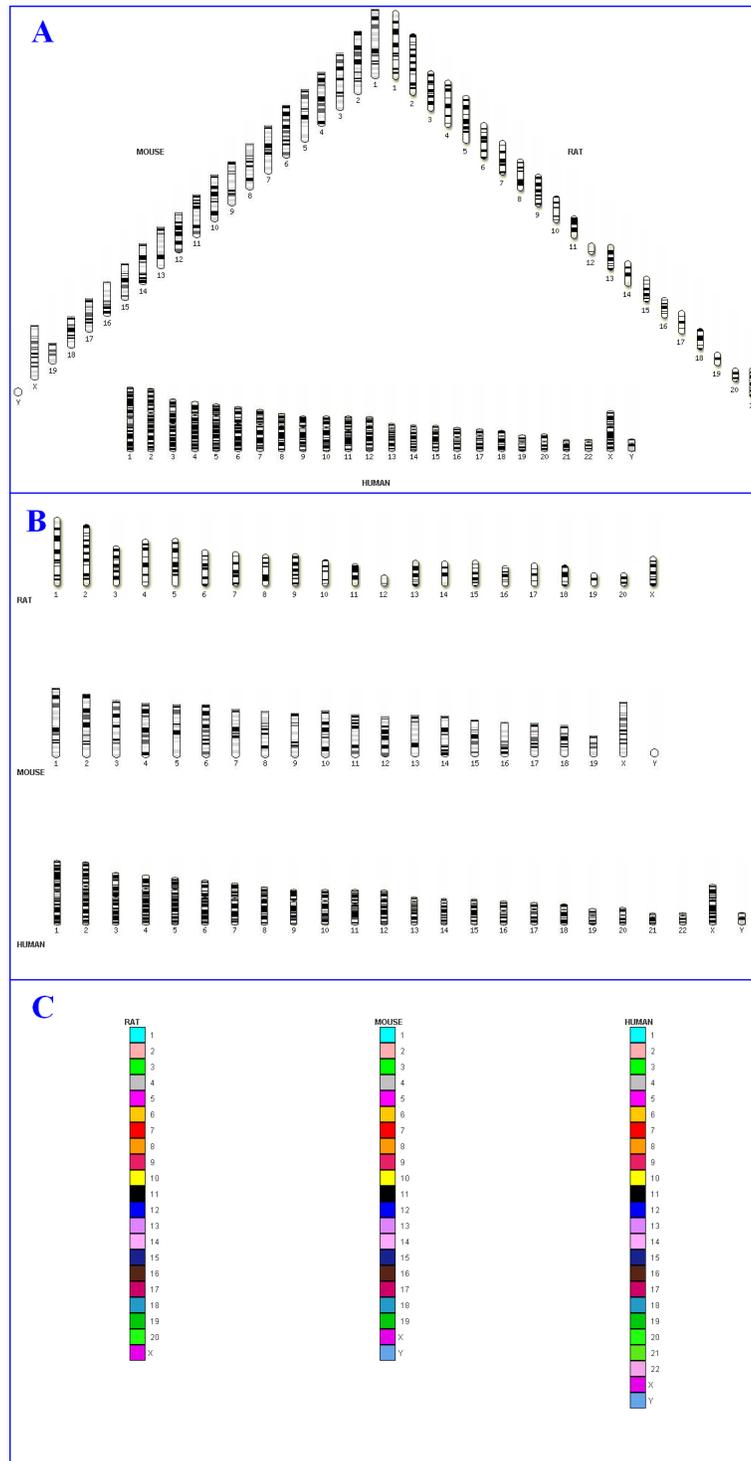


Figure 6.4: The three views for marking which chromosome the users would like to see in detail, part of Q24.

6.4 The Frequent Errors

In T1 we saw that the participants were looking for Affymetrix probes and couldn't find them even though they were accessible. However, the main cause of failure in T1 was that the subjects made mistakes e.g. typed 1 Mbp instead of 10 Mbp. In VisGenome the subjects frequently forgot to mark the whole chromosome to show all available data or marked half of the chromosome instead of the whole. In Ensembl a number of users entered the coordinates and marked 'Region' instead of 'Base pair', and some did not use the overview offered by Ensembl but tried to mark the whole chromosome in ContigView. This usually crashed the web browser and required a restart.

T3 required showing the longest QTL. In a chromosome with many small QTLs, the subjects could not decide which QTL to choose (four subjects). We suggested that they carry out the task for any of the QTLs. The same solution was suggested where several long QTLs appeared to be of similar length. 8 researchers were successful in T3 in VisGenome. The most frequent mistake in the unsuccessful attempts in VisGenome was choosing a complex of QTLs instead of one QTL. In Ensembl the subjects usually attempted to mark the entire chromosome, and only one person succeeded without crashing the browser. Some subjects tried viewing the chromosome in units of 1 Mbp but gave up after recognising that this would take too long. One user tried to use BioMart and RGD, but this did not help. Most subjects did not realise that the view shown in Ensembl is not the whole chromosome but a small part of it. Several subjects chose a chromosome, clicked on it, viewed ContigView, looked down the screen to find QTLs and saw that they were all longer than the area shown in the browser, and did not know what to do to see the entire length of the QTLs.

When we analyse both correct and partially erroneous task completions (see Figure 6.5), we see a different view of the experiment. 11 users finished T1 and T2 in Ensembl and 5 users finished T3 in Ensembl. Similarly, for VisGenome the completion rate improved. T1 was completed by 12 users and T3 by 10.

Although the use of zooming helped users, and new visualisation features required some learning, we suggest that the experiment highlights another significant issue to be addressed in future development: high error rates in data selection and query specification. The benefits of solving this problem may well out-weigh those arising from new variations on and easy learning of features such as zooming and panning. We note user training is required for both VisGenome and Ensembl. Zooming and panning by mouse manipulation was classified as something very intuitive and natural, but at the beginning of the VisGenome experiment most subjects were confused and disappointed that they had to remember which button and which direction to use to zoom or pan. Some suggested that new visualisation techniques could be bad because biologists are not familiar with them, however they said that acceptance depends on the implementation. A small number of subjects (2) suggested zooming with buttons instead of mouse

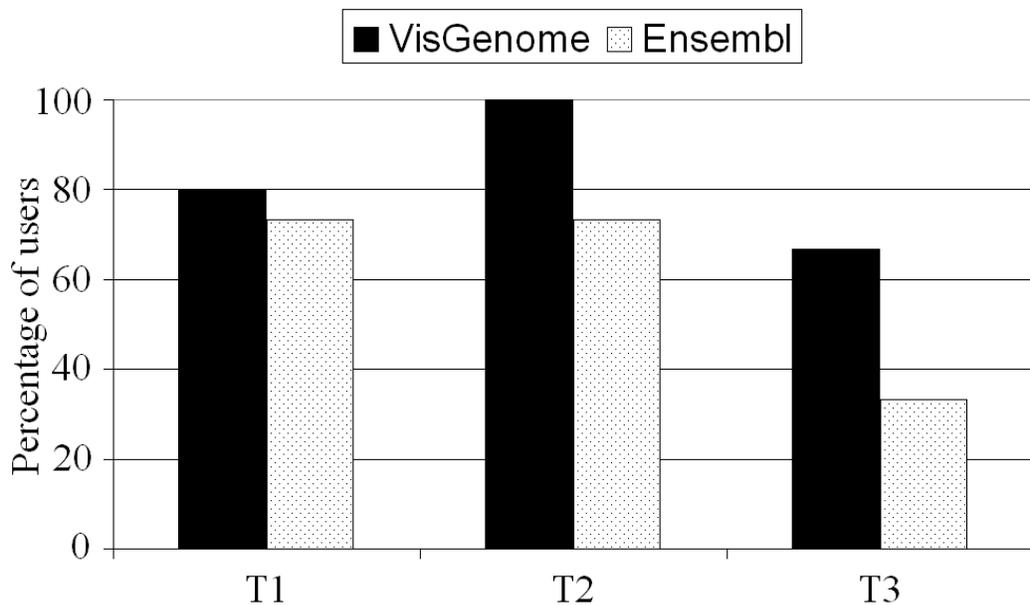


Figure 6.5: Percentage of subjects (out of 15) who finished each task (ignoring errors).

manipulation and were disappointed because of the lack of scrolling.

6.5 Conclusion

The chapter presented our first quantitative user study with the first version of VisGenome and with Ensembl. We introduced methodology used in the study and described the results. We decided to use a human-computer interaction research technique to test its validity in bioinformatics. We found that this approach gave us a number of interesting insights and was extremely helpful in focusing the development of VisGenome on providing improved support for scientific data analysis. We found out that in our experimental setup which targets the analysis of QTLs, synteny and gene expression, the subjects were more successful when using VisGenome than Ensembl. VisGenome was preferable in some aspects, as it showed less data and had fewer controls. All participants liked the techniques they know, such as scrolling and panning, and needed time to adapt to new solutions, such as mouse driven panning and zooming. The study shows that there is still large scope for the application of known visualisation techniques to bioinformatics data. Popular solutions, like semantic zooming, offered by Google Maps, could be very useful in biomedical work. Indeed, Helt and coauthors [39] used it in one of the first genome browsers - BioViews. They offered two types of visualisation: a physical map and a sequence-based map, and supported some forms of semantic zooming. They focused on the importance of easy interaction techniques and data availability through the internet.

During our study a list of user suggestions and requests was gathered, which have been fed into the second version of VisGenome presented in next chapter. The suggestions were mainly combined with visualisation techniques which allow the users to easily navigate their data. They are as follow:

- instead of dragging, use clicking in the single representation,
- add *Focus On*,
- add colour

Also, during the conducting of the experiment, new criteria arose, as follows:

- show homologies before the second chromosome is chosen in the comparative representation,
- improve labelling,
- support viewing data from Ensembl - links for all kinds of data,
- modify panel info.

To improve visualisation in VisGenome we also decided to develop a new scaling algorithm, which allows the users to see exactly what they want, i.e. the position of one kind of data according to other data. We decided to show data not in its original size, but also modified and dependent on other data types.

We recognised that the criteria put at the beginning of VisGenome development were very important and helped during the tasks' execution. Visualisation techniques, such as panning and zooming allow the users to easily navigate their data, the participants were more successful in looking for small parts of data and genes comparison between two chromosomes in VisGenome than Ensembl.

Chapter 7

VisGenome - Extension

This chapter presents the second version of VisGenome which was used during our multi paradigm user study (presented in next chapter). We briefly describe the features introduced to the second version of VisGenome. The work presented in this chapter was published as a technical report [49] and then as a paper at International Conference on Computational Science (ICCS) [50].

7.1 Introduction

In Chapter 5 we described the first version of the VisGenome application, focussing on its basic functionality. However, during the initial quantitative user study described in Chapter 6, we received feedback from the users and decided to create a new version of VisGenome presented in this chapter. All functionalities added in the second version of VisGenome are the results of our observations from initial quantitative user study (scaling algorithm, additional information) or biologists suggestions (colours) or, very often, both i.e. biologists' suggestions and results from the user study (homologies, labelling, supporting data, focus on).

During the user study, we recognised that the participants prefer clicking to dragging, i.e. they preferred choosing chromosomes in the comparative representation by clicking on them than selecting them in the single representation by dragging. The second version of VisGenome offers only clicking in both representations.

The medical researchers wanted to see homologies before the second chromosome is chosen in the comparative representation. They also suggested that we improve labelling (labels overlap and become too big during zooming), support the viewing of data from Ensembl (links for all kinds of data, not only for genes), and add focus on and colour options. Because of the changes, the 'additional info' panel was

also modified - we added buttons for the new functionality. We also developed a new scaling algorithm which we call CartoonPlus. CartoonPlus allows the users to see data more clearly by choosing one kind of data as basis and scaling other data types in relationship to the basis. The solution does not show data in its natural size but allows one to see connections between different kinds of data more clearly, especially in the comparative representation. The subsequent evaluation of CartoonPlus showed that this type of distortion is helpful, as demonstrated via our user study, see pages 142 and 151.

7.2 Visualisation Extensions

7.2.1 Homologies

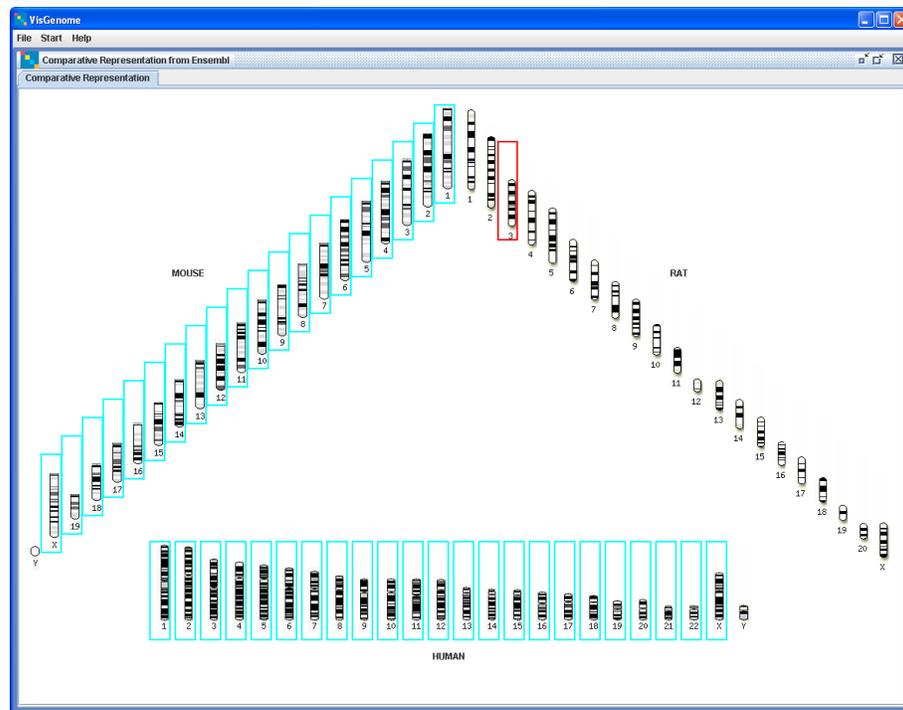


Figure 7.1: The rat chromosome 3 is chosen and other chromosomes which have homology with the rat chromosome 3 are surrounded by blue boxes.

To support comparative genome analysis, we show chromosomes which have homologies with other chromosomes. Our solution allows the users to identify all homologous chromosomes quickly. When a user looks at all chromosomes in a number of species, and clicks on one, all the homologous chromosomes in other species are highlighted, which facilitates the choice of homology for visual analysis (see Figure 7.1).

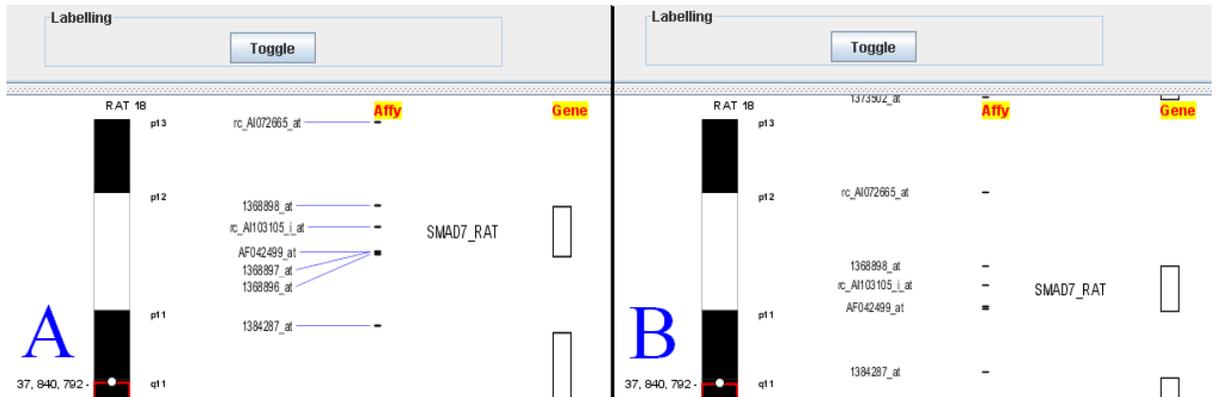


Figure 7.2: The single representation for rat chromosome 18. A shows all labels and links which connect labels with the elements. B shows only a selection of labels that can be shown next to the objects they relate to. The **Toggle** button is used to switch between A and B.

7.2.2 Labelling

Because of the large amount of data, there is a problem with labels, especially for elements that have the same location. To solve the problem, we allow the users to switch between viewing all labels and only a selection. When all labels are visible, they are connected by blue links to the visible elements. When the user moves the mouse close to the element, a link becomes highlighted, which allows the user to localise the element description faster, see Figure 7.2 A. In selected label view, Figure 7.2 B, we display only a small subset of labels. If there is enough room, the element name is displayed. For elements with the same coordinates, it is the first element in alphabetic order. We show as next the label for the next element which has enough room to show its label.

7.2.3 Supporting Data

Ensembl offers data collected from publications and experiments. To help the user contextualise the data we provide access to Ensembl, activated by clicking on a feature of interest. This invokes Ensembl in the browser (Figure 7.3). The functionality is available for all data taken from Ensembl.

7.2.4 Focus On

Focus on (Figure 7.3) makes the focal element large enough so that its name can be read, moves it to the centre of the view, and marks its boundaries in red. This allows the user to see a small part of a viewing history until he changes the region of interest. This means that the user can see which elements he focused on during the session. In a single representation, when the user focuses on an element, all neighbouring elements in the view become proportionally larger in all columns. In a comparative representation only

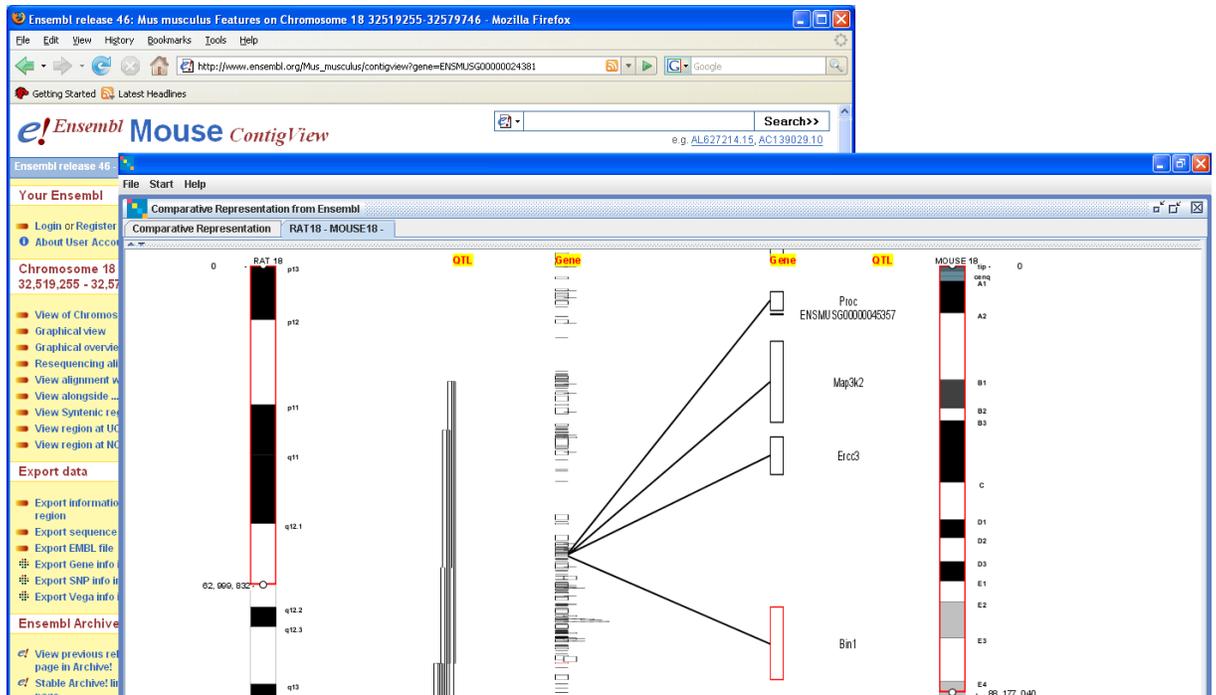


Figure 7.3: The comparative representation for the rat chromosome 18 and the mouse chromosome 18. The gene Bin1 in the mouse chromosome 18 is in focus. The background shows additional information from Ensembl for gene Bin1, activated by clicking on the gene.

elements in the chromosome containing the chosen element are changed, and all elements on the other chromosome maintain the same size. This allows the users to see an overview of elements from one chromosome and details for the selected element in the second chromosome. If the user wants all elements in the two columns to be of the same size, he chooses focus elements in both. Then we set the size of all elements to be the same.

7.2.5 Colours

We use black and white for most data, however, after marking a region of the chromosome, the user can choose colour for each of the elements by clicking on the object while pressing Alt. The default colour choice view is displayed and the user can change the colour of the marked element, see Figure 7.4.1. Additionally, the object boundaries are marked in red during *focus on* and all bands in the chromosomes are coloured by standard colours. We added the colour option in response to biologists' suggestions. They wanted to see their data coloured which would help them see differences between different results from their experiments.

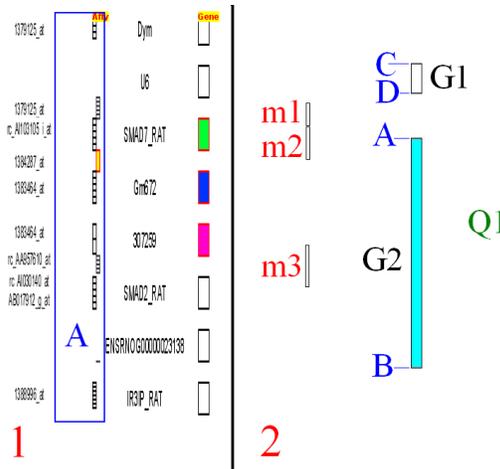


Figure 7.4: 1: Single representation for the rat chromosome 18. Three genes (SMAD7_RAT, Gm672, 307259) and one micro array probe set (1384287_at) are coloured by different colours, selected by the user interactively. 2: CartoonPlus algorithm (see Figure 7.6). G2 is a gene which begins at A and ends at B. m1, m2, and m3 are elements smaller than G2 and Q1 is bigger than G2.

7.2.6 Additional Information

VisGenome supports the display of additional information about the presented data in an adjoining panel, the ‘info panel’, see Figure 7.5. This data (from Ensembl) includes the start and end of each object, its name, and description. The data shown corresponds to the object the mouse is positioned over. In the difference to the info panel offered by VisGenome in Chapter 5, the second version of the application has the option of scaling and labelling, and therefore, two new buttons for the functions appear in the info panel.

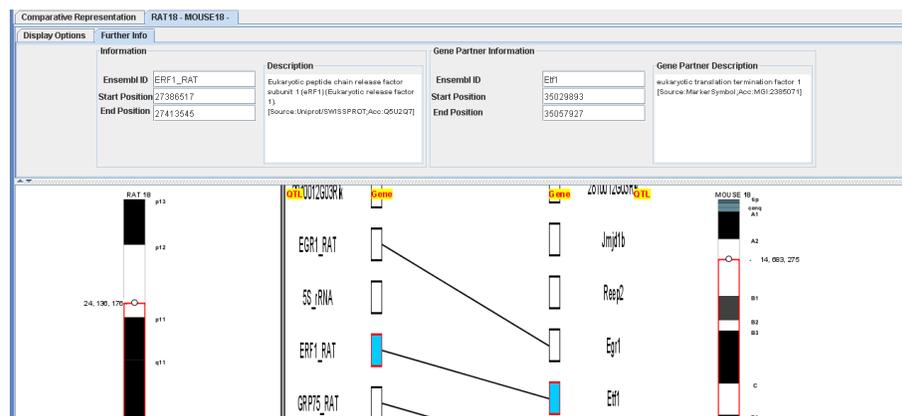


Figure 7.5: The comparative representation for the rat chromosome 18 and the mouse chromosome 18. The data is scaled by the scaling algorithm which makes all genes the same size and QTL size depends on genes. Genes ERF1_RAT and Etf1 are linked by a homology line and marked in blue.

```

1 CartoonPlus() {
2   for(gene in GENES) {
3     ResizeAndPaint(gene)
4     ScaledMarkersBetween = GET_MARKERS_BETWEEN()
5     for(each marker from ScaledMarkersBetween)
6       ResizeAndPaint(marker)
7     ScaledMicroArrayProbesBetween = GET_MICRO_ARRAY_PROBES_BETWEEN
8     for(each micro_array_probe from ScaledMicroArrayProbesBetween)
9       ResizeAndPaint(micro_array_probe)
10    ScaledMarkers = GET_MARKERS_IN()
11    for(each marker from ScaledMarkers)
12      ResizeAndPaint(marker)
13    ScaledMicroArrayProbes = GET_MICRO_ARRAY_PROBES_IN()
14    for(each micro_array_probe from ScaledMicroArrayProbes)
15      ResizeAndPaint(micro_array_probe)
16    ScaledQTLs = GET_QTLS_FOR_GENE()
17    for(each QTL from ScaledQTLs)
18      if (QTL.end>D AND QTL.end<=B)
19        ResizeAndPaint(QTL)
20        delete(QTL from ScaledQTLs)
21  }
22 }
23 GET_MARKERS_BETWEEN() {
24   for(marker in MARKERS)
25     if(marker.start>=D AND marker.end<=A)
26       markers.add(marker)
27   return(markers)
28 }
29 GET_MARKERS_IN(){
30   for(marker in MARKERS)
31     if((marker.start<=A AND marker.end>A) OR (marker.start>A))
32       markers.add(marker)
33   return(markers)
34 }
35 GET_QTLS_FOR_GENE(){
36   for(QTL in QTLs)
37     if(QTL.start>D AND QTL.start<=B)
38       QTLs.add(QTL)
39   return(QTLs)
40 }

```

Figure 7.6: CartoonPlus algorithm. Hierarchy of object sizes: chromosome \geq QTL \geq gene \geq marker and micro array probe.

7.2.7 Scaling Algorithm

We developed a scaling algorithm for arbitrary genomics data. SyntenyVista [46] offers scaling for genes only in a comparative representation. We offer scaling for all data in both single and comparative representations, see Figure 7.5 and 7.4. A user chooses the basis for scaling and then other elements are scaled in relationship to the chosen data type. In the current prototype we chose genes as a basis, so we scale all genes to the same size. An extension of this work is to allow the user to change the basis for scaling interactively. The algorithm looks at other types of data which are smaller or larger than genes, such as markers, micro array probes and QTLs, and scales them with relation to genes. We divide all elements smaller than genes into two groups: elements which are in a gene region and elements which are in the region between two genes, see Figure 7.4.2.

For each type of data holding items smaller than the basis, we create a column with elements which are situated within the gene boundaries, and a second column containing elements which are positioned between two genes. For all elements which are in the gene region we choose the same size for each element,

and the same applies to all elements which are in the area between genes. The size of the elements depends on their number in a gene region. This means that if in an area of a gene there is only one marker, it has the same height as the gene, but if there are 10 markers, they together have the same size as the gene (each marker is set to 1/10th of gene height). When an element is on a gene boundary, it is partially in a gene region and partially between two genes, and we situated it in the gene region. We also scale elements like QTLs which are bigger than genes. We look where a QTL begins and ends and we paint it starting at the gene where it begins and ending at the gene where it finishes. The solution allows us to present clearly a homology between genes in a comparative representation and additionally to show relations between micro array probes, markers, genes, and QTLs in two species.

Figure 7.6 outlines the scaling algorithm. All genes, markers, micro array probes and QTLs are stored in hashtables. The algorithm iterates over all basis objects, here genes (line 2). First, we scale markers and micro array probes which are between genes (the previous gene and the current one), see Figure 7.4.2, object m1 between G1 and G2. Then we scale markers and micro array probes with a start coordinate before the gene and end coordinate inside the gene or start coordinate inside the gene region, see Figure 7.4.2, objects m2 and m3, and Figure 7.6, lines 4-15. Then we place QTLs which begin inside the gene region or in the region between a previous gene and the current gene, see Figure 7.4.2, object Q1. For each gene we check as well where the end coordinate of a QTL is, and depending on this we paint the element. In the pseudo-code we show function `ResizeAndPaint` which for basis data gives all elements the same size. For small objects, such as m1, m2, or m3, function `ResizeAndPaint` calculates how many elements are in the gene area or in the area between genes, and divides the area by the number of elements and then the elements are painted in the calculated size. For large elements `ResizeAndPaint` calculates the height of the elements to span from the beginning of the gene where the QTL starts, to end of the gene where it ends. If a QTL begins or ends between genes, the function takes ending of previous gene or beginning of the next gene as its coordinates.

In Chapter 2 we examined the visualisation techniques used in genome browsers, and recognised that a number of tools used in biological research implement well known visualisation techniques, but only a few experiment with new techniques. `CartoonPlus` adds a novel extension to the array of available solutions. It can be used in single and comparative representations. In a single representation the users can see all data scaled, depending on a chosen basis, which allows them to see clearly which micro array probes and markers are related to a gene. In a comparative representation the scaling makes homologies between genes clearer. Figure 7.7 presents how scaling algorithm is working for the single representation. We see that on Figure 7.7.A all genomic data is in original size. On the other hand, on Figure 7.7.B all genes have the same size and other data is scaled depending on genes. The solution allows the precise localisation and positioning of small elements according to bigger biological data. For example, in the column “MDNoSalt” we have two micro array probes: 1387665_at and 1391417_at and in the column “genes” we have two genes `Bhmt` and `LOC365972`. Looking at Figure 7.7.A it is difficult to say if the two

micro array probes are in the gene's regions or not. Figure 7.7.B clearly shows that micro array 1387665_at is in gene Bhmt's region and micro array 1391417_at is not in gene LOC365972's region. We suggest that CartoonPlus helps the biologists to localize their data easily and precisely present the position of the elements and relationship between them. We are going to test this in our mixed paradigm user study presented in Chapter 8.



Figure 7.7: CartoonPlus: A shows data in original size. B presents scaled data. All genes have the same size and the rest of the data is scaled depend on genes.

Among all genome browsers we studied, only SyntenyVista [46] uses a scaling algorithm, however it was used only in a comparative representation and only for genes. The solution we used is novel and it could be useful not only in genomic data but also in different fields of biology and medicine which use one linear scale for many types of objects. In next chapter, ‘Mixed Paradigm User Study’, we test the new technique in an experiment with biological researchers who use a combination of data from Ensembl and their own lab experiments.

7.3 Conclusion

The chapter presented the second version of VisGenome which extends the first version of the application (see Chapter 5) with new functionalities. In the previous chapter we presented the idea that our users

had some problems during the tasks execution. In T1 and T3 they became lost in the zooming option, therefore we developed *Focus On*, which allows them to easily focus on the element the biologists are interested in. We also observed that the labelling confused the biologists and did not precisely show all the micro array probes with descriptions. Therefore we expanded the existing labelling to allow the users to see all the labels or selected subset from the labels. We recognised that supporting data from Ensembl was very helpful for genes, therefore we decided to expand supporting data from Ensembl to all kinds of data available in VisGenome and Ensembl. We added colours because biologists wanted to see their data coloured and also a new scaling algorithm, which should help biologists to see details of their data. We expect all the improvements to help the users to navigate around their data and make it possible for them to do their work more effectively.

Next chapter shows how we used VisGenome in our mixed paradigm user study. We test how the users interact with the new features, especially how they use the new scaling algorithm CartoonPlus, but also if they use the improved techniques. We conduct the mixed paradigm user study which is different from our first user study to discover that which was not possible to see before, i.e. how the users interact with the different tools available during their daily work and find factors which influence the use of the applications. We want to identify the best features in VisGenome and other tools used by biologists. We also want to identify the worst features in the tools to eliminate them.

Chapter 8

Mixed Paradigm User Study

This chapter describes the second user study conducted with the medical researchers from the BHF Glasgow Cardiovascular Research Centre. During the experiment we used the second version of VisGenome (VG) described in the previous chapter. This chapter presents the experimental design and introduces its results. Work presented in the chapter was the last stage in my PhD research, therefore in the following chapters we discuss, present plans for future work, and conclude the thesis.

8.1 Introduction

During the second experiment we studied the usability of VisGenome and other genome browsers and tools used by the biologists. We wanted to identify the best features in VisGenome and other tools used by biologists in their work. We did not propose any specific tasks but allowed the users to work with the available tools and observed them during their work. We were interested in finding out how they interact with different sets of tools, if they use only some of the tools to carry out their tasks, or if they perform the same tasks using the tool which supports the particular task. We were also interested in how they learn to use VisGenome and how the use of VisGenome changes in time. We gave them a version of VisGenome containing their biological data and asked them to use it for at least two weeks. After the end of this period, as we discovered, some of them were still using the tool. We advised the participants that I would spend with each of them about 2 hours in total observing their work and recording their activity. During those two hours I ‘shadowed’ them and recorded some of their activities using a video camera. However, I also met with some of them before or after the experiment, so I had the occasion to observe them for more than the two hours. I attended their Thursday meetings where they exchange information about their research progress. I also contacted them a few times to offer technical support and help with VisGenome. At the end of the experiment, I interviewed them about the usefulness of

visualisation techniques they used.

The main aims in the mixed paradigm user study are:

- discover features which were not possible to observe during initial quantitative user study, such as interaction with different sets of tools, which allows us to better understand the user
- find other factors which influence the use of VisGenome
- test CartoonPlus which is new visualisation technique
- check if the visualisation techniques improved after the initial quantitative user study are more effective

8.2 User Study

This experiment aimed to identify the most important application features, with respect to visualisation techniques, in VisGenome and other tools used by biologists in their work. We expected to find the best features which help the biologists, and the worst ones which are also important and need to be improved. These aims differ from our previous experiment in terms of methodology. The results are also more detailed than the results from the previous ‘Initial Quantitative User Study’ described in Chapter 6.

8.2.1 Participants

There was no pilot study or users from the BRC this time, in contrast to the ‘Initial Quantitative User Study’. We asked five people (one female and four males) from the BHF Glasgow Cardiovascular Research Centre to take part in the experiment. Subjects were named with a letter: **D**, **M**, **W**, **J** and **L**. They all hold a PhD and have significant experience with biological data. We divided them into three groups: **D**, **M** and **W** as one group and **J** and **L** as two singletons, based on the nature of their work.

- **D**, **M** and **W** work together with the same rat data in the same project. They enjoy using Windows. They use the mouse instead of the keyboard where possible. Their database knowledge is very limited. They often use MS Office and Access. However, collecting data in Access and querying is done by somebody else and not by themselves. The subjects only look at reports and the results. These three users often work with animals and use specific applications to manage animal data.
- **J** is a statistician and his experience with genome browsers and computer applications is different from that of the other users. Only he uses Linux and prefers this operating system. In his work at the BHF Glasgow Cardiovascular Research Centre he was persuaded to use Windows. He uses a number of applications which run under Windows via Linux emulation. He prefers using LaTeX or

R [149] to tools offered by Windows (Word or Excel). His habits are a bit different as well. He uses the keyboard whenever he can, as he knows many key short cuts. Everything he does is done with great precision and attention to detail.

- **L** works mainly with people from other biological centres around the world and his main interest is human genetics. He spends a lot of time in the laboratory and uses the computer mainly for e-mailing and looking at biological results from his experiments. He has some knowledge of databases and uses Access and R for statistical calculations.

Two of the users, **M** and **W**, took part in the first experiment presented in Chapter 6 - 'Initial Quantitative User Study'. However, all of the users had been shown how to use VisGenome before the experiment. To make sure that they were familiar with the tool, on the first day of the experiment I showed them all options offered by VisGenome. Because **D**, **M**, **W**, and **J** share the same office and were available at the same time, I gave a small presentation to all of them at the same time. I installed and ran VisGenome on their computers and showed all options offered by VisGenome. I showed examples of how to use the application. I made sure that everybody knew how to carry out the example tasks. **L** wanted to get a demo at a different time, therefore I showed him VisGenome separately. I demonstrated all options offered by the tool and at the end I asked him to execute example tasks to be sure he knew how to use the application. Some of the users made notes during the presentation. They were also familiarised with help offered by VisGenome. The presentation took up to one hour.

8.2.2 Methodology

We began the experiment by asking each user to sign a participant consent form which can be found in Appendix J. We asked the users to use VisGenome for two weeks at least. They all began the experiment on 10.09.2007. However, some of them took holidays or attended conferences during this time, so their participation in the experiment was not continuous. The last user officially finished the experiment on 07.11.2007. The majority of users are still using VisGenome to view data from their experiments.

During the experiment we used a variety of information gathering techniques: video and voice recording, questionnaire, paper diary, and log file recording, as detailed in the following sections.

Video and Voice Recording

We used video recording only for two participants (**J** and **W**) who do not carry out animal experiments, due to ethical reasons, see Figure 8.1 and Figure 8.2. We also used voice recording for **L**, who carried out laboratory work with genetic material without animals and stressed that video recording could disturb his work. The participants were asked to behave as normal during the whole experiment and do their



Figure 8.1: One of the participants prepares graphs using Ingenuity Pathway Analysis [136] using data from his experiments. The image was captured using a video recorder.

work as if I was not observing them. However, when they saw the video camera, their behaviour did not seem to be natural. They tried to explain to me everything they did, and led me through why they used the selected tools or applications. When a student or another worker from the Centre appeared in their office, very often the first thing they informed the new arrival about was the video camera.

Interviews with all participants were recorded by voice recorder for later analysis. I felt that the participants' behaviour seemed to be more natural during voice recording than during video recording.

Questionnaire and Diary

Before the experiment I gave each user a paper diary in which they could record notes on VisGenome. I asked them to write a note each day they used the tool, any positive or negative observations, and any other suggestions about the tool and the whole experiment.

At the end of the experiment we asked the participants to fill in a questionnaire which asked



Figure 8.2: One of the participants prepares samples of human genetic material which will be used in an experiment. The picture was captured using a camera. We see **L** in his laboratory space. He is placing genetic material into probes, using adjustable pipettes. He measures the amount of the material in microliters, depending on guidelines received from his cooperators. After preparing the samples, he sends them to his collaborators in London.

them to tick the applications and tools they used or add any tools not listed. The list initially included: VisGenome, Ensembl, Excel, DSI Acquisition [127], Rat tail blood pressure determination [145], Word or other text editor, Outlook or other e-mail browser, and Internet browser. There were also blank rows for participants to add information on other tools they used. I observed during the experimental period how they used VisGenome together with other applications. Then, I asked the participants to answer the following questions for each application they used.

Q1. What kind of visualisation techniques do you use in this application?

Q2. How frequently do you use the application with its visualisation techniques (1-daily, 2-weekly, 3-monthly, 4-rarely, a few times per year, 5-never)?

- Q3.** What kind of data do you use in this application?
- Q4.** Is the application useful?
- Q5.** Did you succeed in using the application?
- Q6.** Do you have any additional comments about the application?

The participants could add names of other visualisation techniques in Q1.

Log Files

We added a logger to VisGenome which recorded in a log file the type of user action carried out by the participants, the time of action, the context (single or comparative representation) and the animal type (human (H), rat (R), or mouse (M)). Every time a user ran VisGenome, a new log file was created, see Figure 8.3. We asked the users to create at least 5 log files during the two week experimental period.

```

2007-09-10 10:53:36 SR [RAT, 2]
2007-09-10 10:54:24 SR - PAN
2007-09-10 10:54:48 SR - FocusOn Affy - 1392627_x_at
2007-09-10 10:54:52 SR - FocusOn Affy - rc_AI071051_at
2007-09-10 10:54:55 SR - FocusOn Affy - rc_AA892512_at
2007-09-10 10:55:23 SR scaling: original size -> the same size
2007-09-10 10:56:17 SR labelling: selected labels -> all names
2007-09-10 10:56:47 SR - PAN
2007-09-10 10:57:07 SR - Set Region Position by Button - Chromosome Region Start - 210000000 - [RAT, 2]
2007-09-10 10:57:34 SR scaling: the same size -> original size
2007-09-10 10:58:10 SR - ZOOM

```

Figure 8.3: The example taken from log file D_2007-09-10 10 53 17.log. In the log files SR stands for single representation and CR for comparative representation.

Observation

Observing the users was the most important part of the experiment. I spent at least two hours together with each participant. Moreover, I attended a number of meetings in the BHF Glasgow Cardiovascular Research Centre, which provided more opportunities to observe the users.

8.3 Experimental Results

8.3.1 Overview

We analysed video and voice recordings, the questionnaires, the diaries, the log files and handwritten notes on my observations. The data gathered from the log files is presented in Appendix P. In this chapter we used abbreviations listed below.

SR - single representation

CR - comparative representation

C - colour change (colour an object)

DR - drag region (on the chromosome icon)

FO - focus on

LL - labelling (switch between two labelling modes - all labels or selected labels)

LK - link to Ensembl

P - pan

PS - pan session - pans reduced to sessions (see section 8.3.1 - 'Overview')

S - scaling (CartoonPlus)

RS - set region (set chromosome region for navigation using info panel)

Z - zoom

ZS - zoom session - zooms reduced to sessions (see section 8.3.1 - 'Overview')

H - human

M - mouse

R - rat

During the study we wanted to understand how the users used VisGenome and other tools in their work and to identify good and bad application features. We also wanted to test our new CartoonPlus algorithm and see how improved visualisation techniques (tested in the initial quantitative user study) are used by the biologists. We recorded all user functions in VisGenome because we wanted to adapt it to the biologists' needs. We recognised that although the users were offered a number of different visualisation techniques, the most frequently used ones were the simplest and most common techniques such as *zooming* and *panning*, see data in Appendix P. Therefore, we believe that a good implementation of such simple interaction techniques is very important. VisGenome uses *panning* and *zooming* implemented with the Piccolo toolkit [4], but other genome browsers have completely different kinds of *zooming* and *panning*. Because of these differences, we decided to record *zooming* and *panning* as iteration steps. By an iteration step we understand the smallest unit which is used during *panning* or *zooming*. However, during the analysis we reduced all the iteration steps to sessions. By a session we understand a period of time when the user used only one kind of function continuously. For example, this happened when user **D** used *focus*

on and then *panning* for about 2 minutes and after this again *focus on*. During the 2 minutes of *panning* user **D** made 160 iteration steps in *panning* but it was only one continuous *panning session*. We do not distinguish between the user pressing a mouse button only once and then moving the mouse or pressing the button five times to do *panning* during the 2 minutes. The reduction from iteration steps to sessions was made only for *zooming* and *panning*, because other functions do not require this. The data presented here for *panning* and *zooming* as iteration steps is called *pan* or *zoom*, see Figure 8.7. On the other hand, session data are named *pan session* and *zoom session*, see Figure 8.6.

As can be seen from Figure 8.4 and Figure 8.5, each of the users used VisGenome in a different way. When we consider all *zoom* and *pan* iteration steps, it is clear that **J** has the highest number of all functions together but when we reduced *zoom* and *pan* to sessions, we see that **L** is the one with the highest number of functions. The participants executed 89,004 actions all together (*zooming* and *panning* as iteration steps).

The participants were asked to create at least 5 log files, and **W** created 8 log files, **M** 6 log files, **L** 7 log files, and the others 5 log files. When we summarise the duration time for each function and each chromosome looked at in VisGenome, all users navigated in VisGenome for 11 hours and 36 minutes during the experiment (the total over all participants). This is a long time, especially that half of the time was spent in laboratories and during the time spent in the office they used a number of different tools and carried out other tasks as well. **W** used VisGenome for the longest time: 3 h 8 min, **L** had the shortest time, 1 h 34 min. Four participants used VisGenome during the two first weeks after the beginning of the experiment and only one, **J**, took part in the experiment with a break - because of a holiday and a workshop. **M**, **D**, and **W** are still using VisGenome every time they finish their experiments with rats, which means a few times in every two weeks with a break for another two weeks when they make the experiments.

When we look at the types of actions performed by the participants (see Figure 8.6 and Figure 8.7), it is clear, especially with the version with all *zooming* and *panning* iteration steps, that the two functions are the most popular and common in use during the experiment. We notice that nobody used *set region* during the experiment in the comparative representation. There are two main reasons why this happened. First, the users prefer to *drag* the region, because they know where on a chromosome the region of interest is situated and they do not always remember the coordinates or do not want to enter the numbers. This avoids some of the errors found in the initial quantitative user study (entering wrong numbers). Second, in the case of **J** who helps biologists but does not conduct any experiments himself, he does not know exactly what he is to look for in the comparative representation. Instead, all homology regions are important for him. Therefore, he often looks for genes in a homology area or for other data, and the *drag region* option is much better in his situation than *set region* where a user has to know the region coordinates.

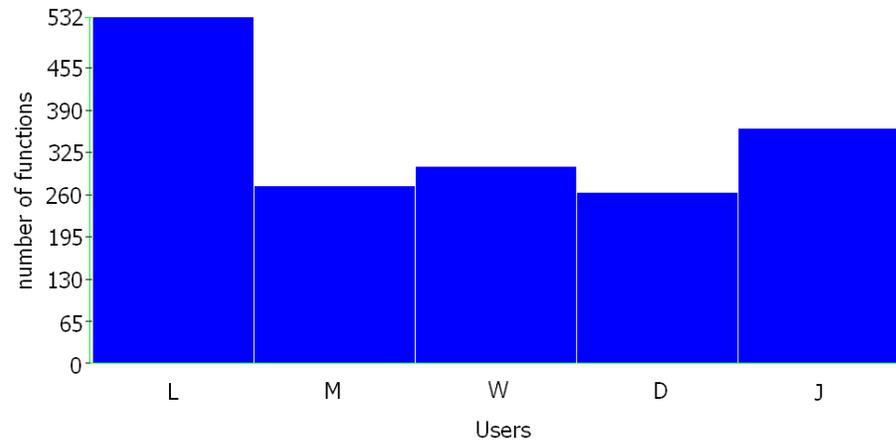


Figure 8.4: All functions recorded in log files in VisGenome - *zoom* and *pan* reduced to sessions. The figure was created using Replayer [76].

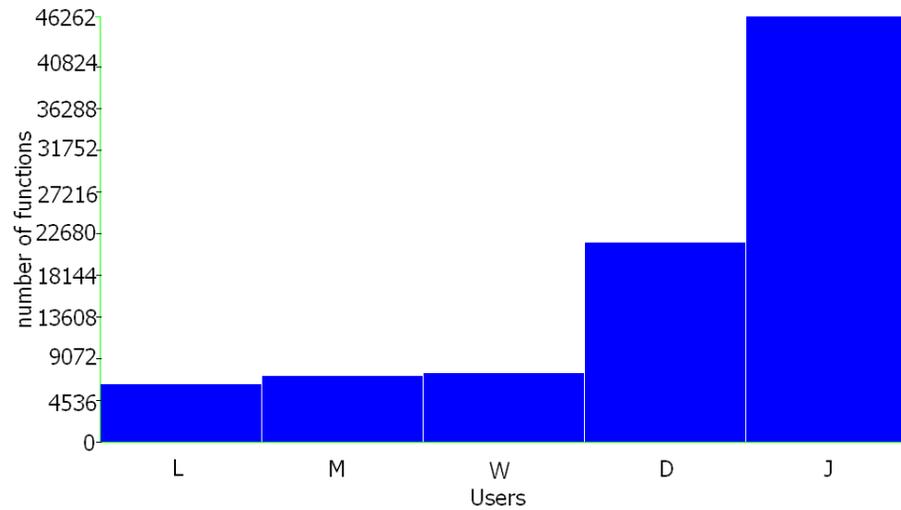


Figure 8.5: All functions recorded in log files in VisGenome - *zoom* and *pan* as iteration steps.

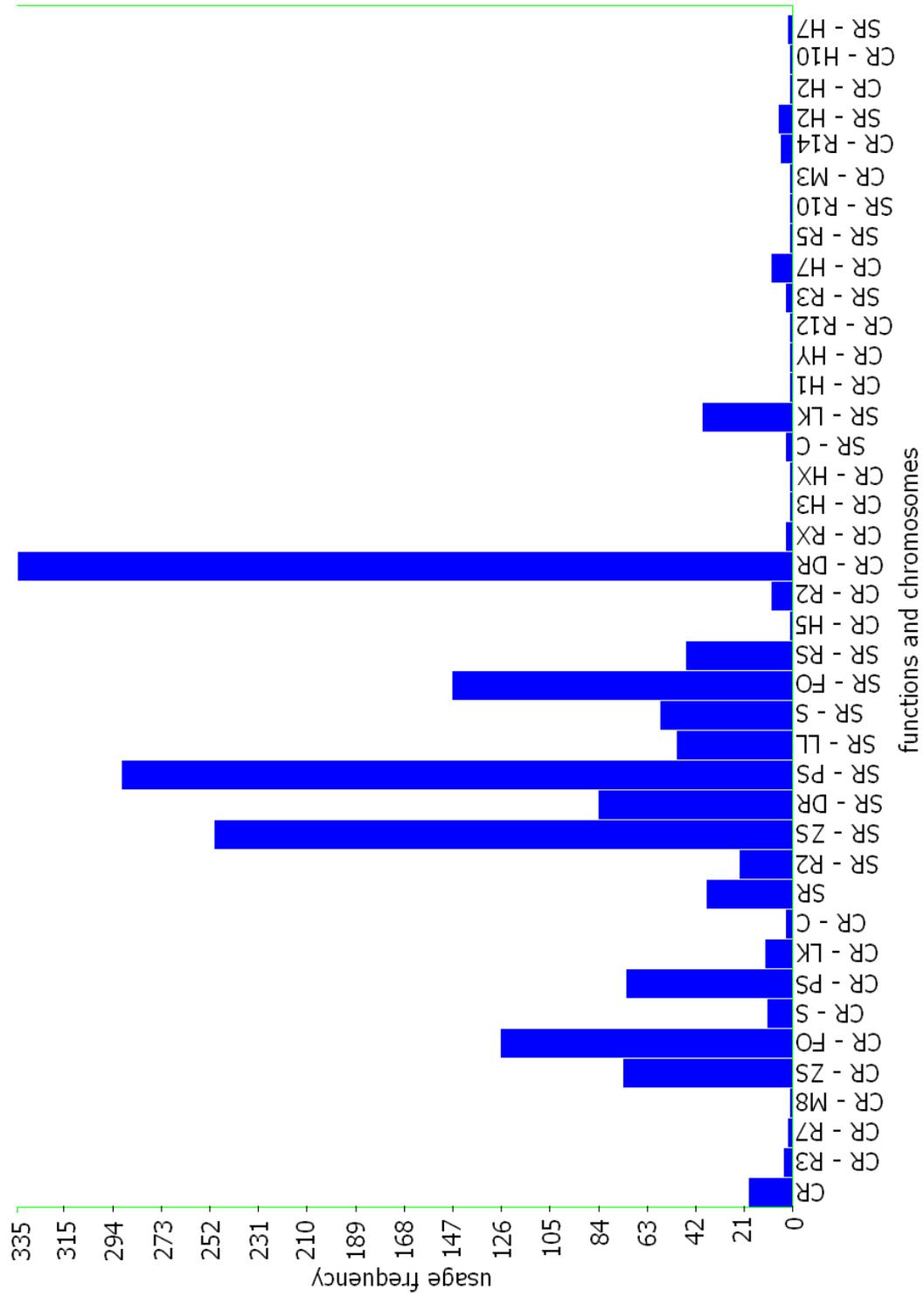


Figure 8.6: Overview of all functions and chromosomes used in VisGenome - *zoom* and *pan* reduced to sessions, totalled over all users. *Drag region* in the comparative representation is represented by the tallest bar. The next two tallest bars represent *zooming* and *panning* in the single representation.

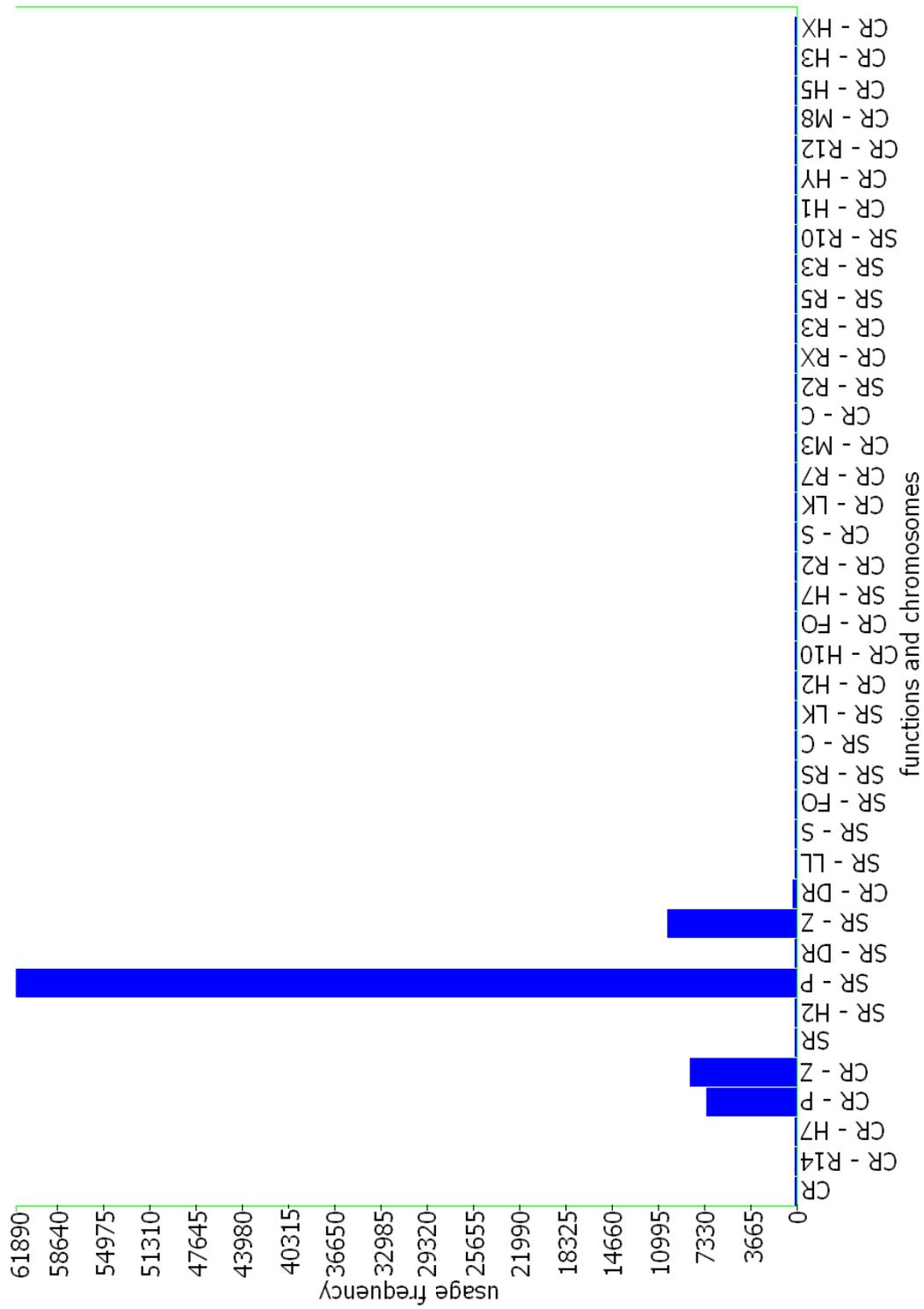


Figure 8.7: Overview of all functions and chromosomes used in VisGenome - *zoom* and *pan* as iteration steps, totalled over all users. The four tallest bars represent *zooming* and *panning* in the single and the comparative representations.

In the comparative representation, nobody used *labelling* because this option is available only in the single representation.

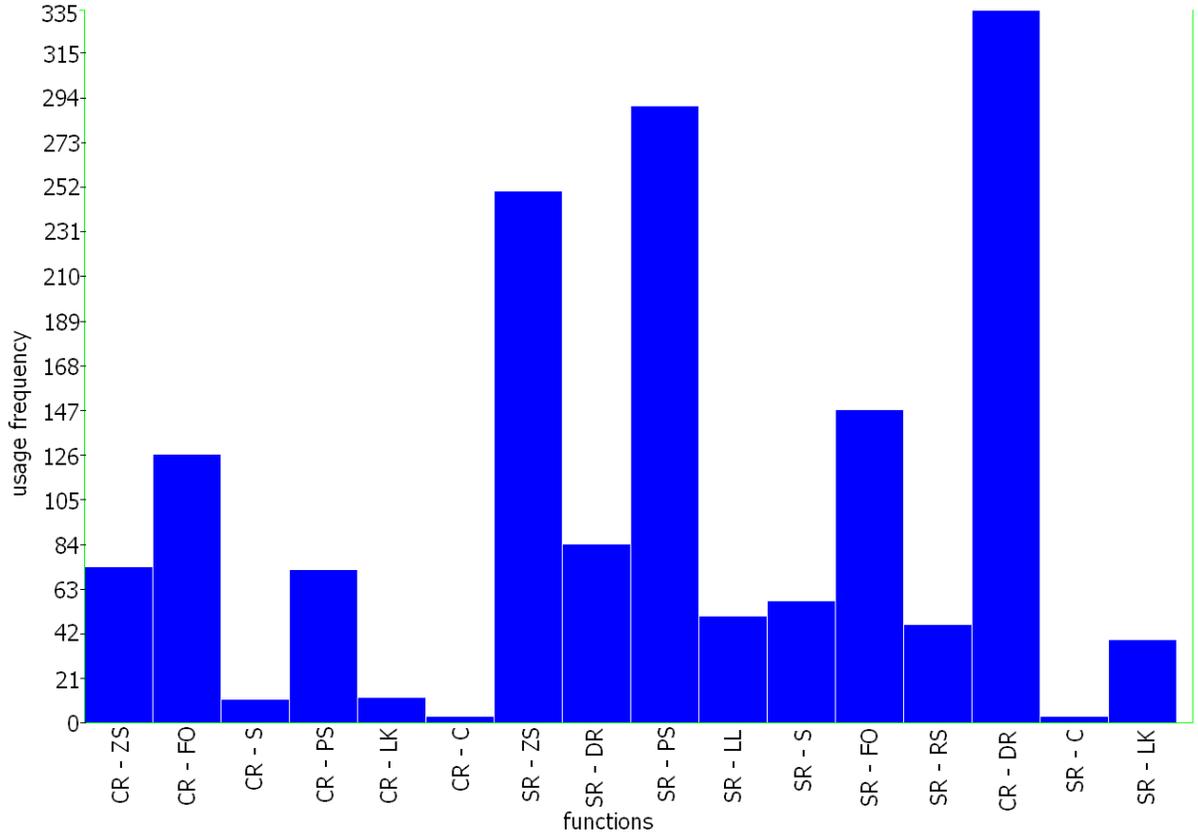


Figure 8.8: Count of all functions recorded by log files in VisGenome - *zoom* and *pan* reduced to sessions, totalled over all animal chromosomes and users.

It is notable that *colouring* was the least frequently used function, see Figure 8.8. We found two main reasons for this. First, the users received their data coloured as they wanted, therefore, for some of them, it was not necessary to re-colour it or add colours to non-coloured data. Some of the participants, including **J**, for whom the *colouring* option could be an excellent solution, forgot that the option was available. **J** remembered about *colouring* at the beginning of the experiment, but after a break he forgot about it. Please note that before the experiment we demonstrated all options available in VisGenome to the participants. Some of the subjects took notes during the presentation, but some of them repeatedly forgot about some features and asked me repeatedly when I visited them. They had permanent access to the user manual (see Appendix G) and the help information offered by VisGenome.

As shown in Figure 8.9, the participants did not look at all chromosomes for all three species, but only at the chromosomes of interest to them. The most popular was the rat chromosome 2 in the single representation. Four participants (**D**, **W**, **M**, and **J**) were interested in data for this chromosome.

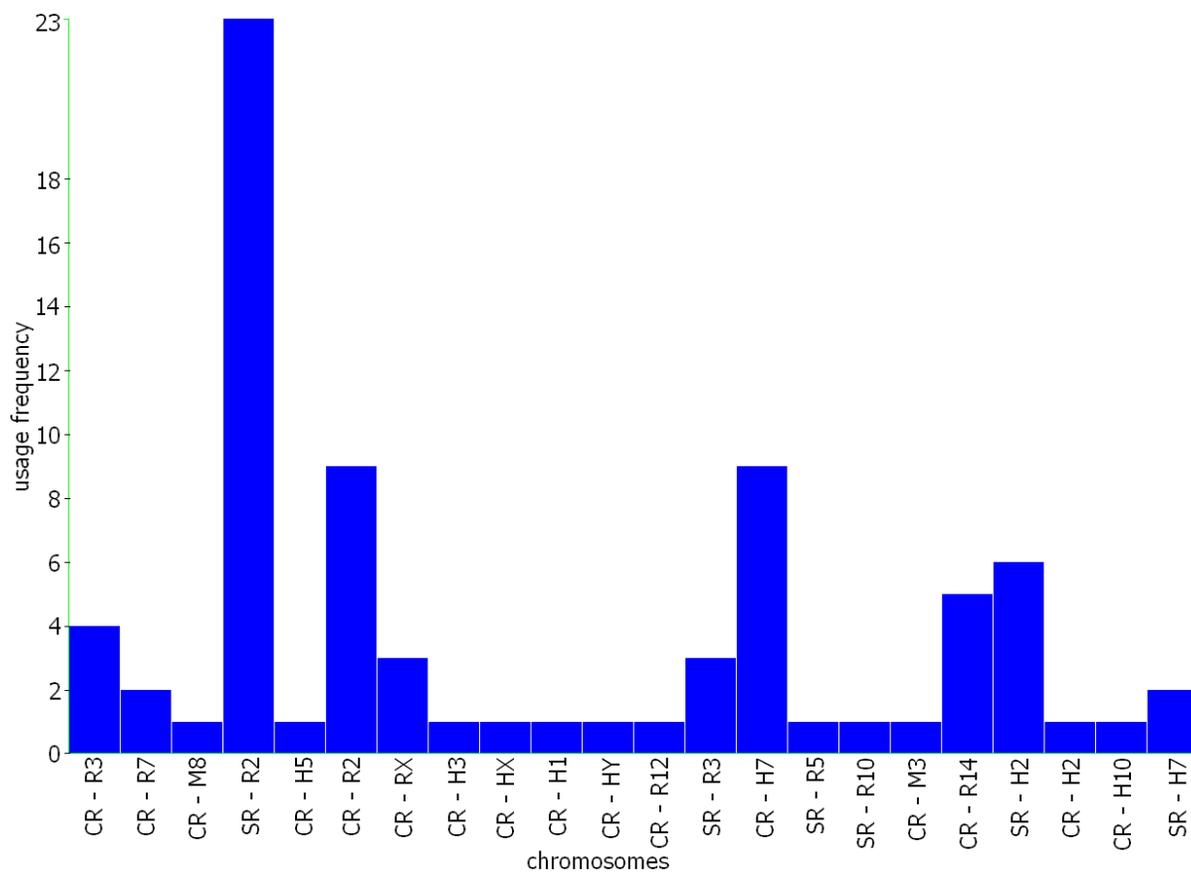


Figure 8.9: Frequencies of usage for all chromosomes viewed by the participants, by representation, totalled over all users.

In the following, we present the results for each of the users in more detail. Because we had only 5 participants we present qualitative analysis instead of statistic analysis.

8.3.2 Log Files Results

Log files recorded from VisGenome usage show that *pan* and *zoom* were used more often overall than other functions. Some functions were used rarely - *focus on* and *link* - and in particular *colour*, which was mainly used once by each of the users. The participants mainly used *zooming* and *panning*, even if the functions were reduced from iteration steps to sessions. Using a feature often does not always mean that it is good, and could show that there are problems with the interface. However, as can be seen from our observations, biologists used some functions a number of times, because very often they looked for nothing explicit. In this situation, *pan* and *zoom* usage gave the most effective and quickest results.

8.3.3 User Questionnaire Results

The questionnaire contained 6 questions (Q1-Q6, see page 129) for each application used by the participants and a number of free slots where the participants could add the names of other computer applications they used. We included VisGenome, Ensembl, Excel, DSI Acquisition [127], Rat tail blood pressure determination [145], Word or other text editor, Outlook or other e-mail browser, and Internet browser. The users added the following applications: Graphpad Prism, Ingenuity Pathway Analysis [136], Access, Power Point, R [149], and Minitab. In the questionnaire we asked about visualisation techniques available in each application, however, we also left room for user suggestions. As it turned out, VisGenome and Ensembl [44] were the only two genome browsers used during the experiment.

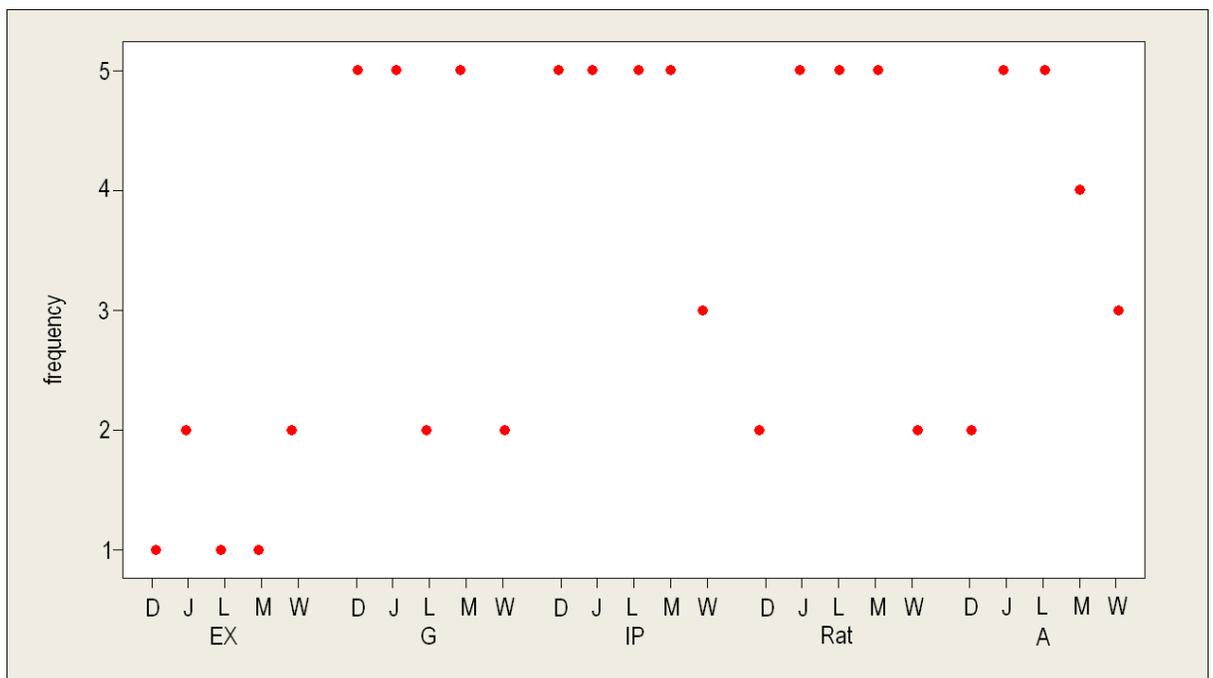


Figure 8.10: The results for frequency in using applications during the experiment. The abbreviations are: EX - Excel, G - Graphpad Prism, IP - Ingenuity Pathway Analysis, Rat - Rat tail blood pressure determination, and A - DSI (Data Sciences International) Acquisition. Frequency corresponds: 1 - daily, 2 - weekly, 3 - monthly, 4 - rarely, a few times per year, and 5 - never.

We observed that Word, Outlook, and internet browser are used by all participants daily. Figure 8.10 shows the frequency of use for some of the applications. The differences in usage frequency do not express user preferences but reflect their work needs.

We observed that in each application the participants used almost all existing techniques. On the other hand, some of the tools offer techniques which are not available in other tools. In Q4 and Q5, we observed that the participants usually succeeded in the task execution and the applications they used

were useful for them.

When we look closely at the comparison of the two genome browsers used during the experiment, we see that user **L** was the only participant who found VisGenome only partially useful and did not succeed in his tasks. The other users found the tool useful and fully succeeded in their tasks. When we asked the user **L** about the reason for his failure, he mentioned problems with finding the data. He wanted to see more data for human which VisGenome did not offer him. We stress here that we asked all the users before the experiment about the data they would like to see and prepared a version of VisGenome with their data. The data problem found by user **L** is a general problem not only found in VisGenome. QTL data for human is not available in Ensembl, and there is no obvious source where it can be found. **W** was the only user who found Ensembl to be partially useful. He succeeded in his tasks also only partially with it. The rest of the users were fully satisfied from Ensembl. Overall, participants who used an application claimed that they found it useful and succeeded in the tasks they wanted to carry out.

These claims of utility and success were made despite apparent high error rates (see Chapter 6 - earlier study).

8.3.4 Video Recording

We used video recording only for two participants (**J** and **W**) who do not carry out animal experiments and agreed to participate in the recording.

- During the video recording, **J** worked with his computer where he tried to install applications necessary to use Ensembl via Perl. I asked him, before the recording started, to behave as if I was not there. However, I was asked questions about the application installation. A few of his comments were left without my response, however I did not want to be rude and therefore I helped him in his work. A number of different voices were also recorded, such as **J**'s colleagues' conversations, printing machines, machines from the laboratory which is opposite **J**'s office, or people walking in the hall. During the recording, **J** focused on Perl installation for Windows, as he had tried a few days earlier to install Perl in Linux environment (Cygwin) without success. He needed this application, together with MySQL, and Ensembl libraries for Perl, to use Ensembl via Perl. He had attended an Ensembl course a few days before and wanted to use his knowledge in practice.

During the work **J** uploaded necessary installation files and began installing them. He read aloud everything he had to do, therefore we could hear: "Yes, so...", "Yes, that is fine", "Do I agree?", "Yes, that is probably a good idea.", "Of course, it sounds fantastic.". When a problem occurred, he spoke with me, he asked me to explain terms such as DBD, DBI, and Visual Studio. He has a good knowledge of commands in Linux, but he had trouble with DOS commands for Windows. He would say out loud what he is typing at the command line, therefore, we could hear "backslash

home, backlash Cygwin, backlash”. He also commented on his work: “I think I have this.”, “I don’t know what happened.”, “I don’t think it is gonna work.”. During the work **J** made a few spelling mistakes and thought aloud about what happened and why something he expected to work was not working (for instance he typed *test.pl* instead of *test1.pl*). He felt uncomfortable when for a longer time I or he did not say a word. Then, he began talking and asked: “It is helpful for you? I mean, this what I am doing now?” or just began talking about his two children and wife Helen. During his work he used the keyboard quite frequently, and the mouse only in situations where it was not possible to use the keyboard. It is also obvious that **J** is a very precise person and wants to do everything perfectly.

- User **W** was also recorded on video. He was preparing a presentation for the following day meeting and all the time he used Graphpad Prism to prepare graphs representing his findings. **W**, similarly to **J**, behaved unnaturally during the recording, however, he spoke less. At the beginning he explained that he has data from rats which had been fed fructose for two weeks. After his the animals received an injection of glucose and he measured their blood stream for next two hours. Then he began his work and spoke no word to me for about 6 minutes. I observed that each of the workers from the BHF Glasgow Cardiovascular Research Centre has a notebook and writes down everything important in it. **W** does the same, therefore, he had to transfer all data from his notebook into Graphpad Prism. During that time he was eating an apple, joking about some part of his work, and speaking with his coworker (rat project manager). His supervisor looked at his data, explained things that were not clear to **W**, and they were planning more experiments with three animals. After this, he made some comments (to himself) about his data - it was obvious that he was disappointed and then he told me that he is not quite happy because of this experiment. He expected different results, and the ones he obtained were in contrast to the one he previously obtained. All the time other people kept coming into the office and **W** spoke with them. He informed them about the video recording, told them about his findings, and helped with printing (each office is equipped with a black and white printer, but his office has an additional colour printer and everybody who wanted to use it had to collect their printing in **W**’s office). The user expressed his strong liking for Graphpad Prism. He mentioned a few times that he tried to persuade people in his team to use the tool, because it is better and more efficient than using Excel, Minitab or R. **W** did not speak as much as **J**, and when he spoke, he addressed his coworkers or himself. A few times he spoke to me, to inform me about his work. **W** stressed that the rats from his last experiment (see last column with data - *WKY* in Figure 8.11) become more fat inside than other rats, and this stresses that being fat is genetically determined. He also mentioned that in some research labs the scientists make rats exercise to keep them fit.

During the experiment **W** left briefly to work with animals, which was not recorded.

During **W**’s work we observed that he was not using keyboard shortcuts, however, he mentioned

that he uses them, but in Graphpad Prism his colour printer settings do not allow for this.

- **L** was not recorded on video, however, he was recorded using the voice recorder. During the observation he prepared probes for his collaborators in London (see Figure 8.2), which took more than one hour. After this, I made a suggestion that this is a perfect job for a robot, and asked why they do not use medical robots they have in the hospital instead. He responded that he did not feel as a robot, and a robot does not do the work as precisely as he does. He decided to show me all the new and old equipment use for large-scale sample analysis (see Chapter 4).
- **D** carried out work with animals which is also described in Chapter 4 and no voice or video recording was taken.
- **M** carried out a lot of supervisory work. I spent a lot of time following after him from one building to another in the hospital and observing his meetings with his PhD students and coworkers whom he instructed about work.

8.3.5 Interview After the Experiment

During the interview after the experiment the participants stressed that they prefer Outlook to WebMail offered by the BHF Glasgow Cardiovascular Research Centre (**D**, **W**, and **M**). **J** preferred Thunderbird to Outlook. As mentioned before, the users, especially **M**, **W**, and **D**, used a mouse everywhere they could, instead of the keyboard. This is easy in Outlook, whereas WebMail does not allow one to easily drag e-mails between folders. User **D** complained about Access. The participant stressed that using the application is difficult as only one user knows how to enter new data. **J** complained about Ensembl: he wants to know how to use its full potential. The user took part in a few workshops and courses about Ensembl and still does not know where to find all interesting data. **J** also stressed that he does not like Excel and uses it as little as possible. He also prefers LaTeX and WinEdt to Word. The other users use Word for everything they have to write, such as papers, notes, or reports. **M** complained that VisGenome has trouble visualising items of data which are far apart, but he also complained about Ensembl. He stressed that Ensembl is not as useful as it could be and it is very difficult to visualise large genomic regions. **M** praised internet browsers which allow the users to interact with other applications by using Java Web Start [138].

As can be seen, the majority of biologists spend only a small part of their time interacting with computer applications and do that to see their laboratory results. Therefore, it is very important to provide for them a genome browser that can visualise their laboratory results and allow easy navigation. VisGenome connected to other tools, such as Excel, allows them to carry out data entry more easily (see Chapter 10). On the other hand, they can still find useful data in other sources, such as Ensembl, and then visualise the data in VisGenome.

8.3.6 User Diary Results

At the beginning of the experiment we gave paper diaries to the participants and asked them to note what they found to be positive or negative in VisGenome, and write down any suggestions related to the tools they used and the experiment itself. The diary was completed regularly by four participants, only **J** did not complete it regularly.

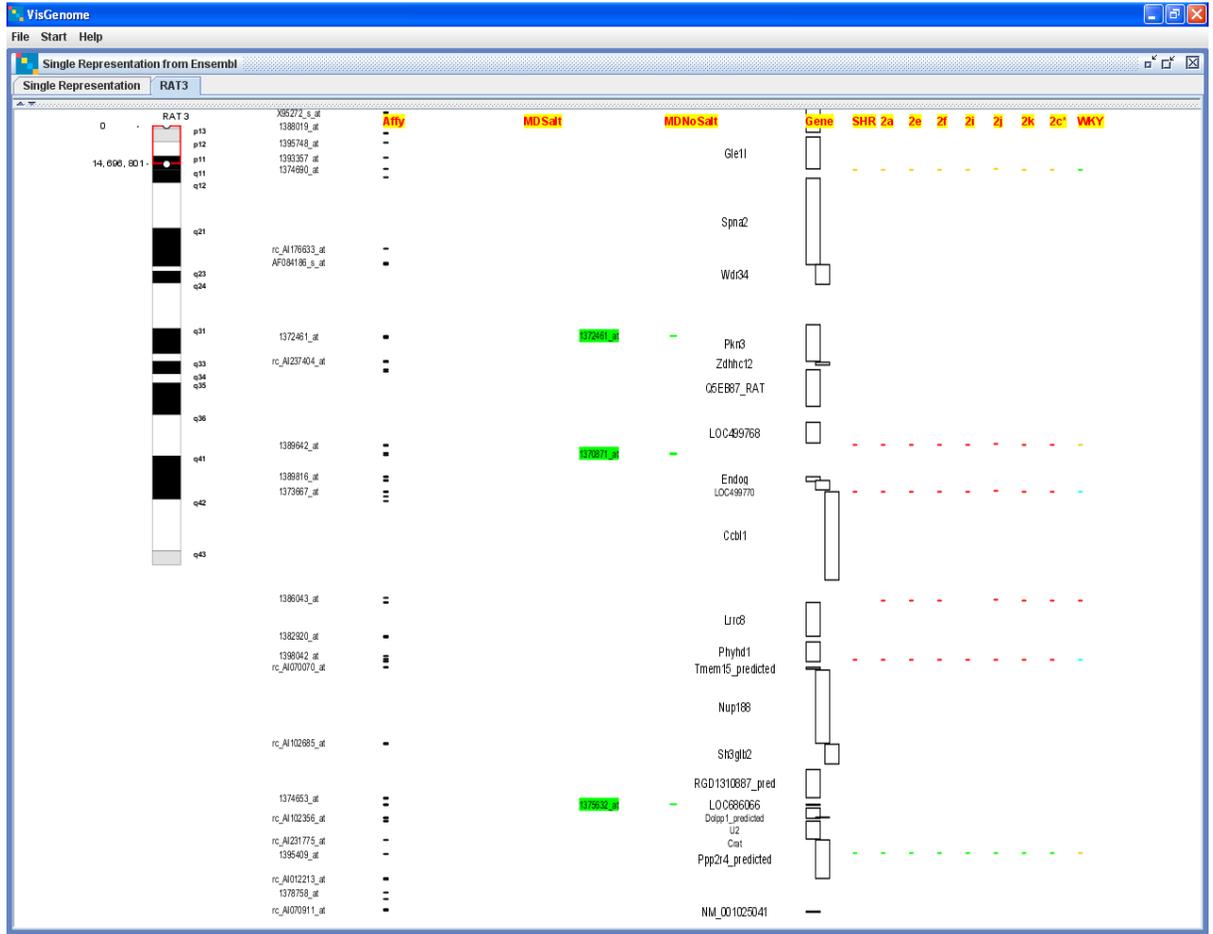


Figure 8.11: VisGenome - version 2 - used in the experiment. The tool presents SNPs data for 9 animals (SHR, 2a, 2e, 2f, 2i, 2j, 2k, 2c*, and WKY). The data is coloured depending on whether A, C, T, or G appears in the SNP. The solution helped the users to find the differences between the species quickly.

The participants were very happy to have coloured SNP data, especially for rat chromosome 3 (see Figure 8.11). This benefit was stressed by the majority of users. They liked the fact that *scaling* allowed them to visualise “significant” (the word used by **M**) micro array probes and SNPs. This functionality was not available in other tools. They were able to check which genes were polymorphic in their rat strains. Some of them recognised that their data which they expected to be similar behaved differently, depending on which biological experiment they were looking at. They discovered that gene **Tmsb4x** is

apparently differentially expressed in parentals. The participants were happy that in the comparative representation they were able to mark a particular part of a chromosome and view it. The users stressed that the possibility to see an Ensembl web page via a link from VisGenome was very helpful for them and they liked it.

The participants also found a number of features which could be improved in the future. One as mentioned previously, is having to remember the fact that to change colour they have to press Alt and a mouse button, and to link to Ensembl they have to press Shift and a mouse button. They suggested a button list in the info panel instead of the current solution. They would like to have a choice of columns with data which could be changed and replaced at any time so as to see only some rat strains at a time (like in Excel). One user (**W**) found VisGenome quite “laggy”(slow). He also would like to have an option to select and copy on-screen elements into PowerPoint. Some of the users stressed that when they marked a number of data items, VisGenome was working quite slowly and they would like to see a progress bar which tells them how quickly the tool is working. They were used to using a mouse scroll wheel, and would prefer to use it in VisGenome instead of the current solution for *panning* which requires *dragging*. **L** had a number of suggestions related to the data. He wanted more data for human QTLs and preferred gene data to be coloured by the genetic material it represents (if it is pseudo gene, protein coding, snRNA, or miRNA). He also would like to see a summary which would tell him how many elements of each category are in a view. He also stressed that for him VisGenome offers too much choice for selecting a region of interest in a chromosome. He especially did not like the option *set region* which required pressing *Set Position* button. During the trial we recognised that the majority of participants forgot about pressing the button to set the position. After that, they often recognised that nothing had happened, and used *drag region* instead.

8.3.7 Observations

The participants enjoyed using VisGenome. The users used the single representation almost twice as often as the comparative representation. **J** and **L** used the two representations with similar frequency. **M**, **W**, and **D** used the single representation much more frequently than the comparative representation. These three participants (**M**, **W**, and **D**) work in the same project, and were interested in micro array probe sets and SNPs in a chromosome more than in the homologies between chromosomes.

The participants were instructed that they do not need to open a new window with a single or comparative representation when they want to see another chromosome. They could use an already opened view and just add a new tab. However, only **J** used the option. The other users always opened a new window for each chromosome they wanted to navigate. This did not change the quality of the view, in both situations they see things properly. However, when a user opens two different windows, VisGenome allows him to set them close to each other and see both at the same time. No user used this possibility.

They always switched between the windows instead of viewing them side by side at the same time.

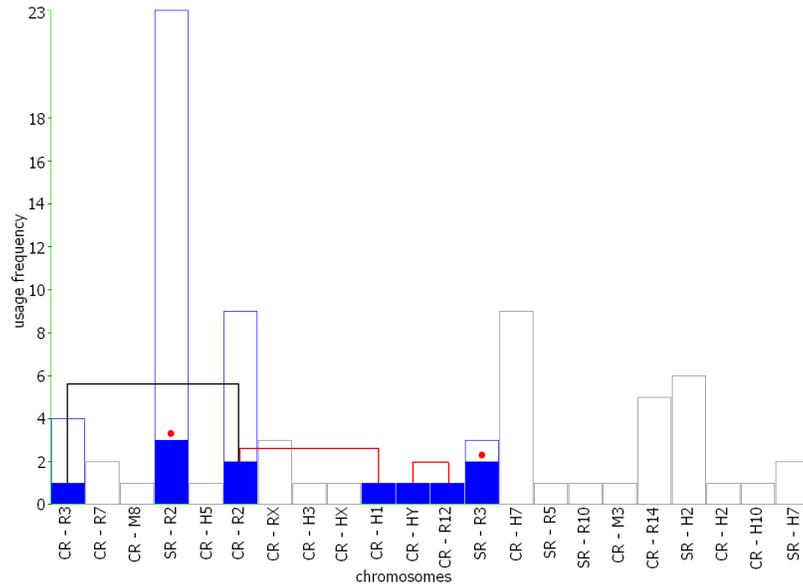


Figure 8.12: Chromosomes viewed by **D** (CR - comparative representation, SR - single representation). The white area represents chromosomes viewed by all participants, the blue area - chromosomes used by **D**.

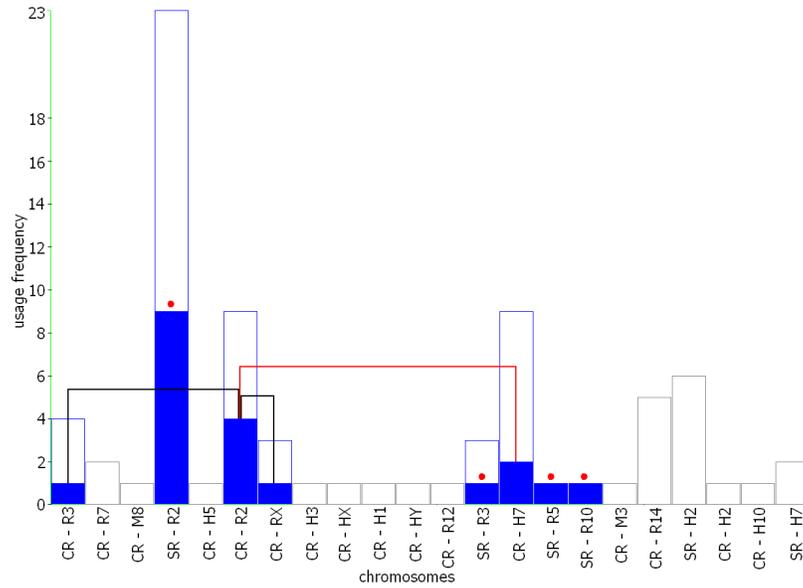


Figure 8.13: Chromosomes viewed by **W** (CR - comparative representation, SR - single representation). The white area represents chromosomes viewed by all participants, the blue area - chromosomes used by **W**.

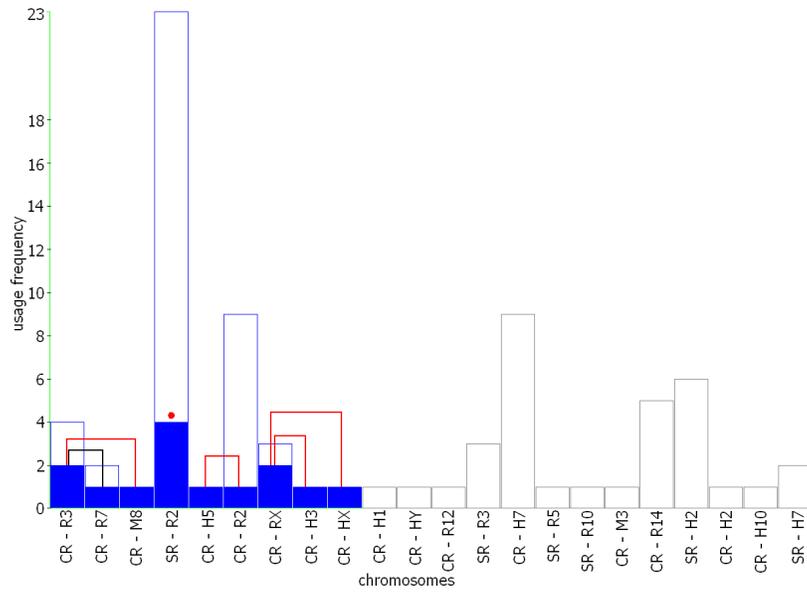


Figure 8.14: Chromosomes viewed by **J** (CR - comparative representation, SR - single representation). The white area represents chromosomes viewed by all participants, the blue area - chromosomes used by **J**.

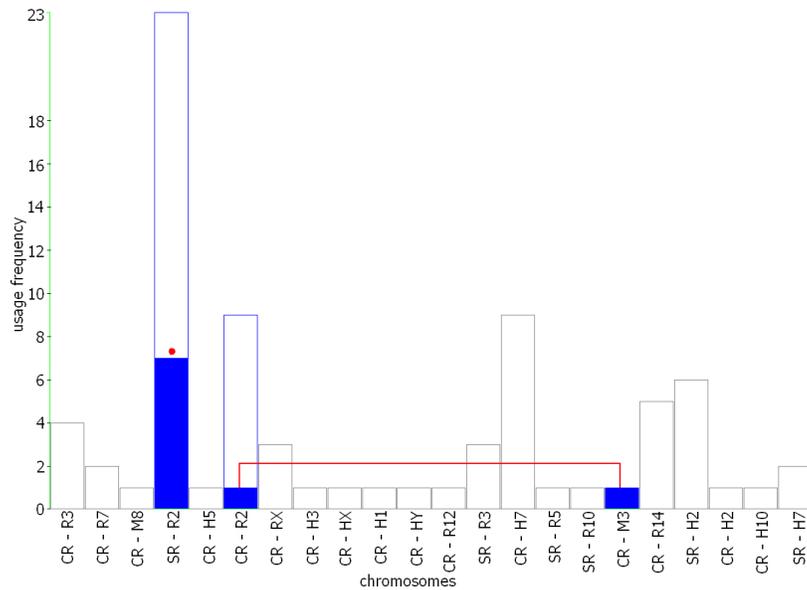


Figure 8.15: Chromosomes viewed by **M** (CR - comparative representation, SR - single representation). The white area represents chromosomes viewed by all participants, the blue area - chromosomes used by **M**.

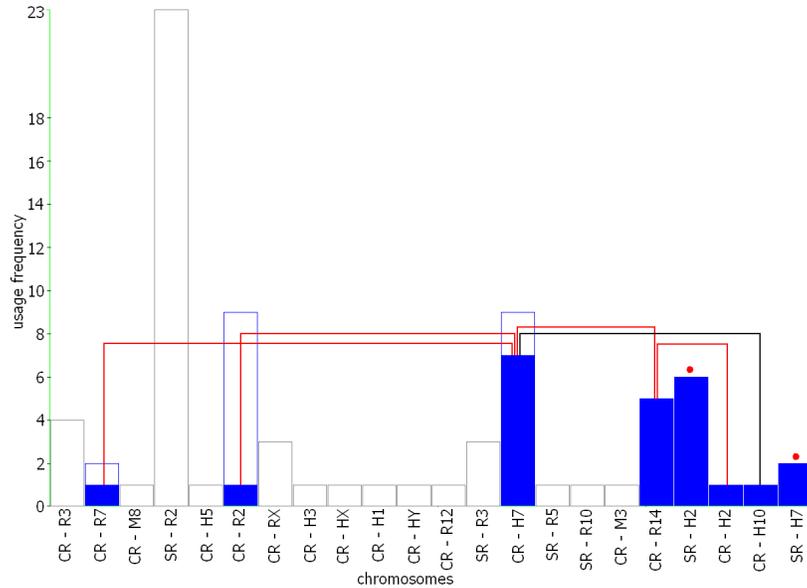


Figure 8.16: Chromosomes viewed by **L** (CR - comparative representation, SR - single representation). The white area represents chromosomes viewed by all participants, the blue area - chromosomes used by **L**.

User **D** viewed the rat chromosomes 2 and 3 in the single representation (see red dots in Figure 8.12) and comparisons between the human chromosome 1 and the rat chromosome 2, and the human chromosome Y and the rat chromosome 12, see Figure 8.12. The participant looked at the comparison between the rat chromosomes 2 and 3 probably accidentally, because there is no homology between chromosomes from the same species. This interpretation is backed up by fact that the user only opened the comparative view with the wrong chromosomes and did not navigate it at all.

User **W** viewed the rat chromosomes 2, 3, 5, and 10 in the single representation (see red dots in Figure 8.13). He looked also at the comparison of the human chromosome 7 and the rat chromosome 2. Similar to **D** he made mistaken comparisons for the rat chromosome 2, first to the rat chromosome 3, and then to the rat chromosome X.

User **J** viewed only the rat chromosome 2 in the single representation (see red dots in Figure 8.14) and the comparison of the mouse chromosome 8 and the rat chromosome 3, the human chromosome 3 and the rat chromosome X, the human chromosome 5 and the rat chromosome 2, and the human chromosome X and the rat chromosome X, see Figure 8.14. The user also made one mistake and tried to look at the comparison between rat chromosomes 3 and 7.

User **M** viewed the smallest number of chromosomes. He viewed only the rat chromosome 2 in the single representation (see the red dot in Figure 8.15) and the comparison of the mouse chromosome 3 and the rat chromosome 2, see Figure 8.15. User **M**, as the only participant, made no unnecessary

comparisons between chromosomes for the same species as made by the other users. A possible reason for that is that he had previously used VisGenome and asked me to add to VisGenome his own data before the experiment. He also took part in the experiment comparing Ensembl and VisGenome [53, 52].

User **L** was the only one who viewed the human chromosomes 2 and 7 in the single representation, see red dots in Figure 8.16. The other users viewed only rat chromosomes singly. He also compared the human chromosome 7 with the rat chromosome 14, the human chromosome 2 with the rat chromosome 14, the human chromosome 7 with the rat chromosome 2, and the human chromosome 7 with the rat chromosome 7, see Figure 8.16. Similarly to other users, he made a mistake of comparing the human chromosome 7 with the human chromosome 10.

8.3.8 Summary

The experiment shows that users have different needs with respect to the data they want to study, as dictated by their research question. **M** and **W** viewed the rat chromosome 2 in the single representation most frequently. However, the rat chromosome 2 was also interesting for **D** and **J**, but not for **L** - he looked at it only to compare it to the other chromosomes. **L** focuses on a different set of chromosomes. He looked at human chromosomes and their comparison with rat chromosomes.

The users quite often made a mistake and looked at a comparison of two chromosomes from the same species. Usually, they quickly recognised their error and changed to another view. After choosing a first chromosome in the comparative representation, VisGenome highlights other chromosomes which have homologies with the one already chosen. The users found this function very useful. The experiment showed that the application should also block the possibility of choosing a chromosome from the same species as a second chromosome in this particular research context. The reason why they made so many mistakes is that they often chose a first chromosome and then they consult Excel or Word to be sure which other chromosome to choose. Then they activated a VisGenome window and clicked at a chromosome which is the closest to the mouse cursor, by accident. They clicked in the window instead of the frame. In some situations, they made a different mistake, as they wanted to see a single representation for a chromosome and they clicked on it in the comparative representation instead. One person was simply interested to see what happens if he clicks on a chromosome from the same species.

The participants used almost all functions available in VisGenome, see Table 8.1, except for *set region* in the comparative representation (see Figure 8.17, 8.18, 8.19, 8.20, and 8.21). However, they behaved differently, depending on what they were looking for and how they activated it.

L and **J** used *drag region* a number of times. The participants, especially **J**, very often looked for nothing explicit. They wanted to know, in general, what data are available in a region or in the comparative view they wanted to see homologies between genes. They were not sure where on a chromosome they have

Table 8.1: Functions available in VisGenome. All functions are described in Chapters 5 and 7.

	VisGenome Functions
1	zoom
2	focus on
3	scaling
4	pan
5	link
6	colour
7	drag region
8	labelling
9	set region

to look for the data. Therefore, they just dragged the mouse through region after region on a chromosome and tried to find interesting elements on it. However, their individual techniques for using VisGenome were completely different, as shown by their use of *panning* presented as interaction steps and sessions. **L** often looked for QTLs and **J** for SNPs. There was only a small number of QTLs and they are quite large. Therefore **L** had no problem with finding them quickly. **J**, on the other hand, wanted to see SNPs in detail and often used a number of *pans* and *zooms* before he found the data. On the contrary, **M** performed the smallest number of *drag regions*. He knew exactly his region of interest on the rat chromosome 2 and, after opening a view in VisGenome showing the chromosome, he entered the coordinates and used *set region* to directly locate to the region of interest.

Colour was hardly used by the participants. As mentioned in section overview, the participants quite often forgot about the function or did not need it because the data were already coloured. Twice *colour* function was invoked but no colour was chosen (**L** and **M**). The users just wanted to check how

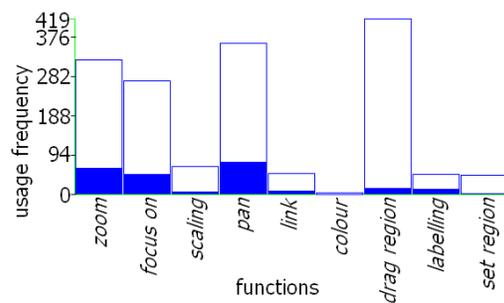


Figure 8.17: Functions used by **D** during the experiment. The white area represents functions used by all participants, the blue area - functions used by **D**.

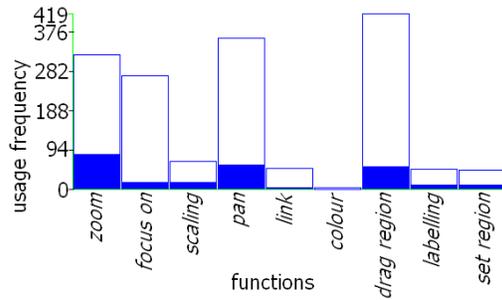


Figure 8.18: Functions used by **W** during the experiment. The white area represents functions used by all participants, the blue area - functions used by **W**.

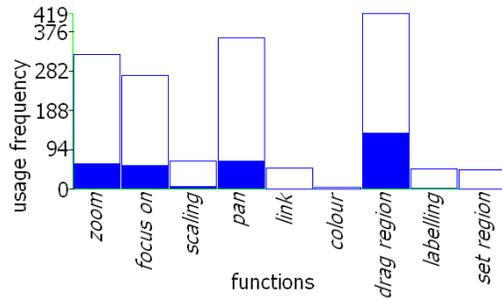


Figure 8.19: Functions used by **J** during the experiment. The white area represents functions used by all participants, the blue area - functions used by **J**.

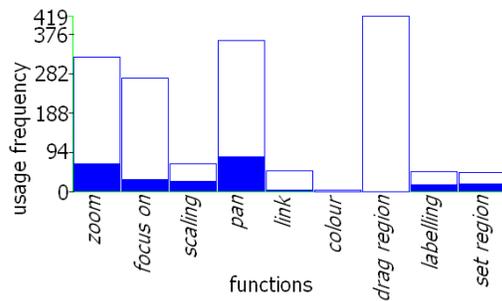


Figure 8.20: Functions used by **M** during the experiment. The white area represents functions used by all participants, the blue area - functions used by **M**.

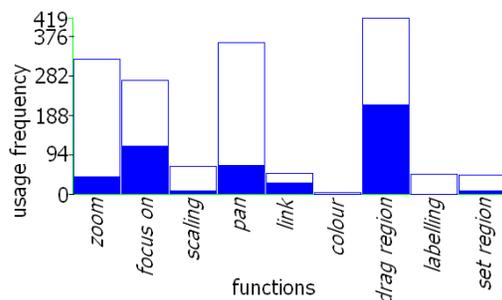


Figure 8.21: Functions used by **L** during the experiment. The white area represents functions used by all participants, the blue area - functions used by **L**.

the function worked or selected it accidentally. When a participant changed colour, it was selected from the green-blue range.

Link (to Ensembl) was used quite frequently, especially by **L** who was interested in human genes. Only twice was *link* used for micro array probe data (1393390_at and 1384691_at). Genes ENSG00000105926 and ENSG00000170264 were the most frequently viewed items via a link to Ensembl.

Focus on was used even more frequently than *link*. The participants used the function mostly for genes, with few exceptions. Users **D** and **J** focused on some micro array probes which are also available in Ensembl. **J** and **W** focused on their micro array data which I had been asked to add to VisGenome before the experiment. **W** also looked at SNP data in detail. **L** focused especially on human chromosomes and QTLs.

D, **M**, and **W** used *labelling* quite frequently, compared with the other users. *Labelling* is implemented only in the single representation for micro array probes. The three participants were interested in micro array probes from Ensembl, and compared them to the coloured micro array probes from their experiments. *Labelling* allowed them to see the differences between the micro array probes more clearly in detail. Ensembl offered them a nice overview of micro array probe sets (see Figure 8.22). However, in the last few versions Ensembl changed this representation, and now micro array probes are represented as a table (see Figure 8.23). The biologists do not like the new “visualisation”, but Ensembl motivated the changes by a structural improvement, to make queries faster.

On the other hand, Ensembl implements an algorithm checking the relationship between micro array probe sets and genes (the small blue triangle in Figure 8.22). If at least 50% of micro array probes from the micro array probe set (on a chromosome) are in an enlarged gene region, Ensembl assumes that the micro array probe set is in the gene region. The users could easily find out if a micro array probe set is in a gene region in the previous version of Ensembl. On the other hand, the majority of participants were not interested in the Ensembl algorithm. They wanted to know if any micro array probe was in a

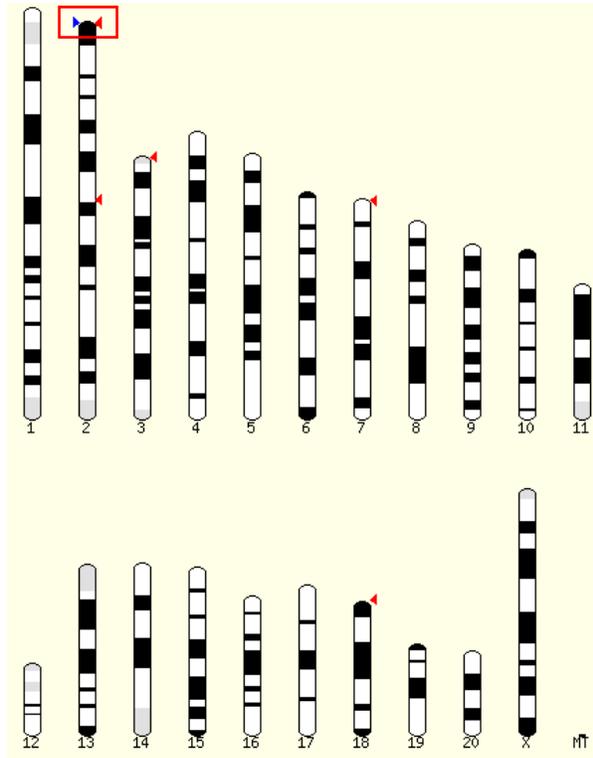


Figure 8.22: Old representation of micro array probe set 1396288_s_at for rat from Ensembl. The red square marks micro array probe set 1396288_s_at on the rat chromosome 2 on band q11, see also Figure 8.23. Only one gene (ENSRNOG00000040342) is correlated with the micro array probe set according to Ensembl algorithm. The gene is marked by the blue triangle.

gene region. To check this for micro array probes from Ensembl, they have to do this manually for each gene and micro array probe, which takes a lot of time. Also, they can not do this easily for their own experimental data. Ensembl offers the option to place private data, but this seems to be an impossible task for the medical researchers from the BHF Glasgow Cardiovascular Research Centre, and nobody knows how to do this. I tried to use the option but I have not managed to succeed yet. Partly because of this, the medical researchers suggested that they would like to see in VisGenome single micro array probes rather than only micro array probe sets. In the end, I suggested to them that a solution (in the future) could be to show all micro array probes but also highlight other micro array probes from the same set via colour or other UI feature.

The participants found *scaling* to be very useful, especially **W** and **M**. They used the function to see if an SNP is in a gene region. Quite frequently, especially when the SNP is close to a gene, it is difficult to say if the SNP is in a gene region or only close to it. *Scaling* made the SNP big enough to disambiguate its position.

Affy rat230 2	Band	Ensembl Gene ID
1396288_s_at	q11	ENSRNOG00000040342
1398871_at	q11	ENSRNOG00000030606
1398871_at	q11	ENSRNOG00000034271
1371297_at	q11	ENSRNOG00000033945
1398871_at	q11	ENSRNOG00000030836
1386252_at	q11	ENSRNOG00000009997
1388356_a_at	q11	ENSRNOG00000009997
1399161_a_at	q11	ENSRNOG00000009997
1389191_at	q11	ENSRNOG00000010278
1397070_at	q11	ENSRNOG00000010353
1391422_at	q11	ENSRNOG00000010353
1373409_at	q11	ENSRNOG00000010702
1388217_at	p13	ENSRNOG00000040334
1382386_at	q11	ENSRNOG00000005306
1387362_at	p13	ENSRNOG00000005223
1387584_at	q11	ENSRNOG00000006120
1377793_at	q11	ENSRNOG00000040329

Figure 8.23: New representation of micro array probe sets from Ensembl. The red rectangle marks micro array probe set 1396288_s_at on the rat chromosome 2 on band q11, see also Figure 8.22. If a user wants to find all elements from micro array probe set 1396288_s_at, he has to click, one by one, on all elements from the list called 1396288_s_at.

Panning and *zooming* were essential. All participants used the two interaction techniques and found them to be useful. However, the majority of them would change *panning*. They would use a mouse scroll wheel instead of moving the mouse up and down.

Ensembl offers both a large amount of data and a number of search options. The participants very often looked for something using Ensembl searching. Sometimes they found interesting information in Ensembl, and sometimes Ensembl provided the data from other sources, such as RGD [144] and OMIM [143]. When the participants knew exactly where to look for the data, they came back to VisGenome and looked at a particular part of a chromosome. Some of the biologists used their notes from Excel or Word, where they looked for a special data item or coordinates, and when they found where the element is situated on a chromosome, they came back to VisGenome. They stressed that they like visualisations offered by VisGenome and are happy that the application shows their own data alongside the data from Ensembl.

The participants changed their application use slightly over the course of the experiment. More experienced users, such as **M**, seemed to use VisGenome the same way all the time. However, we could see that the participants used *panning* and *zooming* much more at the beginning of the experiment than at the end. After the first week of using VisGenome, the majority of the participants knew what they wanted to look for and where the data is situated on a chromosome. They needed less time to look up

their notes or other sources, or they needed no notes. They knew more about interaction techniques offered by VisGenome when they used it frequently. On the other hand, they forgot about the interaction techniques they used rarely, like colouring.

Definitely, the participants used the comparative representation to a smaller extent in the second week of the experiment. **D**, **M**, and **W** used no comparative representation at the end, as they used only the single representation. The main cause for this was the character of their work. Most of their own experimental data were in the single representation, and at this stage of their work during the second week of the experiment, they were more interested in the relationships between different types of data in one species than the comparisons between two different species.

8.4 Conclusion

In the medical research context, the novelty of the visualisation techniques is not as important as their utility. This study is based on the view that the best way to find what visualisation techniques are the most appropriate for biological data is a user study with biologists working in their everyday environment. This chapter described an experiment carried out with biologists from the BHF Glasgow Cardiovascular Research Centre. The user study showed how biologists used a variety of computer packages to support their work. The experiment allowed the participants to use different applications and work in their natural environment, which let us understand more of their work, how they combine applications, and how they use VisGenome.

We recognised that appropriate visualisation techniques could save time and effort, and make work more effective. The most important techniques were simple interaction techniques, such as *panning* or *zooming*, which were most frequently used, and not techniques which are used only in special situations.

Our new algorithm: CartoonPlus (*scaling*) was found very useful, especially for small regions of data such as SNPs.

VisGenome was found useful and easy to use, however, mainly because of the lack of *searching* functionality, the application was used with other tools simultaneously. Where the participants knew a number of ways to do a task, they always chose the easiest way to do it. For example, if they used an internet browser and had a choice between using *scrolling* and mouse wheel, they always used the mouse wheel. They quite often repeated that I should implement the option of using the mouse wheel for *panning* in VisGenome. It was very difficult for them to believe that mice without scroll wheel still exist.

During the interview after the experiment, but also during the experiment, we received a number of suggestions how to improve VisGenome (see Chapter 10 - 'Future Work'). We hope further work with VisGenome will be done in the future.

According to biologists' suggestions and our observations, in the future we want to add or improve:

- connection between VisGenome and Excel - the biologists copied their data all the time from one tool to another
- uploading data into VisGenome - when biologists wanted see a new data they asked me to upload the data, it will be helpful if they can do this themselves
- movable columns and elastic windows - frequently biologists compare data from two columns, and the function is more effective when the columns are close each other
- micro array probes and micro array probe sets visualisation - the biologists mentioned a few times that they would like to see both data types
- speed optimisation - the users were a bit disappointed when they navigate a huge number of data and had to wait for rendering new views

We also want to improve CartoonPlus, we found *scaling* very useful and would like to extend it to different data types (as a basic). Another improvement which is worth to develop is a new layout (see necklace layout in Chapter 10). We want to see if it is more or less useful than standard linear layout used already for genomics data.

The next chapter discusses our research work.

Chapter 9

Discussion

This chapter discusses the research methods and findings. We discuss VisGenome with regard to other visualisation techniques. Next chapter of the thesis sketches plans for future work.

9.1 Weaknesses and Strengths of Both User Studies

The mixed paradigm user study was more difficult from the observer point of view, but it gave more meaningful results and confirmed the results from the first experiment. We observed that during both user studies the participants liked the techniques they know like *panning* and *zooming*, and those are much more important than other interaction techniques. We recognised that the participants learned new technology quite quickly and sometimes found it very useful, e.g. *scaling* during SNP navigation in the second experiment. During the initial quantitative user study we tested VisGenome and Ensembl separately. The mixed paradigm user study allowed us to see how the users interacted with both genome browsers and with other tools, which was not possible during the first experiment.

The initial quantitative user study allowed us to make quantitative measurements. We counted the number of mouse clicks and task duration. The second experiment allowed us to make only some quantitative measurements, such as event counts and duration for each function used in VisGenome, but we gathered qualitative information as well. In both experiments, the participants gave us a number of suggestions as to how we could improve VisGenome, and we heard their opinions about Ensembl, Excel, Access, and other tools they use and how VisGenome might work with them.

The different methodology used during the user studies influenced the whole experiment, i.e. the participants behaviour during the experiment, and the results. The initial quantitative user study was our first experiment and we made a few errors which we successfully avoided during the mixed paradigm

user study.

Different setups of both studies found different results, for example, during the second user study we recognised that it would be very helpful for the biologists to combine VisGenome with Excel, which was not possible to observe during the initial quantitative user study. The biologists copy their data all the time. First, they note information in their note books, frequently, from other biological tools such as Rat Tail Blood Pressure Measured (see Chapter 4). After this, they copy the notes from paper into Excel. Second, they frequently copy the information from Excel to Graphpad Prism, Access, or other applications. They also copy coordinates or an item name and put them into Ensembl search option. In my opinion, it would be a good idea to link VisGenome with Excel. The solution would allow the biologists to easily and quickly access their data and would eliminate errors occurring during copying the data.

Also both experiments give different answers for the question about suggestions. During the first experiment, the user suggestions were concerned more with visualisation used in data presentation, in the second user study they expected more biological data and had suggestions regarding the data layout used in VisGenome.

9.1.1 Weaknesses

It is obvious that there is a number of things which could be done better during the user studies. The biggest mistake made during the initial quantitative user study was the fact that the order of using VisGenome and Ensembl by the users was not random. The order we used (VisGenome first) should have favored Ensembl, however, due to our mistake, VisGenome achieved better results than Ensembl. We examined only 15 participants in the initial quantitative user study and 5 in the mixed paradigm user study. This was enough for our studies, however a larger number of participants could give us more details about their preferences and would be better for further analysis. The first user study was carried out in the laboratory space, and following the suggestions of other specialists in human-computer interaction. We changed this during the second study, where participants worked on their own PCs, in their offices for two weeks. Of course, the second user study duration could be longer. Again, this could give more results. During the mixed paradigm user study we used video recording, however, not all participants were recorded due to ethics constraints. In the future, longer user studies could be conducted and all participants could be recorded for a longer time. This would allow them to become more familiar with video recording (in the user study they always paid attention to the video camera) and could produce more naturalistic results.

9.1.2 Strengths

The largest benefit of our work was VisGenome evaluation. We knew that there are many different genome browsers, created by different research groups, and frequently used only for one biological project or a small subset of biological data (see Chapter 3). Only a few user studies were conducted in the field of biology and no studies of browsers have been published before. Our work shows that a user study may benefit biology. It is novel in the field of bioinformatics and worth doing, because in other fields people conduct user studies with success and they help them to create better computing solutions. Even our initial quantitative user study, where the most important findings were negative, showed that medical researchers make a number of mistakes and errors even when they are specialists in the area. We suggest that knowing this is highly valuable. Thanks to the studies we knew what problems arise in the use of genome browsers and we should create solutions that reduce errors in the future.

It is worth stressing that it was quite difficult to find access to right people who are specialists in the field and agree to be observed during their work. We cooperate with biologists from the BHF Glasgow Cardiovascular Research Centre and they agreed to take part in our user studies. The studies will benefit biologists in the long run, but also will improve our understanding of HCI. Our highly specialised power-users, biologists and medical researchers who took part in the experiment, could express their expectations and wishes with regard to VisGenome. We gave them a tool which helps them in their research. We have novel information on user activities, behaviour, daily work and needs which have not been known before in human-computer interaction. During the mixed paradigm user study, the participants could play with VisGenome, they did not feel observed, they could discover more things in VisGenome like the labelling, scaling or colouring. This contrasts strongly with the initial quantitative user study where they used the tool in an environment which was new to them, and they did specified tasks recorded by a screen recorder. This longer time in the second study allowed them to discover more contexts where they could use VisGenome and combine using our application with the tools they used already.

One of the most important benefits was to the medical researchers who took part in the experiment and could see their data in a new way. On the other hand, the users made mistakes due to human memory limitations. During the second user study, we introduced to them all VisGenome functions and showed where the user manual is available, however, they forgot about the *colouring* function and used it only when reminded about it. We also observed that even if the biologists feel to be experts at using a tool, they use only a small part of it and do not know all its functions (see Ensembl during the first experiment in Chapter 6).

The user studies allow us to better understand the real use of genome browsers and other software tools used in biomedical research. They also helped us to generate new design ideas which could be used in future work (see Chapter 10 - 'Future Work').

9.2 Visualisation of Genome Data

Not only the user studies could be improved, but also VisGenome. The application could work faster and allow the users easier data manipulation.

9.2.1 Genome Browsers

Visualisation of genome data is an important research tool in biology and medicine. There are a variety of genome browsers (see Chapter 3) which in practice should perform the same function - show the chromosomes of some species in detail. As discussed in Chapter 3, genome browsers could be divided according to three dimensions: number of species, object size, and complexity. It is not possible to present all existing genome browsers, therefore, in Chapter 3 we chose only the most popular genome browsers used by our collaborators, and genome browsers which are or were important in biology or history of genome browsers. We tried to show at least one tool from each category of genome browsers.

A biologist could use a number of genome browsers from each category, but this is not always possible. Some of the tools allow the users to see only some subset of their data, while other tools show all data, which makes the view cluttered and biologists can not find data which is interesting for them. Another category of the genome browsers does not allow to add biologists' own data at all, as the tools only offer some available data and only this data can be visualised. The superiority of VisGenome over other genome browsers is that our tool allows the users to add as much data as they want to see, only depending on memory available on a machine, and does not force them to see data they do not want to navigate, as happens, for example, in Ensembl. VisGenome also offers several visualisation techniques not available in other genome browsers. It allows for smooth zooming and panning, and it also implements a novel CartoonPlus algorithm, which helps the users to see small data items and visualise them in clearly way. VisGenome allows the users to compare their results with the existing data by using the comparative representation.

As observed during the mixed paradigm user study, there is another obvious question with respect to genome browsers: should we produce genome browsers if biologists still use a pen and paper. We are strongly convinced that even if biologists still use their note books and paper, there is a reason to produce a general-purpose genome browser. It is obvious that some people do not like new technology and still use pen and paper, but they also use a number of computer applications, as it is not possible to do all biological calculation and experiments without computer support. As we saw during the second experiment, the biologists make notes in their note books and then they use computer applications. It is worth developing a universal browser, which could be interfaced with Excel and text editor. This would allow the biologists to take notes directly in the application. They could also use tablet computers so as to make their notes in electronic form [128]. Probably, there still will be some biologists who like their

paper note books, but some of them could convert to use the new solutions, or just use it in some cases. Automated data flow between applications could reduce the probability of errors arising during copy and paste operations [19].

9.2.2 Visualisation Techniques

The InfoVis world is very rich and it was not possible to present all visualisation techniques in Chapter 2. Therefore, we described only a small subset of the techniques, as we wanted to show the variety, and make it clear that only a small number of them are used in biology.

Some of the visualisation techniques we looked at seem to be out of date (see Playfair’s parallel time-series bar chart in Chapter 2) or especially designed for special tools as PDA devices¹ (see flip zooming in Chapter 2). On the other hand, the same techniques are often new or even “not yet discovered” in biology.

We also see that not all techniques were evaluated via a user study. On the other hand, some of them, including alphaslider (see Chapter 2) were developed in several versions and considered user preferences. However, the users often were specialists in the visualisation field and not biologists.

It is also worth saying that most InfoVis work assumes a large screen or several displays, while biologists often have a small single screen. During one of our presentations, a specialist in the HCI field asked us about the possibility of using two screens for the presented data. However, this is not possible in small offices where medical researchers work, and where even space for filing their papers seems to be difficult to find. This is also the reason why only a small number of InfoVis techniques, which seem to be ideal for visualising biological data, is used.

We observed that the majority of genome browsers presents genomic data in a linear arrangement. This raises the question of whether other layouts or visualisation techniques could be used in combination with traditional linear maps and if they would be useful for the biologists. DNA structure for the human, the mouse, and the rat chromosomes is linear, and this is the main reason why it is represented as a linear structure. However, when we take any two chromosomes and visualise their comparison, this also could be represented as a graph. In Chapter 10 we present our suggestions for an alternative visualisation in VisGenome.

¹A personal digital assistant (PDA) is a handheld computer, also known as small or palmtop computer.

9.3 Conclusion

In this chapter we discussed the research methods used in this thesis and then the findings (for example usefulness of VisGenome combined with Excel and the number of mistakes made by specialists from biology). We also presented what we learned about our users and their needs during the two experiments. Next chapter describes future work. It presents how the things the biologists like and need could be implemented in VisGenome, and how another user study could be conducted to recognise what other unsatisfied needs they have.

Chapter 10

Future Work

This chapter presents possible directions for future work. First, VisGenome development is presented, which then could be evaluated according in user study.

10.1 Introduction

As can be seen in previous chapters, we developed two versions of VisGenome which were evaluated during different user studies. In the future, we would like to create another (3rd) VisGenome version which could be evaluated via another user study. According to the participants' suggestions and our observations (see Chapter 8), in the section, VisGenome, we present improvements which could be applied to the tool. We also want to add a new feature such as necklace layout, which we believe, improve the biologists work. The suggested improvements are as follow:

- connection between VisGenome and Excel
- uploading data into VisGenome
- adding movable columns and elastic windows
- adding necklace layout
- visualise micro array probes together with micro array probe sets
- optimise speed of VisGenome
- CartoonPlus improvement

We also present a long term user study to evaluate the application.

10.2 VisGenome

VisGenome and Excel

A number of participants use Excel to store their data. They write down notes in their note books and then copy the data into Excel. During the whole process, the medical researchers may make mistakes when transcribing the data. Therefore, we think a good idea would be to support writing of selected data from VisGenome to Excel. The solution would allow the biologists to load their data into Excel without unnecessary data copying.

Uploading Data into VisGenome

During the mixed paradigm user study we prepared a version of VisGenome with biological data, separately for each of the participants. We also consulted them and gave them technical support, however, it is also very important that they be able to upload their data themselves. Currently the users place data in text file format in a special folder (gene, marker, QTL, micro array probe) to see it in VisGenome. We would like to give them the option to chose from the menu a file with their data (txt, Excel, Access, or other format), containing coordinates and data names, and add it in easy way to an existing specified data type or as a new data type.

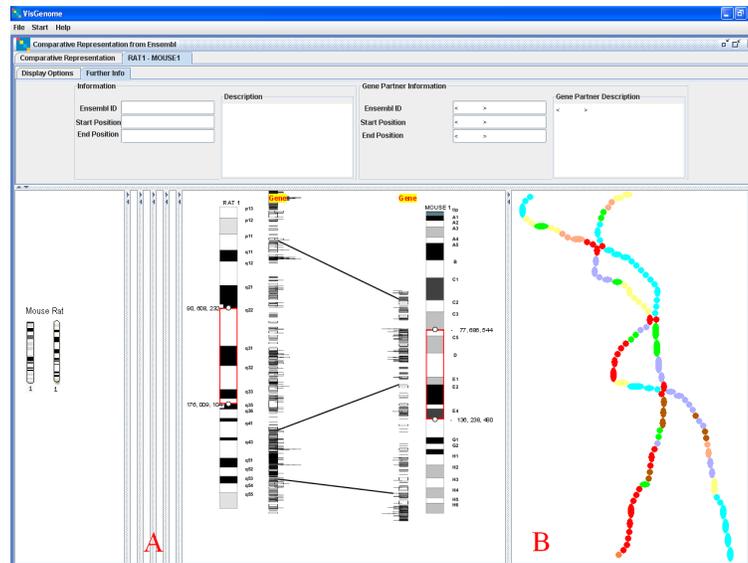


Figure 10.1: Future work - an alternative for visualising gene comparisons - elastic windows (A) and necklace layout for genome representations (B).

Movable Columns and Elastic Windows

A few participants mentioned the requirement to move a column in VisGenome, as it is done in Excel. They would like to compare different types of data and it is easiest when the data to be compared is situated side by side, and not separated by other data. In the new VisGenome, we would like to provide the option to move, add, remove, and hide columns. Each column could have a handle which would be used for moving. The functionality of *hiding columns* could be implemented by elastic windows [55], see Figure 10.1 A. The functionality allows the users to easily hide the windows they do not use in a given data analysis scenario.

Necklace Layout

As can be seen in Chapter 3, all genome browsers present biological data using a linear layout. This seems to be obvious for mammalian genome data, where the chromosomes have a linear structure, in difference to some bacteria and viruses with a circular chromosome structure. However, we could offer new presentation options, see Figure 10.1 B. We would like to represent all genetic data in its order on a chromosome, however, we would like to give a user the option to see comparisons between chromosomes not only as lines comparing homology genes, but also as similar genes situated close to each other. The distance between genes would represent the level of similarity between genes in chromosomes.

Micro Array Probes and Micro Array Probe Sets

Another problem occurring during the mixed paradigm user study was visualising single micro array probes and also micro array probe sets. As described in Chapter 8, the users wanted to see both types of data at the same time. Ensembl holds all the available data, however, the tool uses an algorithm (see Chapter 8) which strictly associates a micro array probe set with a gene which is selected by the algorithm. The medical researchers would like to see all relationships between all micro array probes and genes, not only between micro array probe sets and genes.

We suggest to visualise all micro array probes and group them into micro array probe sets when a probe from the set is marked, see Figure 10.2.

Speed Optimisation

During the mixed paradigm user study, a few participants complained about the slowness of VisGenome. We would like the application to work faster. To do this, we would like to use a different rendering concept. Currently, we download data for a chosen chromosome and paint it in detail. It is a good idea to present an overview at the beginning and after marking a region, but we could avoid downloading all

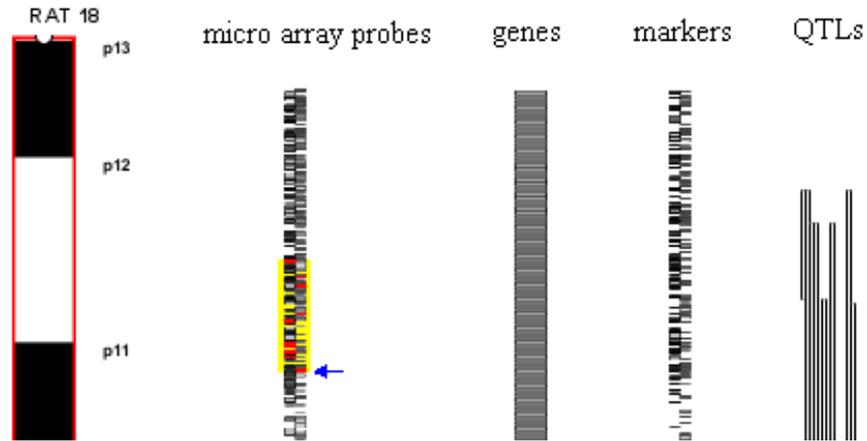


Figure 10.2: Future work - a visualisation of micro array probes and micro array probe sets. The arrow marks a micro array probe, automatically other micro array probes from the same set and the region of the whole micro array probe set are highlighted.

data and paint only a specified region. In an improved solution, we would have less data to render, which would make VisGenome faster.

CartoonPlus Improvement

The novel scaling algorithm, CartoonPlus, described in Chapter 7, takes gene data as a basis (see Figure 10.2) while other data types are scaled depending on genes. We would like to extend the algorithm and allow the users to choose any data type as a basis.

Error Elimination

As mentioned in Chapters 6 and 8, during the experiment we observed a number of user errors. During VisGenome evaluation we removed the source of some errors. For example, visualising QTLs in one line meant that the users could not see where the one QTL finished and another started. However, during the second user study other user errors occurred, such as forgetting about features offered by VisGenome. A user manual was available from the VisGenome web page and from within the application, however, the participants did not use it during the experiment. In the next version of VisGenome, we would like to add to info panel some *'reminders'*, which provide information about features available in the tool. We do not promise to solve all *unsolved InfoVis problems* [15], however, we will try to make VisGenome more useful and easy to use.

10.3 A New User Study

After developing VisGenome version 3, we would conduct a long term user study. As can be seen from our short user studies (see Chapter 6 and 8), a different kind of user study could give different results and fuller answers to the same questions. We believe that during a long term user study we could gain some new insights. Also, the participants would be more familiar with the video camera and could behave in a more natural way.

We would like to conduct a long term user study with the specialists from the fields of biology, medical researchers from the BHF Glasgow Cardiovascular Research Centre, but also with biologists from Edinburgh and London who cooperate with them and with us. We would like to engage up to 10 people and observe them for three months. The methodology used during the future experiment would be observation, video recording, diaries, interviews and voice recordings. If the participants agree, we could install video cameras in their offices and they could be switched on always when they use computers. Additionally, we could spend 3 days with each of the participants, following them and observing their daily activities. During the long term user studies, we could study issues such as:

- Visualisation techniques not used so far in biology could help in genomics data visualisation. For instance, the usability of new types of local scale distortion, the necklace representation, hovering menus, and other techniques should be tested.
- Users prefer access to additional functionality by clicking on buttons in panels (see *labelling* in VisGenome, Figure 7.2) and not by choosing it from a menu. In particular, various placements of buttons, and various representations of panning, zooming controls could be tried out.
- Users prefer access to functions by clicking combined with a key (*Alt*, *Shift*, or *Ctrl*). This would have to be tested with various users, as it is not clear how much the users can remember and what key combinations are the easiest to use. Also, visual aids to such shortcuts could be added to the interface and their usefulness tested.
- Users prefer to select a function from a list in an info panel. The counter hypothesis is that menus or context-driven hovering menus are better.
- History and knowledge about other users activities are helpful for users. The choice of history representation and access to past knowledge would be investigated in detail.

During the new user studies we could log not only the use of VisGenome, but also of other genome browsers and applications used by the biologists. Log data from Ensembl and other tools could give us a number of quantitative and qualitative data which mixed together may give very valuable results, especially that Ensembl is used by a huge number of users interested in genomic data.

We would like to highlight that some medical researchers from the BHF Glasgow Cardiovascular Research Centre still use VisGenome, which can be taken as a concrete demonstration of usability of the tool. This small user population, who continues using VisGenome in their work, could be used in the next, long term user study.

10.4 Conclusion

In this chapter we presented future work. We described future VisGenome development and a new user study. The next chapter concludes the thesis and summarises our contributions.

Chapter 11

Conclusion

The thesis presents work done during my PhD studies between January 2005 and June 2008. It describes VisGenome’s design and its evaluation via user studies with medical researchers from the BHF Glasgow Cardiovascular Research Centre and the BRC at the University of Glasgow. This research started by recognising that there is a large number of genome browsers and visualisation techniques. At the same time we saw that not all known visualisation techniques have been applied to biology and not all of them are appropriate for genetics data (see Chapter 2 and 3). The technique of user study is new in the field of bioinformatics but, as user studies are widely applicable and provide good feedback, we decided to use this type of evaluation. This research focused on *genome visualisation and user studies in biologist-computer interaction*.

The thesis statement postulated that both structured and field-based user studies are an effective methodology in developing new algorithms and visualisation techniques for biological research. We proved this in Chapters 6 and 8 where we presented our initial quantitative and mixed paradigm user studies. We developed a new genome browser, VisGenome (see Chapter 5), and a novel algorithm, CartoonPlus (see Chapter 7), which with other existing visualisation techniques support candidate gene analysis. We subjected VisGenome and CartoonPlus to experimental assessment via user studies. In the following section we list the resulting contributions to HCI, visualisation, and bioinformatics.

11.1 Contributions

Here we present the work’s main contributions to HCI, visualisation, and bioinformatics.

The first contribution presented in Chapter 5 is VisGenome which, together with improved visualisation techniques, supports candidate gene analysis and subjects them to experimental assessment

via user studies. VisGenome is a novel implementation of the genome browser, which was published in Bioinformatics [51]. VisGenome was designed to support candidate gene analysis and is the only tool in existence that can show QTLs, genes, micro array probes and SNPs side by side in a fashion that supports biomedical research. The tool is in use in Glasgow, which shows that it fulfils its purpose. Its usefulness was confirmed by our second user study (Chapter 8). Genome browsers should have visualisation techniques which allow the users to easily navigate their data. As we presented in Chapter 4 only a small part of a biologists' work is connected to computers, but rather the work is integrated with other activities. Therefore it is important that the tools allow them to do their work effectively and quickly.

The second major contribution is a novel algorithm, CartoonPlus, which was designed, implemented and tested via a user study. The algorithm together with VisGenome improves gene visualisation. The scaling algorithm CartoonPlus is presented in Chapter 7 and published in the departmental technical report [49] and then in Lecture Notes in Computer Science (LNCS) [50]. The scaling uses one type of genomic object as a basis and then scales all other objects with respect to the base type, to support human understanding of linear data relationships on a chromosome, or in a comparison of two chromosomes. VisGenome and CartoonPlus make a novel contribution to bioinformatics and visualisation, and are used by biologists and medical researchers from the BHF Cardiovascular Research Centre and the BRC who enjoy the application and find it useful, especially for small genomics objects (SNPs, see Chapter 8). New visualisation techniques could be very useful, therefore it is important to have novel techniques or techniques not used in bioinformatics tested with specialists from this field.

The third contribution presented in this thesis is a novel genome browser classification. The classification is described in Chapter 3. It classifies existing genome browsers with respect to the level of data granularity and the number of species presented, and observes that the main differences are the distinctions between single and multiple genome representations. This was published as a technical report [48]. Our genome browser classification is a contribution to bioinformatics, where a large number of genome browsers exist, but have not been classified so far.

Other contribution to human-computer interaction is the feedback from the user studies presented in Chapters 6 and 8, which represent effective methodologies in developing new algorithms and visualisation techniques for biological research. So far, only the initial quantitative user study was published in LNCS [52], but we also aim to publish our mixed paradigm user study. The user studies were conducted with specialists from the field - medical researchers. The new information about way how biologists work, their daily activities and tasks descriptions gathered from the user studies are necessary for further development of genome browsers. The main findings are that reducing clutter, the number of colours, and presenting the chromosomes vertically in a genome visualisation makes it easier to use, as compared to Ensembl. Also, users do not like dragging, but prefer clicking on objects to gather additional information, and they do not like pop-up windows which obscure the view. They prefer additional information to be presented

in a separate panel. Furthermore, they use visualisation alongside other tools, and need to be able to transfer data smoothly between various applications using the same data. They work in an environment which puts them under considerable cognitive strain, as they use many tools and techniques, both in the lab and at their computer, and suffer thus from information overload, as they forget about software functions, and have no time or patience to study handbooks or manuals which could help them to take advantage of existing software.

11.2 Closing

In this thesis we presented VisGenome which was improved during user studies evaluation. Genome browsers should be more user-friendly and allow the users to easily find the information they require. It is important that this kind of application implements interaction techniques which shorten navigation time. Therefore, we suggest that genome browsers should offer more visualisation techniques which should be tested via user studies.

As we presented, user studies are valuable techniques for application evaluation and could provide suggestions which make an application more useful. During our user studies we used a variety of methodologies. We think that the different kinds of methodologies used could give different results, see Chapter 6 and 8. However, it is worth remembering that not all methodologies are possible in some situations. Therefore, if a set of users conduct a work with animals, video recording is not allowed. In situations when video recording could be used, it is very helpful and could be used for future analysis.

The thesis presents new ideas for supporting the work of medical researchers and biologists and helping them to find the genes responsible for diseases. We developed VisGenome which is successfully used by the members of the BHF Cardiovascular Research Centre, implemented CartoonPlus algorithm, which is a novel visualisation technique in biology, and evaluated the application and the algorithm via user studies. The user studies are very popular in different scientific fields, however, not common in bioinformatics. The risk of applying user studies to our work was worth taking. We received a lot of feedback and new suggestions (see Chapters 6, 8, and 10) which were used in the redevelopment of VisGenome and during the evaluation. Our work contributes to the future development of genome visualisation tools, offering many potentially useful directions for future work.

Glossary

Affymetrix

Affymetrix is a manufacturer of micro arrays. Information on Affymetrix can be found at <http://www.affymetrix.com>, see Chapter 4.

anthropology

Anthropology is a study of humanity. Anthropology has origins in the natural sciences, the humanities, and the social sciences, [116], see Chapter 4.

array

see micro array, see Chapter 4.

artificial intelligence

Artificial intelligence is the study and design of intelligent agents, where an intelligent agent is a system that perceives its environment and takes actions which maximize its chances of success, [97], see Chapter 4.

base

Nucleotide sequences (DNA and RNA) are composed of bases. DNA is composed of the four bases adenine (A), cytosine (C), guanine (G), and thymine (T). The same bases from RNA, with the exception of thymine which is replaced with uracil (U).

base pair (bp)

When two bases in different nucleotide sequences bind to each other they are said to be a base pair. In DNA sequences G pairs with C and A with T. Generally, DNA sequences are double stranded, consisting of two sequences that have paired with each other. The length of DNA sequence is given in bases or base pairs (bp).

benchmarking

A means of assessing performance. Methods are compared to some predefined standard, or benchmark, against which performance can be compared, see Chapter 6.

biological sequences

Biological sequences include DNA sequences, RNA sequences and protein sequences.

British Heart Foundation Glasgow Cardiovascular Research Centre (BHF GCRC)

The aim of the British Heart Foundation Glasgow Cardiovascular Research Centre is to consolidate internationally recognised cardiovascular research groups and to provide a multidisciplinary research environment. The BHF Glasgow Cardiovascular Research Centre was funded jointly by the British Heart Foundation and the University of Glasgow. We cooperate with the Centre during VisGenome development.

cDNA

see complementary DNA.

chromosome

Chromosome houses the majority of an organism's genetic material. In organisms such as the rat chromosomes are linear structures, with the exception of the mitochondrion which has a circular genome.

C.I.

see confidence interval.

cognitive psychology

Cognitive psychology is a school of thought in psychology that examines internal mental processes such as problem solving, memory, and language. Cognitive psychologists are interested in how people understand, diagnose, and solve problems, concerning themselves with the mental processes which mediate between stimulus and response, see Chapter 4.

complementary

Complementary sequences are those that can base pair, or hybridise, with each other. When running in the opposite direction, a complementary sequence contains the bases that can bind to those in the other strand in the corresponding positions.

complementary DNA (cDNA)

A DNA sequence generated from an RNA template.

computer science

Computer science (computing science) is the study of the theoretical foundations of information and computation and their implementation and application in computer systems, see Chapter 4.

confidence interval

A confidence interval (C.I.) is an estimated range of values which is likely to include an unknown population parameter, the estimated range being calculated from a given set of sample data, see Appendix O ('Statistical Methods').

deoxyribonucleic acid (DNA)

DNA is composed of the four bases adenine (A), guanine (G), cytosine (C), and thymine (T). In double stranded DNA molecules two chains of these bases bind to, or base pair, with each other, A pairing with T, and C with G, see Chapter 3.

design

Design refers to the process of originating and developing a plan for a product, structure, system, or component. It is also used for either the final (solution) plan or the result of implementing that plan, see Chapter 4.

DNA

see deoxyribonucleic acid.

engineering

Engineering is the discipline of acquiring and applying scientific and technical knowledge to the design, analysis, and/or construction of works for practical purposes, see Chapter 4.

Ensembl

A resource provided by EMBL-EBI and the Sanger Centre giving access to genome annotation, Available at www.ensembl.org, see Chapter 3.

eQTL

expression Quantitative Trait Locus, see quantitative trait loci.

ergonomics

Ergonomics (human factors) is “traditionally the study of the physical characteristics of the interaction: how the controls are designed, the physical environment in which the interaction takes place, and the layout and physical qualities of the screen”, [23], see Chapter 4.

European Molecular Biology Laboratory (EMBL)

Located primarily in Heidelberg in Germany, EMBL (www.embl-heidelberg.de) represents a European effort to conduct research in molecular biology. It includes the EMBL bioinformatics outstation, the European Bioinformatics Institute (www.ebi.ac.uk), which is located at Hinxton in the UK.

exon

Part of the genomic sequence that encodes part of a protein sequence. These sequences are spliced together, being divided on genome by introns.

GenBank

A database containing all publicly available nucleotide sequences and part of the International Nucleotide Sequence Database Collaboration provided by the USA through the National Center for Biotechnology Information. Available at www.ncbi.nlm.nih.gov/Genbank/.

gene

An interval on the genome containing the code and regulatory mechanisms for a transcript. In eukaryotic organisms, where the coding DNA is split into exons, divided by introns of non-coding DNA, the exons may be spliced together in different ways producing a number of variants.

gene ontology

Created by the Gene Ontology Consortium, the gene ontology uses a set of controlled vocabularies to describe the relationships between genes at a variety of levels.

genome

The entire genetic complement of an organism.

homology

Homologous sequences are those that are believed to share an evolutionary relationship with each other. Homolog gene is a gene related to a second gene by DNA sequence [125]. There are three different types of homologous comparison possible: orthologs are genes related in different species, paralogs are genes related by duplication within a genome, and xenologs, where one gene is compared

to the duplicated gene in a different species. Orthologs retain the same function in the course of evolution, whereas paralogs evolve new functions, even if these are related to the original one. In the genetic sense orthologs, paralogs, and metalogs are all homologs but the one most commonly used is the orthologs relationship.

homology search

A search for homologous sequences. Generally, this involves searching for sequences that are likely to have an evolutionary relationship to a query sequence. Homology searches are a specific type of similarity search, where the type of similarity sought is that which is likely to indicate an evolutionary relationship.

hybridisation

Hybridisation occurs when single stranded nucleotide sequences that have been incubated together base pair with each other.

hypertension

Persistent high blood pressure.

International Nucleotide Sequence Database Collaboration

An organisation with members in Europe, the United States of America and Japan, each of which maintains and provides access to a copy of the public nucleotide collection.

intron

A region of non-coding genomic sequence. Introns divide coding exons.

Kaplan-Meier survival plot

The Kaplan-Meier survival plot is used for analysing survival data. It is very popular in medical research, where the patients receive medicaments which help them (they survive) or in some situations the patients die, see Appendix O ('Statistical Methods').

linguistics

Linguistics is a scientific study of language, which can be theoretical or applied, see Chapter 4.

Mann-Whitney test

The Mann-Whitney test (also known as Wilcoxon-Mann-Whitney test) is the non-parametric equivalent of the 2-sample t-test. The Mann-Whitney tests two independent samples of numerical or ordinal values. These samples do not need to contain the same number of observations, see Appendix O ('Statistical Methods').

McNemar's test

The McNemar's test is a categorical data method used for the comparison of proportions from paired samples. Q. McNemar introduced this test in 1947, using it on 2×2 contingency tables with a dichotomous trait with matched pairs of subjects, see Appendix O ('Statistical Methods').

mean

A mean used in the thesis is an arithmetic mean (\bar{x}) which is calculated by summarising all numbers from a list of numbers and then dividing the number by the number of items in the list, see Appendix O ('Statistical Methods').

median

A median is defined as the number separating the higher half of a sample, a population, or a probability distribution, from the lower half, see Appendix O ('Statistical Methods').

messenger RNA

An RNA molecule which contains the code for a protein sequence, see Chapter 3.

micro array

An array of DNA probes, see Chapter 4.

mRNA

see messenger RNA.

National Center for Biotechnology Information (NCBI)

Part of the National Institutes of Health in the USA and member of the International Nucleotide Sequence Database Collaboration which provides access to GenBank and other resources. Further information is available at www.ncbi.nlm.nih.gov.

nonparametric methods

The nonparametric methods require fewer assumptions about a population or probability distribution than the parametric methods and are applicable in a wider range of situations, see Appendix O ('Statistical Methods').

normal distribution

The normal distribution is also called the Gaussian distribution and it is very important in statistics. The distribution is defined by two parameters: the mean and variance. All normal distributions are symmetric and have bell-shaped density curves with a single peak, see Appendix O ('Statistical Methods').

nucleotide

Nucleotides are the bases adenine (A), cytosine (C), guanine (G), thymine (T), and uracil (U). See base.

null hypothesis

The null hypothesis, traditionally represented by the symbol H_0 , represents the hypothesis of no change or no effect, see Appendix O ('Statistical Methods').

oligo

see oligonucleotide.

oligonucleotide

An oligonucleotide is a short nucleotide sequence.

p-value

A p-value is a measure of how much evidence we have against the null hypothesis, see Appendix O ('Statistical Methods').

parametric methods

The parametric methods assume the data follow a normal distribution, see Appendix O ('Statistical Methods').

philosophy

Philosophy is the discipline concerned with questions of how one should live (ethics); what sorts of things exist and what their essential natures are (metaphysics); what counts as genuine knowledge (epistemology); and what the correct principles of reasoning are (logic) [25], see Chapter 4.

physical maps

Physical maps show the physical length of DNA measured in base pairs. In molecular biology, two nucleotides on opposite complementary DNA or RNA strands that are connected via hydrogen bonds are called a base pair (often abbreviated bp). In the canonical Watson-Crick base pairing, adenine (A) forms a base pair with thymine (T), as does guanine (G) with cytosine (C) in DNA. Sequence-based maps improve with the scientific progress and are perfect when the genomic DNA sequencing of the species has been completed. In the meantime, it is worth to mention the genetic map, which was used when physical and sequence-based maps did not exist. A scale in a genetic map was measured in centimorgan (cM) or map unit (m.u.), which is a unit of recombinant frequency for measuring genetic linkage. It is often used to imply distance along a chromosome. The number

of base-pairs it corresponds to varies widely across the genome (different regions of a chromosome have different propensities towards crossover), and is about 1 million base pairs in humans. The centimorgan is equal to a 1% chance that a marker at one genetic locus on a chromosome will be separated from a marker at a second locus due to crossing over in a single generation. A 50 cM distance means that the genes will reassort when an odd number of crossings happen, which happens 31.8% of the time, see Chapter 3.

population

A population is the set (often infinite) of all possible individuals we would have sampled, see Appendix O ('Statistical Methods').

pQTL

physiological Quantitative Trait Locus, see quantitative trait loci.

probe

On Affymetrix arrays, a 25 base DNA sequence forming part of an array of such probes. By base pairing, or hybridising, with transcripts probes can be used to detect the levels of transcripts with complementary sequences present in a sample, see Chapter 4.

protein

A structure built from a chain of amino acids. Over 20 characters are used to represent amino acids.

QTL

see quantitative trait loci.

quantitative trait loci

A quantitative trait loci is a part of a chromosome which is correlated with a physical characteristic, such as height or disease. Micro array probes are used to test gene activity (expression), see Chapter 3. Two types of QTLs exist: pQTLs and eQTLs. Expression Quantitative Trait Locus (eQTL) mapping tries to find genomic variation to explain expression traits. One difference between eQTL mapping and traditional pQTL mapping (physiological QTL) is that a traditional mapping study focuses on one or a few traits, while in most of eQTL studies, thousands of expression traits will be analyzed and thousands of pQTLs will be declared.

ribonucleic acid (RNA)

Usually a single stranded molecule, RNA is composed of the same four bases as DNA but with uracil (U) replacing thymine (T). While RNA typically forms different structures from DNA, the bases found within RNA molecules still base pair and can also base pair with single stranded DNA molecules.

RNA

see ribonucleic acid.

sample

A sample is a set of data collected for an experiment, see Appendix O ('Statistical Methods').

sign test

The sign test is used to test the null hypothesis that positive and negative results are equally likely. It can also be used to test a hypothesis about a median, because the hypothesis that a median equals 7 (for example), is the hypothesis that equal numbers of cases fall above (positive results) and below (negative results) 7, see Appendix O ('Statistical Methods').

single nucleotide polymorphism

A single nucleotide polymorphism (SNP) is a DNA sequence variation occurring when a single nucleotide (A, T, C, or G) in the genome (or other shared sequence) differs between members of a species (or between paired chromosomes in an individual). For example, two sequenced DNA fragments from different individuals, AAGCCTA to AAGCTTA, contain a difference in a single nucleotide (C and T), see Chapter 3.

SNP

see single nucleotide polymorphism.

social and organizational psychology

Social psychology is the study of how social conditions affect human beings, see Chapter 4.

sociology

Sociology is the scientific study of society, including patterns of social relationships, social interaction, and culture, see Chapter 4.

splice

The sequence of coding exons are spliced together to form transcripts in eukaryotes. Exons are divided by introns, regions of non-coding DNA which do not form part of the transcript.

standard deviation

A standard deviation is variance's square root, see Appendix O ('Statistical Methods').

standard error

A standard error (SE) is the standard deviation divided by the square root of the sample size, see Appendix O ('Statistical Methods').

statistical significance

Statistical significance means that $p < 0.05$ (p is short for p-value), see Appendix O ('Statistical Methods').

t-test

The t-test assesses whether the means of two groups are statistically different from each other, see Appendix O ('Statistical Methods').

transcript

An RNA molecule produced by transcription.

transcription

A process resulting in the production of an RNA sequence, transcribed from a DNA sequence. Transcription forms part of the process of gene expression, which occurs when a gene is "switched on".

translation

The synthesis of a protein from an RNA sequence.

"trend towards significance"

"Trend towards significance" means that although we cannot classify results as statistically significant, the result is borderline (typically $0.05 < p < 0.1$). The implication when sample sizes are small is that a larger sample might have given a significant result, see Appendix O ('Statistical Methods').

UniGene

The UniGene database contains entries which are sets of transcripts that appear to share the same transcription locus. These entries are linked to additional information.

<http://www.ncbi.nlm.nih.gov/UniGene>.

University of California at Santa Cruz (UCSC)

Used to refer to the bioinformatics resources provided by a group based at UCSC. Among the materials available at the UCSC site is a genome browser, providing access to genomes and their annotation. <http://genome.ucsc.edu>

variance

A variance is one measure of statistical dispersion, averaging the squared distance of its possible values from the expected value (mean), see Appendix O ('Statistical Methods').

Wilcoxon signed rank test

The Wilcoxon signed rank test (also known as Wilcoxon matched pairs test) is a non-parametric test used when one sample (or paired) t-test is not appropriate and it tests the median difference in paired data. Each individual in the sample generates two paired data values, one from first measurement and one from second measurement. Differences between the paired data values are used to test for a difference between the two populations, see Appendix O ('Statistical Methods').

1-proportion test

The 1-proportion test to test categorical data. It is a hypothesis test of a population proportion, and examines the population proportion using information from one sample and comparing it to a target value, see Appendix O ('Statistical Methods').

2-sample t-test

The 2-sample t-test is a parametric method in statistics. The test assesses whether the means of two groups are statistically different from each other. It compares the difference between two means in relation to the variation in the data, see Appendix O ('Statistical Methods').

Bibliography

- [1] C. Ahlberg and B. Shneiderman. Alphaslider: A Rapid and Compact Selector. *Proc. CHI Conf.*, ACM Press, New York, 1994.
- [2] R. M. Baecker and W. A. S. Buxton. Readings in Human-Computer: A Multi-disciplinary Approach. Los Altos, CA:Morgan Kaufmann, 1987.
- [3] E. H. Baehrecke, N. Dang, K. Babaria, and B. Shneiderman. Visualization and analysis of microarray and gene ontology data with treemaps. *BMC Bioinformatics*, **5**:84, 2004.
- [4] B. B. Bederson, J. Grosjean, and J. Meyer. Toolkit Design for Interactive Structured Graphics. *IEEE Transactions on Software Engineering* **30**(8), 535-546, 2004.
- [5] B. B. Bederson and J. D. Hollan. Pad++: A Zooming Graphical Interface for Exploring Alternative Interface Physics. *Proc. ACM User Interface Software and Technology '94*, 17-27, 1994.
- [6] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: the see-through interface. *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 73-80, September 1993.
- [7] J. Bingham and S. Sudarsanam. Visualizing large hierarchical clusters in hyperbolic space. *Bioinformatics* **16**, 660661, 2000.
- [8] J. M. Bland. An Introduction To Medical Statistics, Third Edition, OUP, Oxford, 2000.
- [9] W. B. Boehm. A Spiral Model of Software Development and Enhancement. *Computer*, v.21 n.5, 61-72, May 1988.
- [10] R. Bourqui, L. Cottret, V. Lacroix, D. Auber, P. Mary, M. F. Sagot, and F. Jourdan. Metabolic network visualization eliminating node redundancy and preserving metabolic pathways. *BMC Systems Biology*, 1:29, 2007.
- [11] S. K. Card, J. D. Mackinlay, and B. Shneiderman. Readings in Information Visualization; Using Vision to Think. Los Altos, CA, Morgan Kaufmann, 1999.

- [12] S. Card, G. Robertson, and J. Mackinlay. The Information Visualizer - an information workspace. *In Proceedings of the Conference on Computer Human Interaction*, ACM, 181-188, 1991.
- [13] K. Chakrabarti and L. Pachter. Visualization of multiple genome annotations and alignments with the K-BROWSER. *Genome Research*, 2004.
- [14] J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey. *Graphical Methods for Data Analysis* Pacific Grove, CA: Wadsworth, 1983.
- [15] C. Chaomei. Top 10 Unsolved Information Visualization Problems. *IEEE Computer Graphics and Applications*, 25(4), 12-16, July-August 2005.
- [16] H. Chernoff. The Use of Faces to Represent Points in k-Dimensional Space Graphically. *J. Am. Statistical Assoc.*, vol. 68, 361-368, June 1973.
- [17] J. M. Cherry, C. Adler, C. Ball, S. A. Chervitz, S. S. Dwight, E. T. Hester, Y. Jia, G. Juvik, T. Y. Roe, M. Schroeder, S. Weng and D. Botstein. SGD: Saccharomyces Genome Database. *Nucleic Acids Research* vol. 12, no. 1, 73-79, 1998.
- [18] R. Chimera. Value bars: An information visualization and navigation tool for multiattribute listings. Proc. ACM CHI92 Conference: *Human Factors in Computing Systems*, 293-294, 1992.
- [19] A. Chou, J. Yang, B. Chelf, S. Hallem, and D. R. Engler. An Empirical Study of Operating System Errors. *Proc. Symp. Operating Systems Principles*, 73-88, 2001.
- [20] S. Choudhuri. The Path from Nuclein to Human Genome: A Brief History of DNA with a Note on Human Genome Sequencing and Its Impact on Future Research in Biology. *Bulletin of Science, Technology & Society*, 23(2003):360-367, 2003.
- [21] L. B. Christensen. *Experimental Methodology*. Atlantic Avenue, Boston, Massachusetts, 1977.
- [22] P. Craig, J. Kennedy, and A. Cumming. Coordinated Parallel Views for the Exploratory Analysis of Microarray Timecourse Data. *In CMV'05: Proceedings of the Coordinated and Multiple Views in Exploratory Visualization (CMV'05)*, 314, Washington, DC, USA, 2005.
- [23] A. Dix, J. Finlay, G. D. Abowd, and R. Beale. *Human-Computer Interaction*, 3rd Edition, Addison-Wesley Pearson Education, London, 2004.
- [24] A. Dominiczak, D. Negrin, J. Clark, M. Brosnan, M. McBride, and M. Alexander. Genes and hypertension - from gene mapping in experimental models to vascular gene transfer strategies. *Hypertension*, 35, 164-172, 2000.
- [25] W. Durant. *Story of Philosophy: The Lives and Opinions of the World's Greatest Philosophers*. Pocket, ISBN 0671739166, ISBN-13 978-0671739164, 1991.

- [26] R. Durbin and J. Thierry-Mieg. A C. elegans Database. Documentation, code and data available from anonymous FTP servers at lirmm.lirmm.fr, cele.mrc-lmb.cam.ac.uk and ncbi.nlm.nih.gov, 1991.
- [27] E. B. Ebbesen and M. Haney. Flirting ith death: Variables affecting risk taking at intersections. *Journal of Applied Social Psychology*, 3, 303-324, 1973.
- [28] S. G. Eick, J. L. Steffen, and E. E. Jr. Sumner. SeeSoft - A tool for visualizing line-oriented software statistics. *IEEE Transactions on Software Engineering* 18, 11, 957-968, 1992.
- [29] G. Ellis and A. Dix. An explorative analysis of user evaluation studies in information visualization. In Proc. of BELIV 2006.
- [30] J. D. Fekete and C. Plaisant. Excentric Labeling: Dynamic Neighbourhood Labeling for Data Visualization. *CHI'99* Pittsburgh, PA, USA, May 15-20, 512-519, 1999.
- [31] G. Fischer, S. M. Ibrahim, G. A. Brockmann, J. Pahnke, E. Bartocci, H. J. Thiesen, P. Serrano-Fernndez, and S. Mller. Expressionview: visualization of quantitative trait loci and gene-expression data in Ensembl. *Genome Biology*, 4, 2003.
- [32] G. W. Furnas. The FISHEYE view: a new look at structured files, Bell Laboratories Technical Memorandum, 1982.
- [33] G. W. Furnas. Generalized Fisheye Views, Human Factors in Computing Systems, *CHI '86 Conference Proceedings*, 16-23, 1986.
- [34] A. M. Glazier, J. H. Nadeau, and T. J. Aitman. Finding Genes That Underlie Complex Traits. *Science* 298, 2345-2349, 2002.
- [35] M. Graham, J. Kennedy, and D. Benyon. Towards a methodology for developing visualizations. *International Journal of Human-Computer Studies*, vol. 53, 5, 789-807, 2000.
- [36] R. R. Graham, C. D. Langefeld, P. M. Gaffney, W. A. Ortmann, S. A. Selby, E. C. Baechler, K. B. Shark, T. C. Ockenden, K. E. Rohlf, K. L. Moser, W. M. Brown, S. E. Gabriel, R. P. Messner, R. A. King, P. Horak, J. T. Elder, P. E. Stuart, S. S. Rich and T. W. Behrens. Genetic linkage and transmission disequilibrium of marker haplotypes at chromosome 1q41 in human systemic lupus erythematosus. *Arthritis Res.* 3, 299305, 2001.
- [37] G. Hardiman. Microarray platforms - comparisons and contrasts. *Pharmacogenomics*, 5(5), 487-502, 2004.
- [38] H. R. Hartson and D. Hix. Toward empirically derived methodologies and tools for humancomputer interface development. *Int. J. ManMachine Studies* 31, 477494, 1989.

- [39] G. A. Helt, S. Lewis, A. E. Loraine, and G. M. Rubin. BioViews: Java-Based Tools for Genomic Data Visualization. *Genome Research*, 8, 291-305, 1998.
- [40] A. Herraiez. Biomolecules in the computer. Jmol to the rescue. *Biochem. Mol. Biol. Educ.* 34: 255261, 2006.
- [41] T. T. Hewett, R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, G. Strong, and W. Verplank. Curriculum for Human-computer Interaction. ACM Special Interest Group of Computer-Human Interaction Curriculum Development Group. ACM SIGCHI. New York, 1992.
- [42] W. C. Hill, J. D. Hollan, D. Wroblewski, and T. McCandless. Edit Wear and Read Wear. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 3-9, 1992.
- [43] L. E. Holmquist. Focus+Context Visualization with Flip Zooming and the Zoom Browser. In *Extended Abstracts of CHI'97*, ACM Press, 1997.
- [44] T. J. P. Hubbard, B. L. Aken, K. Beal, B. Ballester, M. Caccamo, Y. Chen, L. Clarke, G. Coates, F. Cunningham, T. Cutts, T. Down, S. C. Dyer, S. Fitzgerald, J. Fernandez-Banet, S. Graf, S. Haider, M. Hammond, J. Herrero, R. Holland, K. Howe, N. Johnson, A. Kahari, D. Keefe, F. Kokocinski, E. Kulesha, D. Lawson, I. Longden, C. Melsopp, K. Megy, P. Meidl, B. Ouverdin, A. Parker, A. Prlic, S. Rice, D. Rios, M. Schuster, I. Sealy, J. Severin, G. Slater, D. Smedley, G. Spudich, S. Trevanion, A. Vilella, J. Vogel, S. White, M. Wood, T. Cox, V. Curwen, R. Durbin, X. M. Fernandez-Suarez, P. Flicek, A. Kasprzyk, G. Proctor, S. Searle, J. Smith, A. Ureta-Vidal, and E. Birney. Ensembl 2007. *Nucleic Acids Res.* 35, Database issue: D610-D617, 2007.
- [45] N. Hubner, C. A. Wallace, H. Zimdahl, E. Petretto, H. Schulz, F. Maciver, M. Mueller, O. Hummel, J. Monti, V. Zidek, A. Musilova, V. Kren, H. Causton, L. Game, G. Born, S. Schmidt, A. Mller, S. A. Cook, T. W. Kurtz, J. Whittaker, M. Pravenec, and T. J. Aitman. Integrated transcriptional profiling and linkage analysis for identification of genes underlying disease. *Nat. Genet.*, 37, 243-253, 2005.
- [46] E. Hunt, N. Hanlon, D. Leader, H. Bryce and A. F. Dominiczak The visual language of synteny. *OMICS* 8(4), 289-305, 2004.
- [47] A. Inselberg. The plane with parallel coordinates. *Special Issue on Computational Geometry of The Visual Computer*, 1(2), 69-91, 1985.
- [48] J. Jakubowska, E. Hunt, and M. Chalmers. Granularity of genomics data in genome visualisation. University of Glasgow, Tech. Rep.: TR-2006-221, 2006.
<http://www.dcs.gla.ac.uk/publications/PAPERS/8212/AsiaElaMatthewCHI06.pdf>.

- [49] J. Jakubowska, E. Hunt, and M. Chalmers. CartoonPlus: A New Scaling Algorithm for Genomics Data. Department of Computing Science, University of Glasgow, Technical Report TR-2007-259, 2007. http://www.dcs.gla.ac.uk/publications/PAPERS/8757/VG_VisTech.pdf
- [50] J. Jakubowska, E. Hunt, and M. Chalmers. CartoonPlus: A New Scaling Algorithm for Genomics Data. *ICCS'08 proceedings*, LNCS 5103, Springer, 158-167, 2008.
- [51] J. Jakubowska, E. Hunt, M. Chalmers, M. McBride, and A. F. Dominiczak. VisGenome: visualization of single and comparative genome representations. *Bioinformatics* vol. 23, no. 19, 26412642, 2007.
- [52] J. Jakubowska, E. Hunt, J. McClure, M. Chalmers, M. McBride, and A. F. Dominiczak. VisGenome and Ensembl: Usability of Integrated Genome Maps. *DILS'08 proceedings* (the paper is nominated for the Best Paper Award), LNCS, Springer, 2008.
- [53] J. Jakubowska, J. McClure, E. Hunt, and M. Chalmers. Usability of VisGenome and Ensembl - A User Study. Dept. of Comp. Sc. Technical Report, TR-2007-244, 2007. http://www.dcs.gla.ac.uk/publications/PAPERS/8510/VG&Ens_TechRep.pdf.
- [54] B. Johnson and B. Shneiderman. Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures. *Proc. IEEE Visualization'91*, 284-291, 1991.
- [55] E. Kandogan and B. Shneiderman. Elastic Windows: a hierarchical multi-window World-Wide Web browser. *Proceedings of the 10th annual ACM symposium on User interface software and technology*, 169-177, 1997.
- [56] D. Karolchik, R. Baertsch, M. Diekhans, T. S. Furey, A. Hinrichs, Y. T. Lu, K. M. Roskin, M. Schwartz, C. W. Sugnet, D. J. Thomas, R. J. Weber, D. Haussler, and W. J. Kent. The UCSC Genome Browser Database. *Nucleic Acids Res.* **31**, 51-54, 2003.
- [57] R. Kosara, C. G. Healey, V. Interrante, D. H. Laidlaw, and C. Ware. Thoughts on User Studies: Why, How, and When. *IEEE Computer Graphics and Applications*, vol. 23, no. 4, 20-25, July/Aug. 2003.
- [58] A. E. Kwitek, P. J. Tonellato, D. Chen, J. Gullings-Handley, Y. S. Cheng, S. Twigger, T. E. Scheetz, T. L. Casavant, M. Stoll, M. A. Nobrega, M. Shiozawa, M. B. Soares, V. C. Sheffield, and H. J. Jacob. Automated Construction of High-Density Comparative Maps Between Rat, Human, and Mouse. *Genome Research*, vol. 11, 1935-1943, November 2001.
- [59] J. Lamping and R. Rao. The Hyperbolic Browser: A Focus+Context Technique for Visualizing Large Hierarchies. *J. Vis. Lang. Comput.*, **7**, 35-55, 1996.
- [60] D. P. Leader BugView: a browser for comparing genomes. *Bioinformatics* **20**, 129-130, 2004.

- [61] U. Leser, R. Wagner, A. Grigoriev, H. Lehrach, and H. Roest Crolius IXDB, an X chromosome integrated database. *NAR* 26(1), 108-111, 1997.
- [62] Y. K. Leung and M. D. Apperley. A Review and Taxonomy of Distortion-Oriented Presentation Techniques. *ACM Transactions on Computer-Human Interaction (TOCHI)*, v.1 n.2, 126-160, June 1994.
- [63] S. E. Lewis, S. M. J. Searle, N. Harris, M. Gibson, V. Iyer, J. Richter, C. Wiel, L. Bayraktaroglu, E. Birney, M. A. Crosby, J. S. Kaminker, B. B. Matthews, S. E. Prochnik, C. D. Smith, J. L. Tupy, G. M. Rubin, S. Misra, C. J. Mungall, and M. E. Clamp Apollo: a sequence annotation editor. *Genome Biology*, 2002.
- [64] J. D. Mackinlay, G. G. Robertson, and S. K. Card. The Perspective Wall: Detail and Context Smoothly Integrated. *Proceedings of the SIGCHI Conference on Human factors in computing systems*, 173-176, 1991.
- [65] M. W. McBride, F. J. Carr, D. Graham, N. H. Anderson, J. S. Clark, W. K. Lee, F. J. Charchar, M. J. Brosnan, and A. F. Dominiczak. Microarray analysis of rat chromosome 2 congenic strains. *Hypertension* 41, 847853, 2003.
- [66] M. W. McBride, F. J. Charchar, D. Graham, W. H. Miller, P. Strahorn, F. J. Carr, and A. F. Dominiczak. Functional genomics in rodent models of hypertension. *J Physiol*, 554, 5663, 2004.
- [67] M. W. McBride, D. Graham, C. Delles, and A. F. Dominiczak. Functional genomics in hypertension. *Curr Opin Nephrol Hypertens* 15(2), 145151, 2006.
- [68] P. McConnell, K. Johnson, and S. Lin. Applications of Tree-Maps to hierarchical biological data. *Bioinformatics*, Vol. 18, No. 9, 1278-1279, 2002.
- [69] B. H. McCormick, T. A. DeFanti, and M. D. Brown. Visualization in scientific computing. *Computer Graphics*, 21(6), 103-111, 1987.
- [70] R. McGill, J. W. Tukey, and W. A. Larsen. Variations of Box Plots. *The American Statistician*, Vol. 32, No. 1, 12-16, 1978.
- [71] Q. McNemar. Note on the sampling error of the differences between correlated proportions of percentages. *Psychometrika* 12, 153-157, 1947.
- [72] J. S. Mill. A system of logic. New York:Harper, 1874.
- [73] R. Molich and J. Nielsen. Improving a human-computer dialogue. *Communications of the ACM*, 33(3), 338-48, 1990.

- [74] A. Monk, P. Wright, J. Haber, and L. Davenport. Improving Your Human-Computer Interface. A Practical Technique. New York:Prentice-Hall, 1993.
- [75] S. B. Montgomery, T. Astakhova, M. Bilenky, E. Birney, T. Fu, M. Hassel, C. Melsopp, M. Rak, A. Gordon Robertson, M. Sleumer, A. S. Siddiqui, and S. J. M. Jones. Sockeye: A 3D Environment for Comparative Genomics. *Submitted Genome Research*, 2003.
- [76] A. Morrison, P. Tennent, and M. Chalmers. Coordinated Visualisation of Video and System Log Data. *Proc. 4th International Conference on Coordinated & Multiple Views in Exploratory Visualisation*, London, 91-102, 2006.
- [77] M. Mueller, A. Goel, M. Thimma, N. J. Dickens, T. J. Aitman, and J. Mangion. eQTL Explorer: integrated mining of combined genetic linkage and experiments. *Bioinformatics* 22(4), 509-511, 2006.
- [78] T. Munzner and P. Burchard. Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space. *Proc. VRML '95 Symp.*, 33-38, 1995.
- [79] J. Nielsen. Usability engineering at a discount. In *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*. Amsterdam:Elsevier, 394-401, 1989.
- [80] J. Nielsen. Finding usability problems through heuristic evaluation. In *Human Factors in Computing Systems CHI'92 Conference Proceedings*, New York:ACM Press, 373-80, 1992.
- [81] J. Nielsen. Risks of Quantitative Studies. Alertbox March 2004.
<http://www.useit.com/alertbox/20040301.html>
- [82] R. D. M. Page. TREEVIEW: An application to display phylogenetic trees on personal computers. *Computer Applications in the Biosciences* 12, 357-358, 1996.
- [83] X. Pan, H. Liu, J. Clarke, J. Jones, M. Bevan, and L. Stein. ATIDB: Arabidopsis thaliana insertion database. *NAR*31(4), 2003.
- [84] H. M. Parsons, C. Ludwig, U. L. Gunther, and M. R. Viant. Improved classification accuracy in 1- and 2-dimensional NMR metabolomics data using the variance stabilising generalised logarithm transformation. *BMC Bioinformatics*, 8:234, 2007.
- [85] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. LifeLines: Visualizing personal histories. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 221-227, 1996.
- [86] S. Pook, G. Vaysseix, and E. Barillot. Zomit: biological data visualization and browsing. *Bioinformatics*, 14(9):807-814, Nov. 1998.
- [87] K. Potosnak. Big paybacks from 'discount' usability engineering. *IEEE Software*, 107-9, 1990.

- [88] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey. *Human-Computer Interaction*. Addison-Wesley Publishing Company. 1994.
- [89] J. Preece, Y. Rogers, and H. Sharp. *Interaction design - beyond human-computer interaction*. John Wiley & Sons, Inc. 2002.
- [90] M. K. Raizada and S. D. Sarkissian. Potential of Gene Therapy Strategy for the Treatment of Hypertension. *Hypertension*, 47(1), 6-9, 2006.
- [91] R. Rao and S. K. Card. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. *Proc. ACM CHI94 Conference: Human Factors in Computing Systems*, 318-322, 1994.
- [92] G. G. Robertson, S. K. Card, and J. D. Mackinlay. Information visualization using 3D interactive animation. *Communications ACM*, 36, 57-71, 1993.
- [93] G. G. Robertson and J. D. Mackinlay. The document lens. *Proc. 1993 ACM User Interface Software and Technology*, 101-108, 1993.
- [94] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone Trees: animated 3D visualizations of hierarchical information. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 189-194, 1991.
- [95] S. F. Roth, P. Lucas, J. A. Senn, C. C. Gomberg, M. B. Burks, P. J. Stroffolino, A. J. Kolojechick, and C. Dunmire. Visage: a user interface environment for exploring information. *Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96)*, p.3, October 28-29, 1996.
- [96] W. W. Royce. Managing the development of large software systems: Concepts and techniques. In *Proceedings of IEEE WESTCON*. (Los Angeles, CA), 1-9, 1970.
- [97] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. 2nd Edition, Upper Saddle River, NJ: Prentice Hall, ISBN 0-13-790395-2, 2003.
- [98] K. Rutherford, J. Parkhill, J. Crook, T. Horsnell, P. Rice, M. A. Rajandream, and B. Barrell. Artemis: sequence visualization and annotation. *Bioinformatics* 16(10), 944-945, 2000.
- [99] B. R. Schatz and J. B. Hardin. NCSA Mosaic and the World Wide Web: Global Hypermedia Protocols for the Internet. *Science*, 256, 895-902, 1994.
- [100] P. Shannon, A. Markiel, O. Ozierand, N. Baliga, J. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, 13:2498-2504, 2003.

- [101] B. Shneiderman. Tree visualization with tree-maps: a 2-d space-filling approach. *ACM Transactions on Graphics*, Vol. 11, No. 1, 92-99, Sept 1990.
- [102] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualisation. In *Proceedings of IEEE Symposium on Visual Languages*, 336-343, 1996.
- [103] A. U. Sinha and J. Meller. Cinteny: flexible analysis and visualization of synteny and genome rearrangement in multiple organisms. *BMC Bioinformatics*, 2007.
- [104] L. Slaughter, K. L. Norman, and B. Shneiderman. Assessing users' subjective satisfaction with the Information System for Youth Services (ISYS). VA Tech Proc 3rd Mid-Atl. *Human Factors Conf.*, 164-170, 1995.
- [105] C. Soderlund, W. Nelson, A. Shoemaker and A. Paterson. SyMAP:A system for discovering and viewing syntenic regions of FPC maps. *Genome Research*, 16:1159-1168, 2006.
- [106] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization. *Mol. Biol. Cell* 9, 32733297, 1998.
- [107] L. D. Stein, C. Mungall, S. Q. Shu, M. Caudy, M. Mangone, A. Day, E. Nickerson, J. E. Stajich, T. W. Harris, A. Arva, and S. Lewis. The genetic genome browser: a building block for a model organism system database. *Genome Research* 12(10), 2002.
- [108] R. Stevens, C. Goble, P. Baker, and A. Brass. A classification of tasks in bioinformatics. *Bioinformatics*:17(2), 2001.
- [109] M. Stoll, A. E. Kwitek-Black, A. W. Cowley Jr., E. L. Harris, S. B. Harrap, J. E. Krieger, M. P. Printz, A. P. Provoost, J. Sassard, and H. J. Jacob. New Target Regions for Human Hypertension via Comparative Genomics. *Genome Research* 10(4), 473-482, 2000.
- [110] M. Tory and T. Moller. Evaluating Visualizations: Do Expert Reviews Work? *IEEE Computer Graphics and Applications*, v.25 n.5, 8-11, September 2005.
- [111] T. Toyoda, Y. Mochizuki, and A. Konagaya. GSCope: a clipped fisheye viewer effective for highly complicated biomolecular network graphs. *Bioinformatics*, 19, 437438, 2003.
- [112] E. R. Tufte. *The Visual Display of Quantitative Information*. Cheshire, CT, Graphics Press, 1983.
- [113] Q. N. Van, H. J. Issaq, Q. Jiang, Q. Li, G. M. Muschik, T. J. Waybright, H. Lou, M. Dean, J. Uitto, and T. D. Veenstra. Comparison of 1D and 2D NMR Spectroscopy for Metabolic Profiling. *J. Proteome Res.*, 7(2), 630-639, 2008.

- [114] H. Wang, S. Yanqi, A. J. Mackey, E. T. Kraemer, and J. C. Kissinger. SynView: a GBrowse-compatible approach to visualizing comparative genome data. *Bioinformatics*, 2006.
- [115] L. Wang, Y. Zhao, K. Mueller, and A. Kaufman. The Magic Volume Lens: An Interactive Focus+Context Technique for Volume Rendering. *IEEE Visualization*, 2005.
- [116] E. Wolf. Perilous Ideas: Race, Culture, People. *Current Anthropology*, 35: 1-7. p.227, 1994.
- [117] L. M. Work, H. Buning, E. Hunt, S. A. Nicklin, L. Denby, N. Britton, K. Leike, M. Odenthal, U. Drebber, M. Hallek, and A. H. Baker. Vascular bed-target in vivo gene delivery using tropism-modified adeno-associated viruses. *Molecular Therapy*, 13, 683-693, 2006.
- [118] M. Wu, C. Thao, X. Mu, and E. V. Munson. A fisheye viewer for microarray-based gene expression data. *BMC Bioinformatics* 7:452, 2006.
- [119] Y. Yang, L. Engin, E. S. Wurtele, C. Cruz-Neira, and J. A. Dickerson. Integration of metabolic networks and gene expression in virtual reality. *Bioinformatics*:21(18), 2005.
- [120] G. H. Zimney. Method in experimental psychology. New York:Ronald Press, 1961.
- [121] AceDB: <http://www.acedb.org>.
- [122] BioMart: <http://www.biomart.org>.
- [123] Chernoff face: <http://bradandkathy.com/software/faces.html>.
- [124] Cinema: <http://umber.sbs.man.ac.uk/dbbrowser/CINEMA2.1>.
- [125] Definition of Homolog: http://homepage.usask.ca/~ct1271/857/def_homolog.shtml.
- [126] Dendrogram: <http://www.mathworks.com/access/helpdesk/help/toolbox/stats/dendrogram.gif>.
- [127] DSI System: http://www.datasci.com/pdf/products/DSI_System_Brochure.pdf.
- [128] Electronic lab notebooks (ELNs): <http://www.scientific-computing.com/scwjunjul06elns.html>.
- [129] Encyclopedia Britanica Online. Academic Edition: <http://www.britannica.co.uk>.
- [130] FlyMine: <http://www.flymine.org>.
- [131] GenBank: <http://www.ncbi.nlm.nih.gov/Genbank>.
- [132] GeneChip: <http://www.roswellpark.org>.
- [133] HCI disciplines: <http://library.gsfc.nasa.gov/SubjectGuides/hcigraphic.gif>.
- [134] HeatMap: <http://en.citizendium.org/wiki/Bioinformatics>.

- [135] Human-Mouse Homology chromosome regions and genes; triple synteny Human-Mouse-Rat: <http://www.softberry.com/berry.phtml?topic=human-mouse>.
- [136] Ingenuity Pathway Analysis: <http://www.ingenuity.com>.
- [137] Interaction Design Encyclopedia: <http://www.interaction-design.org/encyclopedia/wimp.html>.
- [138] Java Web Start Technology: <http://java.sun.com/products/javawebstart>.
- [139] Jmol: an open-source Java viewer for chemical structures in 3D: <http://www.jmol.org>.
- [140] KMLE Medical Dictionary: <http://www.kmle.com>.
- [141] Medical Encyklopedia: <http://www.nlm.nih.gov>.
- [142] Normal distribution: <http://www.steve.gb.com/science/statistics.html>.
- [143] Online Mendelian Inheritance in Man (OMIM): <http://www.ncbi.nlm.nih.gov>.
- [144] Rat Genome Database (RGD): <http://rgd.mcw.edu>.
- [145] Rat tail blood pressure determination: http://www.gencompare.com/Blood_Pressure_Systems.htm.
- [146] Statistics Glossary by V. J. Easton and J. H. McColl's: http://www.cas.lancs.ac.uk/glossary_v1.1/main.html.
- [147] SyMAP: <http://www.agcol.arizona.edu/software/symap>.
- [148] The human chromosome 21 database: <http://chr21.molgen.mpg.de>.
- [149] The R Project for Statistical Computing: <http://www.r-project.org>.
- [150] The t-test: http://www.socialresearchmethods.net/kb/stat_t.php.
- [151] Tree-Maps: <http://www.cs.umd.edu/hcil/treemap-history>.

Appendices

Appendix A:

VisGenome and Ensembl: Usability of Integrated Genome Maps (DILS'08).

Appendix B:

CartoonPlus: A New Scaling Algorithm for Genomics Data (ICCS'08).

Appendix C:

VisGenome: visualisation of single and comparative representations (Bioinformatics'07).

Appendix D:

Genome Visualisation (HCI Engage'06).

Appendix E:

Usability of VisGenome and Ensembl - A User Study (Technical Report'07).

Appendix F:

Granularity of genomics data in genome visualisation (Technical Report'06).

Appendix G:

VisGenome User Manual.

Appendix H:

Ethics Committee Form.

Appendix I:

Participant Consent Form (Initial Quantitative User Study).

Appendix J:

Participant Consent Form (Mixed Paradigm User Study).

Appendix K:

Questionnaire (Initial Quantitative User Study).

Appendix L:

Workload Tests (Initial Quantitative User Study).

Appendix M:

Diary (Mixed Paradigm User Study).

Appendix N:

Interview Form (Mixed Paradigm User Study).

Appendix O:

Statistical Methods.

Appendix P:

Raw Data from Log Files (Mixed Paradigm User Study).

A: VisGenome and Ensembl: Usability of Integrated Genome Maps

Joanna Jakubowska¹, Ela Hunt², John McClure³, Matthew Chalmers¹, Martin McBride³, and Anna F. Dominiczak³

¹Department of Computing Science, University of Glasgow, UK

²Department of Computer Science, ETH Zurich, Switzerland

³BHF Glasgow Cardiovascular Research Centre, University of Glasgow, UK

asia@dcs.gla.ac.uk, hunt@inf.ethz.ch,

jdmc4w@clinmed.gla.ac.uk, matthew@dcs.gla.ac.uk

M.McBride@clinmed.ac.uk, ad7e@clinmed.gla.ac.uk

Abstract. It is not always clear how best to represent integrated data sets, and which application and database features allow a scientist to take best advantage of data coming from various information sources. To improve the use of integrated data visualisation in candidate gene finding, we carried out a user study comparing an existing general-purpose genetics visualisation and query system, Ensembl, to our new application, VisGenome. We report on experiments verifying the correctness of visual querying in VisGenome, and take advantage of software assessment techniques which are still uncommon in bioinformatics, including asking the users to perform a set of tasks, fill in a questionnaire and participate in an interview. As VisGenome offers smooth zooming and panning driven by mouse actions and a small number of search and view adjustment menus, and Ensembl offers a large amount of data in query interfaces and clickable images, we hypothesised that a simplified interface supported by smooth zooming will help the user in their work. The user study confirmed our expectations, as more users correctly completed data finding tasks in VisGenome than in Ensembl. This shows that improved interactivity and a novel comparative genome representation showing data at various levels of detail support correct data analysis in the context of cross-species QTL and candidate gene finding. Further, we found that a user study gave us new insights and showed new challenges in producing tools that support complex data analysis scenarios in the life sciences.

Key words: visualisation of large data sets, genome maps, genome visualisation, user study, QTL, comparative and functional genomics

1 Introduction

Data visualisation helps in the understanding of complex biological relationships, and is widely used in genomics [9, 11, 19, 20, 28], taxonomy [10], proteomics, and pathway analysis [29]. Genome data is usually served by a database system, as the amounts of data that need to be shown exceed by far the amount of RAM available on a user machine. Significant effort goes at design time into deciding how much data to fetch from the database and how to lay it out on the screen [2, 13, 27]. What usually does not happen in bioinformatics is recognising the evolving needs of the visualisation user. New data types and larger volumes cause not only purely technical problems, but also perceptual ones. Adding more data ‘tracks’ to a visualisation, accompanied by more colours and labels, may overwhelm the user, as discussed by Catarci [5], and shown in this paper. Also, the only reliable way of anticipating and discovering user interaction problems is via a user study [7]. This paper addresses the problem of reducing the visual overload in the face of large data volumes, an issue which lies on the boundary of database and visualisation research, via a user study carried out in a controlled environment. The results of this study are being fed into further development work, and are still providing food for thought.

The motivation behind the work we report on is the need to carry out comparative analyses of QTL¹, gene and protein expression and synteny in the human, the mouse and the rat, forming part of the search for genes causing cardiovascular disease, and done in collaboration between several research groups in the

¹ A quantitative trait locus (QTL) is a region of DNA that is associated with a particular phenotypic trait.

UK and abroad. We first tried to find a suitable visualisation, and carried out a short study of the available browsers [18]. We discovered that the development of most browsers was not accompanied by usability studies, or such studies have not been published. We also saw that none of the viewers allowed us to see the data the way we want to view them. Expressionview [9], for example, shows QTLs and micro array probes and no other data, so it was not suitable for our work. SyntenyVista [13] shows a comparative view of two genomes but is limited with regard to other data such as micro array probes. Since the work of the British Heart Foundation Cardiovascular Research Centre at Glasgow [21] and of our collaborators requires the analysis of data of high complexity, we decided to learn from the existing packages and produce yet another genome browser. What we found missing in most browsers was the fact that it was hard to see large and small objects at the same time, and that zooming was a limiting factor. In [11] the authors recently stated explicitly that Ensembl zooming is not as flexible as maps.google.com. Since the main representational problem in our mind is zooming, this is the major technical issue we addressed, and our work examines the use of improved zooming and its contribution to the ease of traversing the genome space. We hypothesise that improved zooming will offer both usability and cognitive benefits, and aim to prove that experimentally, by comparing VisGenome and Ensembl with respect to the ease of finding of large and small objects (QTLs and micro array probes).

This paper presents the following contributions. We summarise the design and results of a user study including 15 participants which demonstrated that the users are more successful in VisGenome than in Ensembl use [16] in the context of candidate gene analysis. Further, we discuss the findings from a user questionnaire, providing evidence that VisGenome is *perceived* to be easier to use than Ensembl. This is due to a combination of factors, including smooth zooming, provision of comparative genome views, and a simpler monochromatic display. The paper is structured as follows. Section 2 focuses on user studies in databases and bioinformatics, and Section 3 introduces VisGenome and Ensembl. Our user study design is presented in Section 4, and the results are described in Section 5. Section 6 gives a discussion, and Section 7 concludes.

2 Related Work

We first review some work spanning the areas of databases, visualisation and human computer interaction, and then summarise a number of bioinformatics user studies.

Catarci [5] was one of the first authors to convincingly argue the importance of user-centred design in the construction of user interfaces to database systems. Query construction is the focus of her work, and the design and testing process has to deliver interfaces that support efficient working and minimise user dissatisfaction and the need for assistance and maintenance. The main argument is that this can only be achieved via user-centred design, and requires the following: user involvement; a clear identification of user requirements, tasks and context; an appropriate split of functions between the user and the system; iterative design; and multidisciplinary competencies in the design team. To determine whether a system satisfies all user objectives, a formal evaluation needs to be carried out in a realistic context. As such evaluations are expensive and time-consuming, they are usually avoided, and the resulting systems are only judged in terms of correctness and functionality, and may well be suboptimal and cause user stress and additional costs to the organisation which commissioned them. One of the important points raised by Catarci is the issue of completeness and correctness of data representation. She finds that over-featured interfaces do not work well, as the complexity gets in the way of understanding the system and working out how to use it. Additionally, usability has an additional cost in terms of decrease in software production rate, and user satisfaction is never considered as an instrument to define the contract terms in software provision. As a result, also in the research context, usability issues are often ignored in favour of a narrow focus on selected information system aspects, such as performance or correctness.

A number of papers on the boundary of visualisation, e-science and database areas deal with provenance, data caching, and workflows, but address usability only in terms of user efficiency. VisTrails [4] solves the problem of visualisation from a database perspective, by managing the data and metadata of visualisation products. Workflows and provenance management are described in [22] and [6]. Here, a visualisation is used to allow the user to understand data provenance and modify existing analysis procedures (workflows). To our knowledge, no user studies have been published.

Recently, Jagadish and co-authors [15] broadened our understanding of the term usability in the context of database work. Starting from the observation that currently DBAs have to mediate between the user and the database, to hide the underlying system complexity, they draw an agenda of database usability challenges. They advocate the development of new database techniques which in their underlying design will focus on enabling direct interaction modalities for a database user. The future will be a WYSIWYG database with instantaneous-response interfaces, contextual displays, zooming and panning applying not just to maps but to all levels of database reality, including schemas, design activities, database evolution and provenance. To achieve that future, the authors propose a new presentation data model, which may be denormalised and will support direct user interaction, that is direct database creation, evolution, data manipulation, and structural changes to data. Some of the presentation modalities will include map mashups [14, 8], graph representations, multidimensional database facilities and tabular metaphors for data display. In data manipulation, the user interface will take advantage of a new simple algebra that will be easy to understand and intuitive to use. The proposed research scenario includes future user studies which will guide the development of both abstract models and practical database optimisations.

We now turn our attention to bioinformatics. In this area, only a small number of user studies have been published, while many application notes and other papers published in journals *Bioinformatics* and *BMC Bioinformatics* claim that the software is ‘user friendly’. For papers published in *Bioinformatics* between January 2000 and December 2007 the journal’s search facility delivers 284 hits for the query ‘user friendly’, two for ‘user study’, and 53 for ‘usability’. This may mean that most usability claims are not well founded. In one of the early papers mentioning the word ‘user’, Stevens et al. [26] presented a survey of bioinformatics tasks undertaken by biologists. They reported on new requirements which could stimulate the development of future applications, but did not conduct a user study. Wu et al. [28] reported on an electronic table that uses fisheye distortion. The table showing gene expression data was a subject of a pilot user study including five researchers completing a Questionnaire for User Interface Satisfaction (QUIS) [25]. Yang and colleagues [29] observed biologists interacting with a new software package and analysing experimental data, however, a formal study has not taken place. Graham and colleagues [10] presented an informal user study with biologists from the Royal Botanic Garden Edinburgh. The users carried out 12 tasks and used two prototypes of a visualisation tool. The authors received feedback from the participants and recognised that none of the prototypes was perfect and they should develop a new one which combined the existing two prototypes. These findings are similar to the views of the users in our study, and the feedback we obtained is reflected in our current engineering work.

3 VisGenome and Ensembl

VisGenome (VG) [17], see Figure 1 (left), shows single and comparative representations of the rat, the mouse and the human chromosomes at different levels of detail, and integrates data from Ensembl [11], locally produced lab results and [12]. It offers an overview of all rat, mouse and human chromosomes. After choosing a chromosome of interest, the user sees it in a new view with detailed data. The view supports interaction by mouse and keyboard, such as smooth zooming and panning [2] which is more flexible than seen in other browsers. The users can keep an area of interest in focus and choose the chromosome region by dragging the box enclosing the region or typing in the coordinates in an info panel. Then only the data in the selected area is displayed. The aim is to provide the context and allow the researchers to navigate the data at the same time. VG retrieves supporting web pages from Ensembl by invoking a link in a browser.

Ensembl (Ens) is probably the most popular system for mammalian genome analysis. It offers 17 different views, including ChromoView, ContigView, GeneView, MultiContigView, SNPView, and SyntenyView. In our experiment, biological and medical researchers used ContigView, MultiContigView and SyntenyView. ContigView, see Figure 1, shows different views of a gene, from broad chromosome context to fine nucleotide detail. These views are in separate horizontal frames, one below the other, and the user has to scroll as all views do not fit on a computer screen. There is also a chromosome overview facility, CytoView, shown in Figure 2. This view does not fit on the screen either and requires scrolling. In Ensembl data items are labelled and searching on names and coordinates is possible. Zooming uses buttons (Fig. 1, panels E and F). MultiContigView is an extension of ContigView and is meant to support comparative

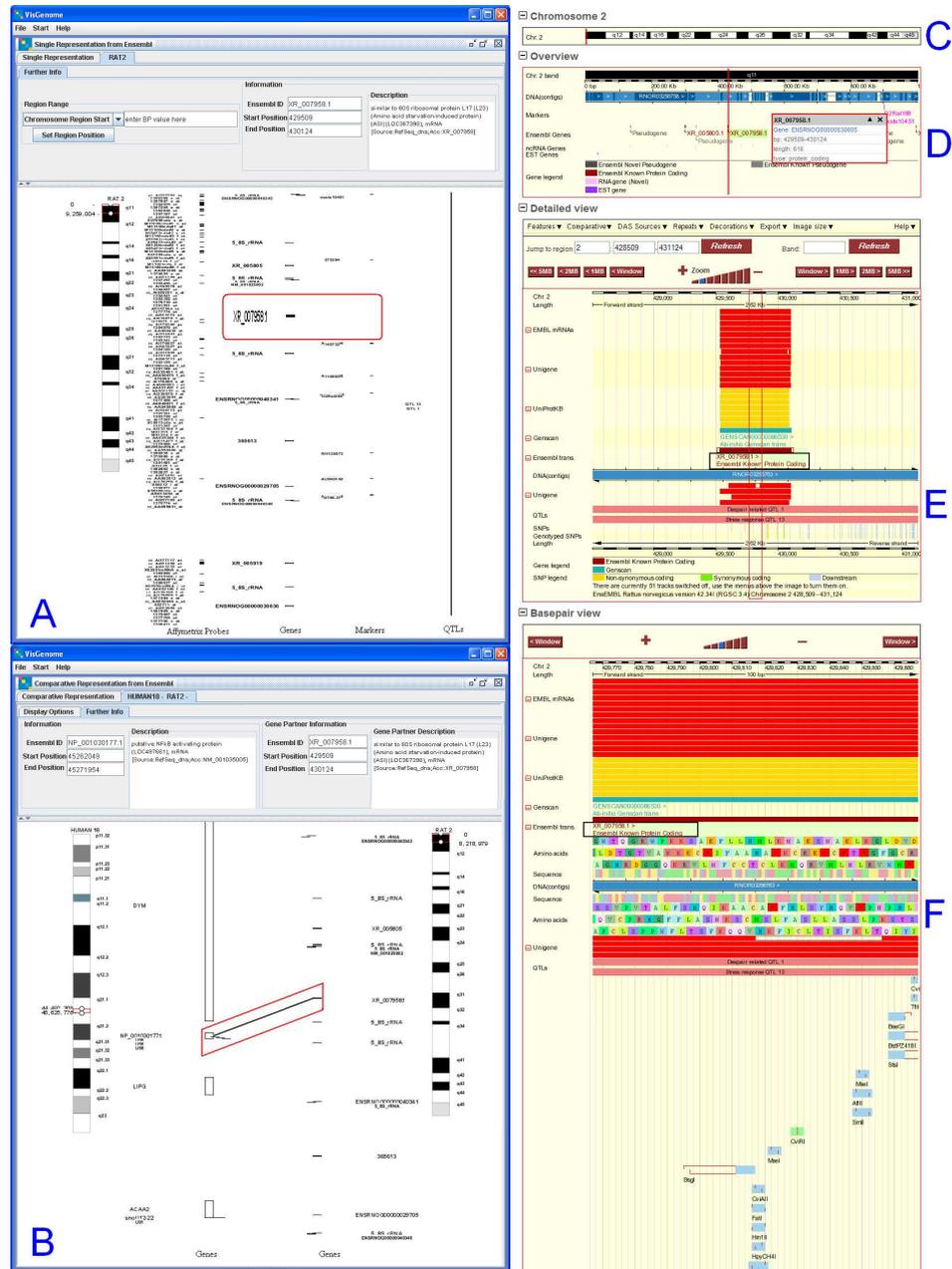


Fig. 1. Gene XR_007958.1 on rat chr. 2 in VisGenome and Ensembl, with the gene name and position in a frame superimposed on the screenshots. A: VisGenome, single chromosome view. B: VisGenome, comparative view of the rat chr. 2 and the human chr. 18. (C-F) Ensembl ContigView. (C) The entire chromosome, (D) An 'Overview' of a region of 1 Mbp, (E) The 'Detailed View' showing markers and genes, and (F) A 'Basepair View' showing protein translations

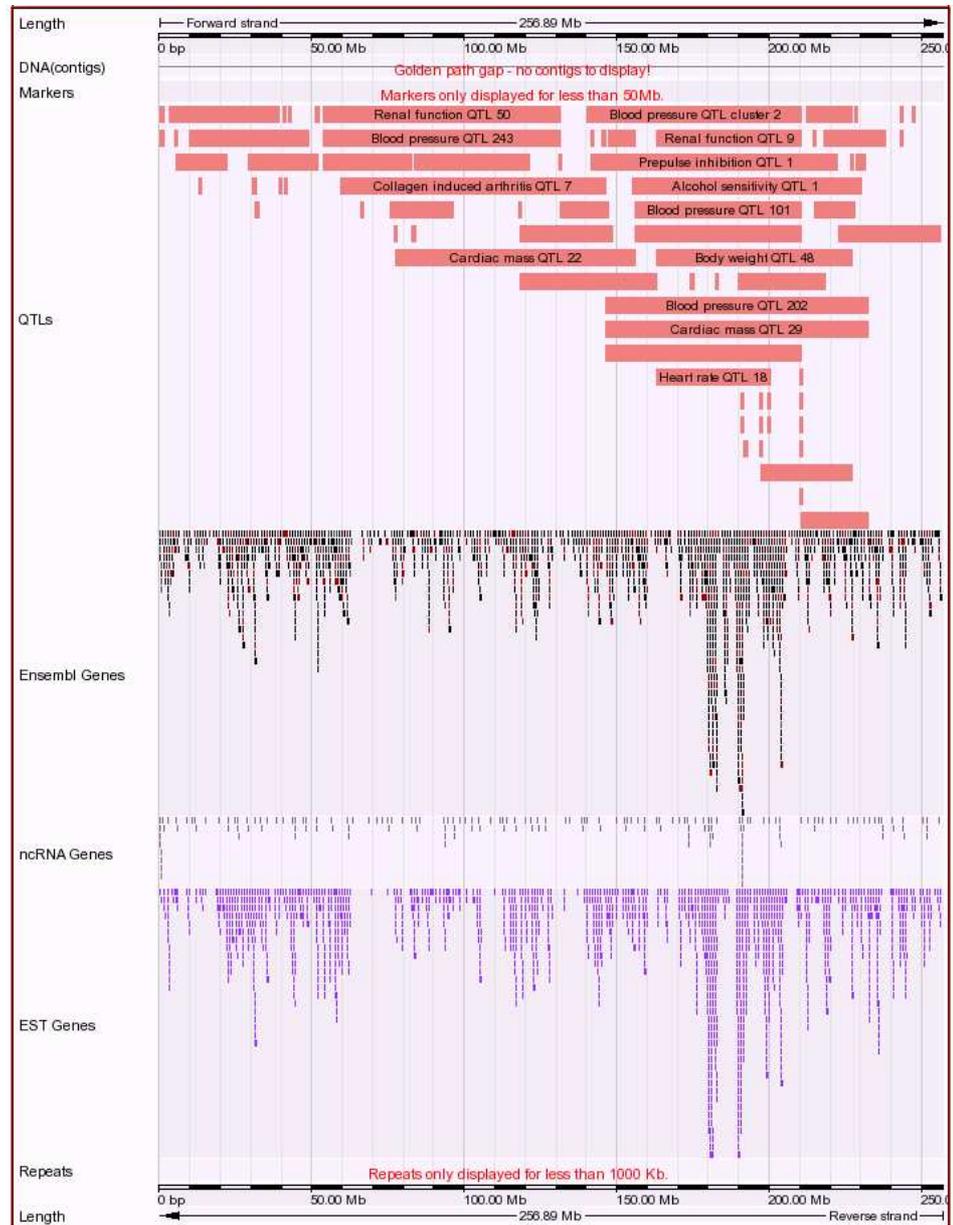


Fig. 2. Overview of rat chromosome 2 in Ensembl version 40

genome analysis. It displays genome annotation for several species. In SyntenyView a clickable high-level view of chromosomes with blocks of conserved synteny is shown.

Table 1. Example data sizes, Ensembl version 40, Aug 2006

Species	Chromosome	length	genes	microarray probesets	microarray probes	SNPs	QTLs
rat	2	250 Mbp	1413	1870	71,141	2740	~100

Data in Ensembl is stored in a relational database system and can be accessed via SQL or a Perl API. When the experiment was conducted, we accessed the database via JDBC and stored local experimental data and data from [12] in a local relational database. We visualised only genes, QTLs and microarray probes, and did not show SNPs or probesets, as those were not required. The requirement to show microarray probe mappings in three species increases the data size by at least a factor of 10, as each gene may have a matching micro array probe set, consisting of up to 10 probes, and each probe may have produced positive or negative results in a number of different experiments. The amount of data to be shown is significant, and is user-specific, as it may include arbitrary data sources, resulting from recent publications or experiments. Table 1 and Figure 2 give an idea of the number of items that have to be fetched from Ensembl to generate a chromosome overview, and make it clear that adding more data items and types will cause both performance and perceptual problems.

4 A User Study

The aim of the user study was to find out if new ways of visually querying the data, via mouse manipulation and zooming, are effective. Another question was whether the layout and colours we proposed supported the user in finding the data they are interested in. As our target users spend most of their time studying QTLs in the mouse, rat and human, we focus on supporting this activity, and ignore other aspects of tool use. As such work is carried out by a number of geneticists in five collaborating centres in the UK, and is poorly supported by existing tools, we wanted to see if VisGenome can facilitate it. We also wanted to gather additional feedback which would guide the development of VG. We compared Ens and VG, as Ens was the closest match to user requirements. Although the tools offer similar functions, Ens shows more data types than VG, as VG does not show sequence level data (view F in Figure 1) or gene structure (view E). VG was under our control, which allowed us to add private user data and make the study more realistic. Incorporating private data in Ensembl was not desirable, because of privacy concerns.

PARTICIPANTS. We first carried out a pilot experiment with two subjects from the Bioinformatics Research Centre (BRC) and five from the Western Infirmary (WI) in Glasgow. Finally, in the experiment we had 15 participants from the WI and the BRC. Six of them use Ens often (Ens Experts: Ex). Nine of them use different tools, such as BugView [20], UCSC GenomeBrowser [19] or AtIDB [23], or were from BRC and do not use genome browsers but know them from presentations (NonExperts: NEx). Three of the participants (Ex) previously took part in a one day Ens course.

METHODS. None of the biologists have used VG before the experiment. We gave a short presentation of VG to all subjects. Several researchers asked us to remind them first how Ens works and where to find information (three participants - NEx). We gave them a short introduction to Ens. Before the experiment, we offered the subjects the opportunity to carry out an experimental task in VG (for NEx also in Ens). We did not randomise task order and VG task came first. The order in which the tools were attempted is thus a confounding factor; although a positive effect on the performance for the second attempted tool (Ens) is the most likely consequence of this, Ens performance was not better than VG.

Prior to the study, two WI subjects had asked to see their experimental data. To that end, we created one version of VG for the majority of participants and two specific versions with private data. In those versions, micro array probes were coloured in both Single and Comparative Representations, see Figure 3. The aim was to receive more feedback from those subjects.

The experiment was divided into two parts (Ens and VG). We explained to the participants what we understand by Single and Comparative Representation and that VG offers Single and Comparative Representations, but in Ens the subjects have to decide if they would like to use MultiContigView or SyntenyView as Comparative Representation, and ContigView or any other Views as Single Representation. Some of the participants asked us if they can use BioMart [1] or RGD [24] (2 users) during the execution of Ens task. They could use all tools available from Ens pages. During the experiment the participants could give up if they thought that it was not possible to complete the task. The majority of the subjects attempted the tasks and only one person gave up and abandoned tasks T2 and T3, see below.

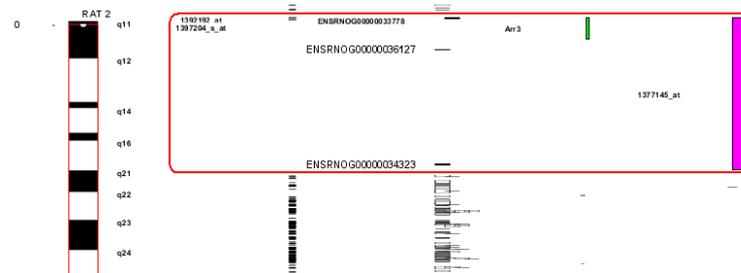


Fig. 3. VisGenome Single Representation for rat chr. 2. From left to right: chromosome overview, Affymetrix probes, genes, eQTLs and pQTLs [12], and Affymetrix probes from a user’s experiment

SEARCH TASKS. Rather than choose our own tasks, which might have created a bias in favour of VG, we asked our biological collaborators to recommend some common search tasks. The experiment was designed to model real-life data use, and follow the pattern of an ‘ecological study’ under real work constraints. This precluded the use of a fully controlled experiment methodology. The users defined three tasks, as follows.

- T1** Single Representation. Choose one of the rat, mouse or human chromosomes. Mark the whole chromosome and show all available data. Then choose the region between 100bp and 10,000,000bp and note the name of the first gene and the last Affymetrix probe inside the region.
- T2** Comparative Representation. Choose rat chr. 18 and human chr. 5. Zoom in and out to find any homologies between genes. Then choose one of the homologies and read out the names of the homologous genes.
- T3** Single Representation. Choose one of the rat chromosomes. Find the longest QTL. Then zoom on it and write down the names of the genes which are the closest to the beginning and the end of the QTL.

We captured screen usage as videos, recorded the time used for each task in minutes (STi, search time), and counted the number of mouse clicks (NoMc) for all tasks in VG and Ens. On finishing the tasks, the subjects filled in a questionnaire and participated in an interview.

5 Experimental Results

The results are quite surprising. The researchers who use Ens frequently are often unsuccessful in task execution. The experts encounter no problems in their everyday work which focuses on a chromosome fragment. However, when they examine similar data in a different part of the chromosome, they encounter problems. We also found that some of the zooming mechanisms in VG were hard to use and that the subjects prefer mouse clicking to dragging. The researchers want to see large amounts of data, but when they are looking for a particular object, they prefer to see only a small part of the data under investigation. ACCURACY AND TASK COMPLETION. Figure 4 shows that T2, the only task involving comparative genome representation, was more successful with VG (100%) than with Ens (60%, 9 subjects). In T3 53% of attempts were successful in VG (8 subjects), while in Ens the success rate was 0. In T1 we note 20% success rate in VG and 0% in Ens. Using the two-sided sign test (where 0=both/neither successful; 1=VG

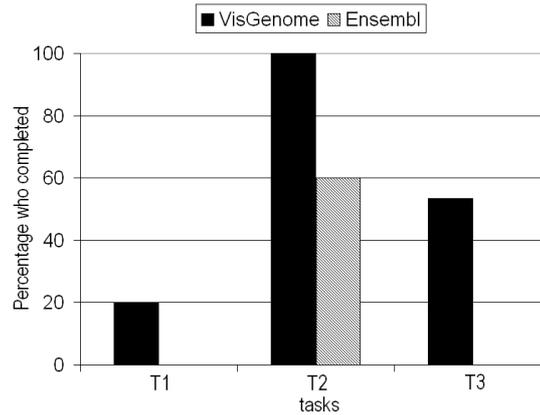


Fig. 4. Percentage of subjects (out of 15) who completed each task

success but Ens not; -1=Ens success but VG not) as an alternative to McNemar’s test [3] the success rate for VG was significantly greater for both T2 ($P=0.0313$) and T3 ($P=0.0078$), but not for T1 ($P=0.25$). The null hypothesis for these tests was that the proportion of successes was the same for both VG and Ens, and the alternative was that they were not. Completion rates were higher in VG than in Ens for all tasks, particularly for T2 and T3. This may be due to the fact that Ens is a much richer interface, with many more options and controls and represents more data. Possibly, the subjects were not able to find out how to generate comparative genome views, or were getting lost while learning to use the system.

TIME TO FINISH. Time was measured in minutes. The biologists who completed the tasks had mean of $T1=5.69'$ (StDev=1.39'), $T2=3.58'$ (StDev=1.17'), and $T3=5.29'$ (StDev=0.97') in VG and mean $T2=2.83'$ (StDev=1.76') in Ens. As no one completed T1 and T3 in Ens, statistics were calculated only for T2. In T2 in Ens and VG 9 researchers correctly completed both tasks. As the differences in times were not normally distributed, the Wilcoxon signed rank test was used ($P=0.554$). We realised that Ex used both tools differently than NEx. Ex usually wanted to see more information, got interested in the data, while NEx subjects just wanted to complete the task. Ex tried to find and show all possible answers they knew, and explore while doing the task. If there were several ways of doing the tasks in Ens they wanted to show all the solutions. In T2, for example, it was enough to show two genes in VG and Ens, and most NEx did that and finished quickly. Most Ex performed T2 and then explored MultiContigView to see more information about homologous genes, which took more time. Users behaved similarly in T3, however nobody succeeded in Ens. NEx showed Affymetrix probes in ContigView, while Ex used FeatureView and looked at the detail. There were also slight differences in server response times for Ens which might have influenced the speed of data analysis. Overall, in T2 there was little difference in task execution time between Ens and VG.

MOUSE CLICKS. Those who completed the tasks had the means of $T1=53$ (StDev=9.54), $T2=51.07$ (StDev=26.65), and $T3=74.38$ (StDev=13.38) NoMC in VG, and the mean for $T2=23$ (StDev=18.93) NoMC in Ens. Only T2 mouse clicks were analysed, due to non-completion in Ens for T1 and T3. 9 subjects completed T2 with both VG and Ens, and despite the mean number of clicks being larger in VG than in Ens, there was no significant difference in NoMC, possibly due to the small sample size. One Ex had a very large NoMC (138) for VG, and only 19 for Ens. This shows that mouse manipulation in VG needs getting used to, as panning and zooming require keeping the left/right mouse button down and moving the mouse at the same time left/right or up/down, and the left/right movement is not offered by many similar applications where clicking on zoom bars is used instead, and smooth zooming is not widely used. This is a potential problem, however, most subjects learned how to use the mouse quickly. On the other hand, Ex often clicked to see additional information and some of NEx clicked because they wanted to find the solution and they were not sure where they had to look for it. This contributed to a large NoMC in some Ex as well as NEx.

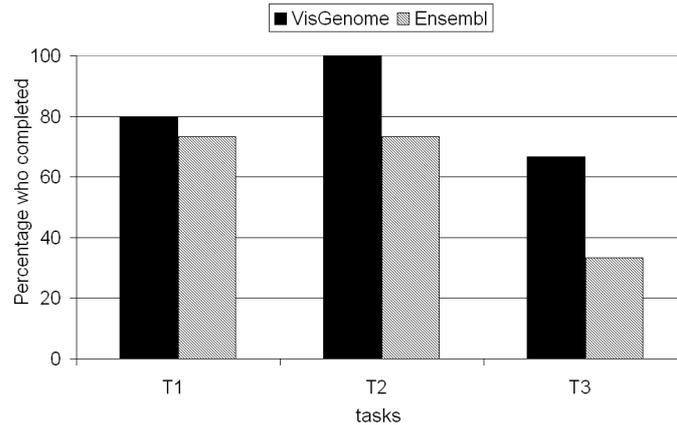


Fig. 5. Percentage of subjects (out of 15) who finished each task with errors

6 Discussion

6.1 User Study

In T1 we saw that the participants were looking for Affymetrix probes and couldn't find them. However, the main cause of failure in T1 was that the subjects made mistakes, e.g. typed 1 Mbp instead of 10 Mbp. In VG the subjects frequently forgot to mark the whole chromosome to show all available data or marked half of the chromosome instead of the whole. In Ens a number of users entered the coordinates and marked 'Region' instead of 'Base pair', and some did not use the overview offered by Ens but tried to mark the whole chromosome in ContigView. This usually crashed the web browser and required a restart.

T3 required showing the longest QTL. In a chromosome with many small QTLs, the subjects could not decide which QTL to choose (four subjects). We suggested that they carry out the task for any of the QTLs. The same solution was suggested where several long QTLs appeared to be of similar length. 8 researchers were successful in T3 in VG. The most frequent mistake in the unsuccessful attempts in VG was choosing a complex of QTLs instead of one QTL. In Ens the subjects usually attempted to mark the entire chromosome, and only one person succeeded without crashing the browser. Some subjects tried viewing the chromosome in units of one 1 Mbp but gave up after recognising that this would take too long. One user tried to use BioMart and RGD, but this did not help. Most subjects did not realise that the view shown in Ens is not the whole chromosome but a small part of it. Several subjects chose a chromosome, clicked on it, viewed ContigView, looked down the screen to find QTLs and saw that they were all longer than the area shown in the browser, and did not know what to do to see the entire length of the QTLs.

When we analyse both correct and partially erroneous task completions, see Figure 5, we see a different view of the experiment. 11 users finished T1 and T2 in Ens and 5 users finished T3 in Ens. Similarly, for VG the completion rate improved. T1 was completed by 12 users and T3 by 10.

6.2 Lessons Learnt

Although the use of zooming helped users, and new visualisation features required some learning, we suggest that the experiment highlights another significant issue to be addressed in future development: *high error rates in data selection and query specification*. The benefits of solving this problem may well outweigh those arising from new variations on and easy learning of features such as zooming and panning. Error rates are possibly due to suboptimal menus and selection boxes, or to the fact that users find it easier to use simple interfaces with fewer options, see [5], than complex ones which offer more functionality.

We note user training is required for both VG and Ens. Although zooming and panning by mouse manipulation was classified as something very intuitive and natural, at the beginning of the VG experiment most subjects were confused and disappointed that they had to remember which button and which direction to use to zoom or pan. A possible solution to this problem would be to offer visual shortcuts to

zooming, as seen in maps.google.com. While some users suggested that new visualisation techniques could be bad because biologists are not familiar with them, some said that acceptance depends on the implementation. A small number of subjects (2) suggested zooming with buttons instead of mouse manipulation and were disappointed because of the lack of scrolling.

VG supports local as well as cross-species QTL and gene expression analysis. This additional functionality offered by our application is essential to the work of our target users. In this context the use of colour will require further research, but our guess is that, based on our questionnaire, see [16], Ens offers too many colours, which is confusing to the user and makes the display hard to read. A possible extension of this work would examine the use of various layout and colouring options to arrive at solutions suitable for most users and giving the user some flexibility in layout, colour and interactivity adjustment.

Web interaction paradigms supported by AJAX (Asynchronous JavaScript and XML) are an alternative way of adding interactivity to a web-based genome map. These technologies are orthogonal to the issues of usability. We envisage that based on this study and further user studies we are planning, one could develop improved AJAX-based genome browsers which offer more interactivity and are more appropriate in the context of comparative genomics.

We confirm the findings reported in [5] about the high cost of usability experiments. The ethics application for this experiment was placed in May 2006. The user study was then refined in the summer of 2006 and conducted between August and start of December 2006. Some of the intervening time was spent on data integration tasks and some on related reading. Data analysis and writing up of the results (from screenshot recordings and questionnaires) took about three months. This represents around 10 months of elapsed time for one PhD student, and about 1-3 hours per user. We believe the time was well spent.

7 Conclusions

We presented a user study comparing VisGenome and Ensembl in the context of comparative genome analysis. We found that in our experimental setup which targets the analysis of QTLs, synteny and gene expression, the subjects were more successful in using VG than in Ens. VG was preferable in some aspects, as it had a simpler interface, showed less data and had fewer controls. All participants liked techniques they know, such as scrolling and panning, and needed time to adapt to new solutions, such as mouse driven panning and zooming. The study shows that there is still large scope for the application of known visualisation techniques to bioinformatics data. Useful solutions, like semantic zooming offered by maps.google.com, could be very useful and should be tested in biomedical work. In particular, this study shows the great potential for usability improvement via a user study.

During the study a list of user suggestions and requests was gathered and ongoing work is addressing those, as well as exploring ways to reduce error rates in data selection and query specification. The next version of VG will be evaluated differently. We will allow the users to see their data and navigate through it. This time, instead of specified tasks, the users will use VG in a real work scenario. We will observe how the subjects interact with VG and what kind of tools and information they use. VisGenome is now usable and can be downloaded from www.dcs.gla.ac.uk/~asia/VisGenome. Full details of our experiment can be found in [16].

Acknowledgments. We thank all the participants for their contributions and Helen Purchase for advice in user study design. EH is an EU Marie Curie fellow, JJ is funded by the MRC, UK (grant to EH and MC), and AFD is funded by the BHF Chair and Programme Grant (BHFPG/02/128) and the Wellcome Trust Cardiovascular Functional Genomics Initiative (066780/2/012).

References

1. Ensembl BioMart. <http://www.ensembl.org/biomart/martview>.
2. B. B. Bederson, J. Grosjean, and J. Meyer. Toolkit Design for Interactive Structured Graphs. *IEEE Trans. Software Eng.*, 30(8):535–546, 2004.
3. J. M. Bland. *An Introduction To Medical Statistics*. OUP, Oxford, 2000.
4. S. P. Callahan et al. VisTrails: visualization meets data management. In *SIGMOD Conference*, pages 745–747, 2006.
5. T. Catarci. What Happened When Database Researchers Met Usability. *Inf. Syst.*, 25(3):177–212, 2000.
6. Susan B. Davidson et al. Provenance in Scientific Workflow Systems. *IEEE Data Eng. Bull.*, 30(4):44–50, 2007.
7. A. Dix, J. Finlay, G. Abowd, and R. Beale. *Human-Computer Interaction*. Prentice Hall, 2004.
8. R. Ennals and M. N. Garofalakis. MashMaker: mashups for the masses. In *SIGMOD Conference*, pages 1116–1118, 2007.
9. G. Fischer et al. Expressionview: visualization of quantitative trait loci and gene-expression data in ensembl. *Genome Biology*, 4(R477), 2003.
10. M. Graham et al. A Comparison of Set-Based and Graph-Based Visualisations of Overlapping Classification Hierarchies. *Proc. AVI 2000*, pages 41–50, May 23-26 2000.
11. Tim J. P. Hubbard et al. Ensembl 2007. *Nucleic Acids Research*, 35(Database-Issue):610–617, 2007.
12. N. Hubner et al. Integrated transcriptional profiling and linkage analysis for identification of genes underlying disease. *Nature Genetics*, 37:243–253, 2005.
13. E. Hunt, N. Hanlon, D. Leader, H. Bryce, and A. F. Dominiczak. The Visual Language of Synteny. *OMICS*, 8(4):289–305, 2004.
14. E. Hunt, J. Jakubowska, C. Boesinger, and M. C. Norrie. Defining Mapping Mashups with BioXMash. *Journal of Integrative Bioinformatics*, 4(3):64, 2007. Proceedings of 4th Integrative Bioinformatics Workshop, Gent, Belgium.
15. H. V. Jagadish et al. Making database systems usable. In *SIGMOD Conference*, pages 13–24, 2007.
16. J. Jakubowska et al. Usability of VisGenome and Ensembl - A User Study. Dept of Comp. Sci., University of Glasgow, 2007. http://www.dcs.gla.ac.uk/publications/PAPERS/8510/VG&Ens_TechRep.pdf.
17. J. Jakubowska, E. Hunt, M. Chalmers, M. McBride, and A. F. Dominiczak. VisGenome: visualization of single and comparative genome representations. *Bioinformatics*, 23(19):2641–2642, 2007.
18. J. Jakubowska, E. Hunt, and M. J. Chalmers. Granularity of genomics data in genome visualisation. Dept of Comp. Sci., University of Glasgow, 2006. <http://www.dcs.gla.ac.uk/publications/PAPERS/8212/AsiaElaMatthewCHI06.pdf>.
19. D. Karolchik et al. The UCSC Genome Browser Database. *Nucleic Acids Research*, 31:51–54, 2003.
20. D. P. Leader. BugView: a browser for comparing genomes. *Bioinformatics*, 20:129–130, 2004.
21. M. W. McBride, D. Graham, C. Delles, and A. F. Dominiczak. Functional genomics in hypertension. *Curr Opin Nephrol Hypertens*, 15(2):145–51, March 2006.
22. O. Biton others. Zoom*UserViews: Querying Relevant Provenance in Workflow Systems. In *VLDB*, pages 1366–1369, 2007.
23. X. Pan et al. ATIDB: Arabidopsis thaliana insertion database. *Nucleic Acids Research*, 31(4), 2003.
24. D. Pasko. Overview of Rat Research Today. May 2003. <http://www.rgd.mcw.edu>.
25. L. Slaughter et al. Assessing users’ subjective satisfaction with the information system for youth services (isys). *VA Tech Proc 3rd Mid-Atl. Human Factors Conf.*, pages 164–170, 1995.
26. R. Stevens et al. A classification of tasks in bioinformatics. *Bioinformatics*, 17(2), 2001.
27. E. Tanin et al. Browsing large online data tables using generalized query previews. *Inf. Syst.*, 32(3):402–423, 2007.
28. M. Wu et al. A fisheye viewer for microarray-based gene expression data. *BMC Bioinformatics*, 7(452), 2006.
29. Y. Yang et al. Integration of metabolic networks and gene expression in virtual reality. *Bioinformatics*, 21(18), 2005.

B: CartoonPlus: A New Scaling Algorithm for Genomics Data

Joanna Jakubowska^a, Ela Hunt^b, and Matthew Chalmers^a

^aDepartment of Computing Science, University of Glasgow, UK

^bDepartment of Computer Science, ETH Zurich, Switzerland
asia@dcs.gla.ac.uk, hunt@inf.ethz.ch, matthew@dcs.gla.ac.uk

Abstract. We focus on visualisation techniques used in genome browsers and report on a new technique, CartoonPlus, which improves the visual representation of data. We describe our use of smooth zooming and panning, and a new scaling algorithm and *focus on* options. CartoonPlus allows the users to see data not in original size but scaled, depending on the data type which is interactively chosen by the users. In VisGenome we have chosen genes as the basis for scaling. All genes have the same size and all other data is scaled in relationship to genes. Additionally, objects which are smaller than genes, such as micro array probes or markers, are scaled differently to reflect their partitioning into two categories: objects in a gene region and objects positioned between genes. This results in a significant legibility improvement and should enhance the understanding of genome maps.

Key words: Genome Visualisation, Visualisation Techniques, Scaling Algorithm, Large Data Sets.

1 Introduction

Medical researchers find it difficult to locate the correct biological information in the large amount of biological data and put it in the right context. Visualisation techniques are of great help to them, as they support data understanding and analysis. We reported our findings from a survey of visualisation techniques used in genome browsers in [8]. We developed a prototype of a new genome browser, VisGenome, which uses the available techniques. VisGenome [9] was designed in cooperation with medical researchers from a hospital. We found that the majority of genome browsers show only a selection of data for one chromosome. This is obvious, because the amount of available information is so large that it is impossible to show all data in one view. Expressionview [4], for example, shows QTLs ¹ and micro array probes and no other data. Some of the tools, such as Ensembl [6], show many types of data but use a number of different data views, which make the users disoriented and lost in the tool and data space. Moreover, Ensembl shows as much information as possible in one view, instead of offering a view or a panel with additional information. A large number of genome browsers show only a chromosome and do not allow one to see a comparison of two chromosomes from different species. Exceptions include SyntenyVista [7] and Cinteny [15] which show a comparative view of two genomes but are limited with regard to other data, such as micro array probes. On the other hand, SynView [17] visualises multi-species comparative genome data at a higher level of abstraction.

We aim to find a solution which clearly presents all the available information, including all relevant information the biologists wish to see. We aim to find a solution for data analysis which overcomes both representational and cognitive problems.

Here, we describe single and comparative genome representations, see Figure 1 and 2. A single representation is a view which shows data for one chromosome. A comparative representation illustrates relationships between two or more chromosomes.

Our contribution is a scaling algorithm which we call CartoonPlus. CartoonPlus allows the users to see data more clearly by choosing one kind of data as basis and scaling other data types in relationship to the basis. The solution does not show data in its natural size but allows one to see relationships between different kinds of data more clearly, especially in a comparative representation.

The paper is organised as follows. Section 2 provides the background about visualisation techniques and their usefulness for medical researchers. Section 3 introduces the visualisation techniques we used in VisGenome and provides details of our new algorithm. We discuss our work in Section 4 and the last section concludes.

¹ A quantitative trait locus (QTL) is a part of a chromosome which is correlated with a physical characteristic, such as height or disease. Micro array probes are used to test gene activity (expression).

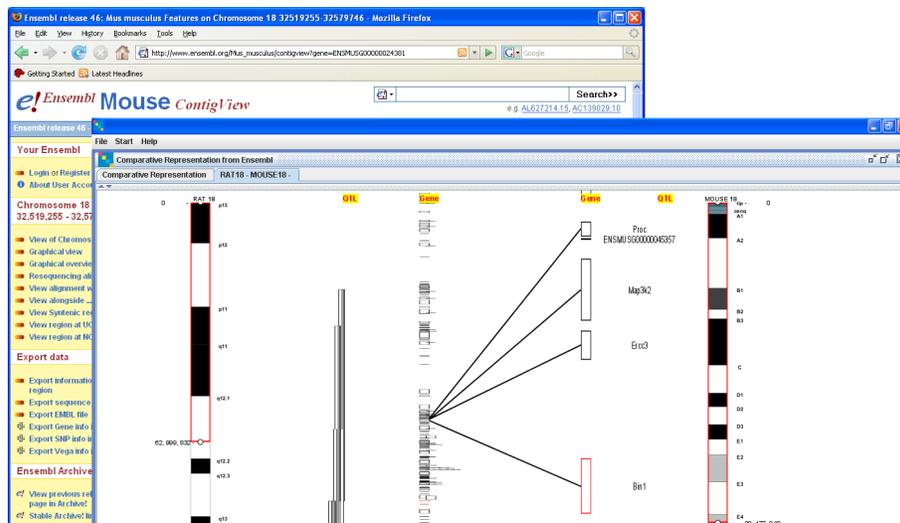


Fig. 1. The comparative representation for the rat chromosome 18 and the mouse chromosome 18. The gene Bin1 in the mouse chromosome 18 is in focus. The background shows additional information from Ensembl for Bin1, activated by clicking on the gene.

2 Related Work

This section examines existing visualisation techniques used in genomics data representation and clarifies why a new scaling algorithm is necessary.

A variety of scientific visualisation techniques are available and could be used for genomics. 2D techniques are very common in gene data visualisation and 3D techniques are rarely used [8]. An exception is [13] which uses a 3D model of the data. In the following we discuss the techniques used in 2D applications.

Fisheye [5] shows detail for an element and its neighborhood, but only an overview for the other elements. It is used in a number of graphical applications, for example for photo corrections, but it is hardly used in biology, with the exception of Wu [12] who used fisheye to show tables representing micro array results. Magic lenses [16] allow the user to transform the data and display extra information, see Zomit [14]. The majority of genome browsers offer scrolling and zooming [1] which are both easy to use. Zooming by buttons is well known and used by the medical researchers. Ensembl [6] uses this kind of zooming. BugView [11] also uses zooming by buttons which makes an impression of smooth zooming. Cartoon scaling is applied to biological data in [7]. The technique deforms the original data and makes it easier to read. SyntenyVista shows all genes in the same size and this makes it clear which genes share a homology link. A true physical representation of genes causes some of them to overlap and the users often cannot precisely see the genes connected by a homology link. This motivated us to design an improved algorithm for scaling for different kinds of data, and not only for genes. Our new algorithm, CartoonPlus, makes the display of biological data clearer in both single and comparative representations. It makes it easy to see which genes and QTLs share a homology link in a comparative representation and highlights differences and dependencies between different kinds of data in a single representation. Objects that are larger than a basis object form one category. Another category consists of objects smaller than the basis or lying in between basis objects. Those objects contained within a basis object are treated differently than the objects in between.

3 Visualisation extensions

VisGenome loads QTLs, genes, micro array probes, bands, and markers, and pairs of homologies from Ensembl. It shows single chromosomes or comparisons of two chromosomes from different species. The application uses the visualisation metaphors and algorithms offered by Piccolo [2]. Piccolo puts all zooming and panning functionality and about 140 public methods into one base object class, called PNode. Every

node can have a visual characteristic, which makes the overall number of objects smaller than in other techniques which require two objects, an object and an additional object having a visual representation, as in Jazz [2]. A Jazz node has no visual appearance on the screen, and it needs a special object (visual component), which is attached to a certain node in a scene graph and which defines geometry and color attributes. Piccolo supports the same core feature set as Jazz (except for embedded Swing widgets), but it primarily uses compile-time inheritance to extend functionality and Jazz uses run-time composition instead. Piccolo supports hierarchies, transforms, layers, zooming, internal cameras, and region management which automatically redraws the portion of the screen that corresponds to objects that have changed.

In the continuation of the section, we present a new scaling algorithm, CartoonPlus, and then we outline other known visualisation techniques which we implemented.

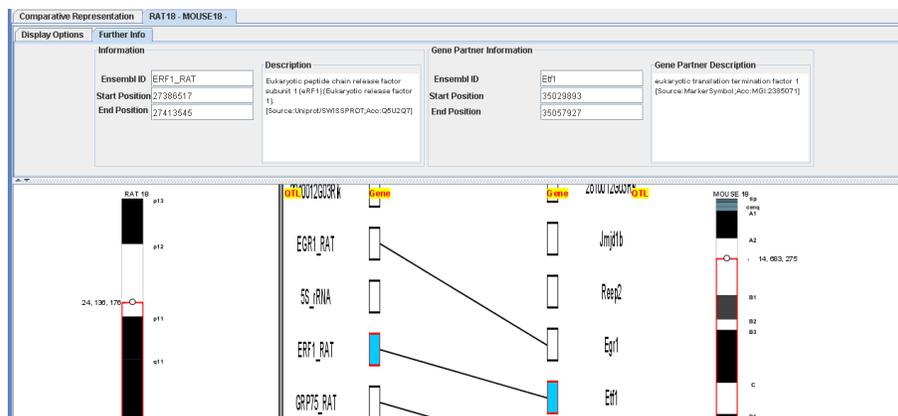


Fig. 2. The comparative representation for the rat chromosome 18 and the mouse chromosome 18. The data is scaled by the scaling algorithm which makes all genes the same size and QTL size depends on genes. Genes ERF1_RAT and Etf1 are linked by a homology line and marked in blue.

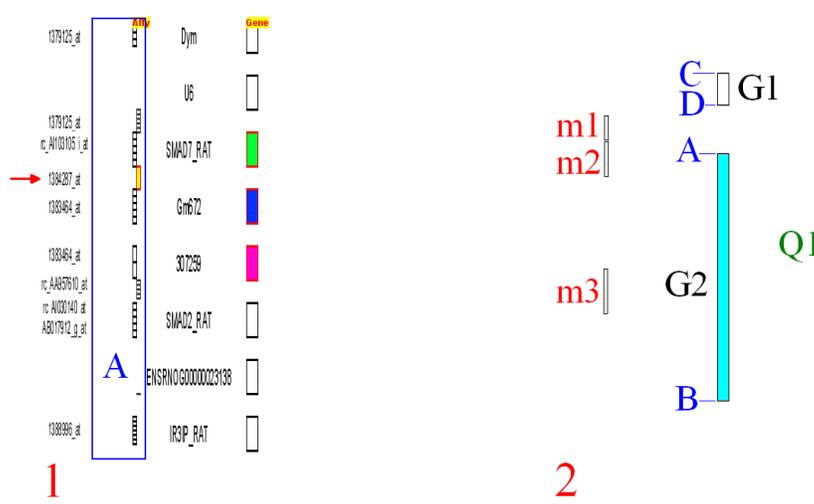


Fig. 3. 1: Single representation for the rat chromosome 18. Three genes (SMAD7_RAT, Gm672, 307259) and one micro array probe set (1384287_at, see arrow) are coloured by different colours, selected by the user interactively. 2: CartoonPlus algorithm (see Figure 4). G2 is a gene which begins at A and ends at B. m1, m2, and m3 are elements smaller than G2 and Q1 is bigger than G2.

Scaling Algorithm: We developed a scaling algorithm for arbitrary genomics data which extends existing

```

1 CartoonPlus() {
2   for(gene in GENES) {
3     ResizeAndPaint(gene)
4     ScaledMarkersBetween = GET_MARKERS_BETWEEN()
5     for (each marker in ScaledMarkersBetween)
6       ResizeAndPaint(marker)
7     ScaledMicroArrayProbesBetween = GET_MICRO_ARRAY_PROBES_BETWEEN()
8     for (each micro_array_probe in ScaledMicroArrayProbesBetween)
9       ResizeAndPaint(micro_array_probe)
10    ScaledMarkers = GET_MARKERS_IN()
11    for (each marker in ScaledMarkers)
12      ResizeAndPaint(marker)
13    ScaledMicroArrayProbes = GET_MICRO_ARRAY_PROBES_IN()
14    for (each micro_array_probe in ScaledMicroArrayProbes)
15      ResizeAndPaint(micro_array_probe)
16    ScaledQTLs = GET_QTLS_FOR_GENE()
17    for(each QTL from ScaledQTLs)
18      if (QTL.end>D AND QTL.end<=B)
19        ResizeAndPaint(QTL)
20        delete(QTL from ScaledQTLs)
21  }
22 }
23 GET_MARKERS_BETWEEN() {
24   for(marker in MARKERS)
25     if(marker.start>=D AND marker.end<=A)
26       markers.add(marker)
27   return(markers)
28 }
29 GET_MARKERS_IN(){
30   for(marker in MARKERS)
31     if((marker.start<=A AND marker.end>A) OR (marker.start>A))
32       markers.add(marker)
33   return(markers)
34 }
35 GET_QTLS_FOR_GENE(){
36   for(QTL in QTLs)
37     if(QTL.start>D AND QTL.start<=B)
38       QTLs.add(QTL)
39   return(QTLs)
40 }

```

Fig. 4. CartoonPlus algorithm. Hierarchy of object sizes: chromosome \geq QTL \geq gene \geq marker and micro array probe.

solutions. SyntenyVista [7] scaled genes only in a comparative representation. We offer scaling for all data, in both single and comparative representations, see Figure 2 and 3. Previous algorithms were constrained, while the new one scales multiple data types together, with reference to the basis. A user chooses the basis for scaling and then other elements are scaled in relationship to the chosen data type. In the current prototype we chose genes as a basis, so we scale all genes to the same size. An extension of this work is to allow the user to change the basis for scaling interactively. The algorithm looks at other types of data which are smaller or larger than genes, such as markers, micro array probes, or QTLs, and scales them accordingly. We divide all elements smaller than genes into two groups: elements which are in a gene region and elements which are in the region between two genes, see Figure 3.1.A. For each type of data holding items smaller than the basis, we create a column holding elements which are situated within the gene boundaries and a second column containing elements which are situated between two genes. For all elements which are in the gene region, we choose the same size for each element, and the same applies to all elements which are in the area between genes. The size of the elements depends on their number in a gene region. This means that if in an area of a gene there is only one marker, it has the same height as the gene, but if there are 10 markers, they together have the same size as the gene (each marker is set to 1/10th of gene height). When an element is on a gene boundary, it is partially in a gene region and partially between two genes, and we situated it in the gene region. We also scale elements like QTLs which are bigger than genes. We look where a QTL begins and ends and we paint it starting at the gene where

it begins and ending at the gene where it finishes. The solution allows us to present clearly a homology between genes in a comparative representation and additionally to show relations between micro array probes, markers, genes, and QTLs in two species.

Figure 4 outlines the scaling algorithm. All genes, markers, micro array probes and QTLs are stored in hashtables. The algorithm iterates over all genes (line 2). First we scale markers and micro array probes which are between genes (the previous gene and the current one), see Figure 3.2 object m1 between G1 and G2. Then we scale markers and micro array probes with a start coordinate before the gene and end coordinate inside the gene or start coordinate inside the gene region, see Figure 3.2 objects m2 and m3, and Figure 4 lines 4-15. Then we place QTLs which begin inside the gene region or in the region between a previous gene and the current gene, see Figure 3.2 object Q1. For each gene we check as well where the end coordinate of a QTL is, and, depending on this, we paint the element. In the pseudo-code we used function `ResizeAndPaint` which for basis data gives all elements the same size. For small objects, such as m1, m2, or m3, function `ResizeAndPaint` calculates how many elements are in the gene area or in the area between genes, and divides the area by the number of elements and then the elements are painted in the calculated size. For large elements, `ResizeAndPaint` calculates the height of the elements as the beginning of the gene where the QTL starts and end of the gene where it ends. If a QTL begins or ends between genes, the function takes the end of the previous gene or start of the next gene as its coordinates.

Navigation: We offer “overview and detail” views which are manipulated by mouse and keyboard interaction. At the beginning the users see an overview of all chromosomes and can choose the one they would like to see in detailed view. When they see all data for a selected chromosome, the tool gives them the possibility to see an overview of all data, but also details for each part of the data. The users can mark a region which is interesting for them and interact only with the selected part. To make the view clear, instead of presenting all information in one view, we use an info panel which shows additional information for the selected elements on mouse-over (Figure 2).

Marking a Region of Interest: The users can choose a chromosome region of interest (via tab ‘Display Options’, visible in Figure 2), and manipulate the view only inside the region. This functionality, which marks the region on the chromosome with a red box, is offered by both single and comparative representations. The red box can be moved along the chromosome and its boundaries can be adjusted. The main view shows only the data for the marked region and the users manipulate the data in the selected area. This means that when the user zooms or pans in the main view, all or some of the data from the red square is available. Data outside the coordinates marked by the square is not shown. We found the functionality useful, especially for the users who work with a particular part of a chromosome.

Zooming and Panning: We offer smooth zooming which supports the visual exploration of the chromosome space, based on Piccolo [2]. This provides efficient repainting of the screen, bounds management, event handling and dispatch, picking, animation, layout, and other features. The zooming technique allows the users to keep an area of interest in focus during interaction with the data. Zooming is manipulated by the right mouse button by moving it to the right (zoom in) or to the left (zoom out). Panning uses the left mouse button. Both interactions are easy to use and the users quickly become familiar with them, as confirmed by our study [10].

Focus On: Focus on (Figure 1) makes the focal element large enough, so that its name can be read, moves it to the center of the view, and marks its boundaries in red, which allows the user to see a small part of a viewing history until he changes the region of interest. This means that the user can see which elements he focused on during the session. In a single representation, when the user focuses on an element, all neighbouring elements in the view become proportionally larger in all columns. In a comparative representation, only elements in the chromosome containing the chosen element are changed, and all elements on the other chromosome maintain the same size. This allows the users to see an overview of elements from one chromosome and details for the selected element in the second chromosome. If the user wants all elements in the two columns to be of the same size, he chooses focus elements in both. Then we set the size of all elements to be the same.

Labelling: Because of a large amount of data, there is a problem with labels, especially for elements that have the same location. To solve the problem we allow the users to switch between viewing all labels or only a selection. When all labels are visible, they are connected by blue links to the visible elements. When the user moves the mouse close to the element, a link becomes highlighted, which allows the user to localize the element description faster, see Figure 5 A. In selected label view, Figure 5 B, we display

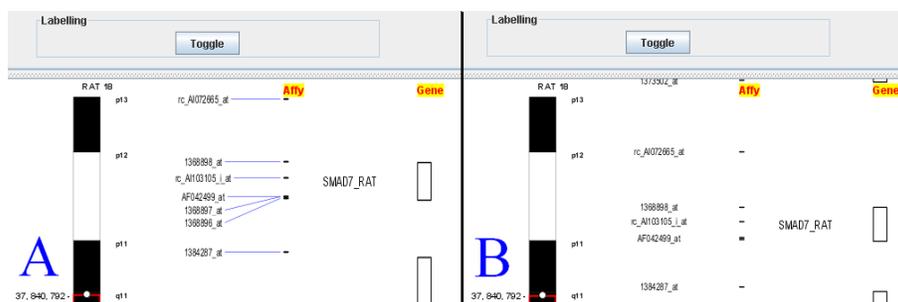


Fig. 5. The single representation for rat chromosome 18. **A** shows all labels and links which connect labels with the elements. **B** shows only a selection of labels shown next to the objects they describe.

only a small subset of labels. If there is enough room, the element name is displayed. For elements with the same coordinates, it is the first element in alphabetic order. We show as next the label for the next element which has enough room for the label.

Additional Information: Many genome browsers place all data into one view, which makes the data difficult to read. We display additional information in an info panel, see Figure 2. In a comparative representation, we show two types of information. We display Ensembl id, coordinates and a description for each element which is pointed to by a mouse. In a comparative representation when the user points to an element from one chromosome which has a homology with an element from the other chromosome, the additional information is displayed for both genes, see Figure 2. Display Options Tab allows the users some data manipulation, like choosing the range of the chromosome region displayed, changing between view with scaled data and unscaled data, or between views with all labels and selected labels. In our solution we do not have to display all information in the main view and this improves usability.

Colours: We use black and white for most data, however, after marking a region of the chromosome, the user can choose color for each of the elements by clicking on the object while pressing Alt. The default colour choice view is displayed and the user can change the colour of the marked element, see Figure 3.1. Additionally, the object boundaries are marked in red during *focus on* and all bands in the chromosomes are coloured by standard colours.

Supporting Data: Ensembl [6] offers data collected from publications and experiments. To help the user contextualise the data, we provide access to Ensembl by clicking on a feature of interest, which invokes Ensembl in the browser (Figure 1).

Homologies: To support comparative genome analysis, we show chromosomes which have homologies with other chromosomes. Our solution allows the users to identify all homologous chromosomes quickly. When a user looks at all chromosomes in a number of species, and clicks on one, all the homologous chromosomes in other species are highlighted, and facilitate the choice of homology for visual analysis (not shown).

4 Discussion

We examined the visualisation techniques used in genome browsers, and recognised that a number of tools used in biological research implement well known visualisation techniques, but only a few experiment with new techniques. CartoonPlus adds a novel extension to the array of available solutions. It can be used in single and comparative representations. In a single representation, the users can see all data scaled, depending on a chosen basis, which allows them to see clearly which micro array probes and markers are related to a gene. In a comparative representation, the scaling makes homologies between genes clearer.

Among all genome browsers we studied, only SyntenyVista [7] uses a scaling algorithm, however it was used only in a comparative representation and only for genes. The solution we used is novel and it could be useful not only in genomic data but also in different fields of biology and medicine which use one linear scale for many types of objects. We are testing the new technique in an experiment with biological researchers who now use a combination of data from Ensembl and their own lab experiments. We conducted a *user study*, to identify future improvements and assess the usability of our solution and

saw that biologists found it useful, especially for scaling small objects (SNPs) [10]. We will next offer the users interactive choice of the basis for the scaling. We want to improve colouring and give the users the option to add colour to a region and not only to a single element.

Conclusions

We designed and implemented a new scaling algorithm and combined it with some known visualisation techniques. Our new technique presents the data more clearly, especially in a comparative representation where the users want to see homologies. We believe our visualisation extension improves on the existing tools which try to present as much data as possible or only a predefined subset of data. The combination of scaling, labelling and focus techniques we offer is likely to support an improved understanding of data relationships, as required in biomedical research. In the long term we see significant potential for user control over exactly how and where scaling is done, as in magic lenses [3], although we emphasise the need for a user study to validate this.

References

1. Bederson, B. B. et al.: Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. UIST'94, ACM, 17–26 (1994).
2. Bederson, B. B. et al.: Toolkit Design for Interactive Structured Graphics. *IEEE Trans. Soft. Eng.*, **30**(8):535–546 (2004).
3. Bier, E. A. et al.: Toolglass and magic lenses: the see-through interface. SIGGRAPH, 73–80 (1993).
4. Fischer, G. et al.: Expressionview: visualization of quantitative trait loci and gene-expression data in Ensembl. *Genome Biol* **4**:R477 (2003).
5. Furnas, G. W.: Generalized Fisheye Views. *CHI*, 16–23 (1986).
6. Hubbard, T. J. P. et al.: Ensembl 2007. *Nucleic Acids Res.* 35, Database issue: D610–D617 (2007).
7. Hunt, E. et al.: The Visual Language of Synteny. *OMICS*, **8**(4):289–305 (2004).
8. Jakubowska, J. et al.: Granularity of genomics data in genome visualisation. TR-2006-221 (2006). <http://www.dcs.gla.ac.uk/publications/PAPERS/8212/AsiaElaMatthewCHI06.pdf>.
9. Jakubowska, J. et al.: VisGenome: visualisation of single and comparative genome representations. *Bioinformatics* **23**(19):2641–2642 (2007).
10. Jakubowska, J. et al.: Mixed Paradigm User Study. *In preparation*.
11. Leader, D. P.: BugView: a browser for comparing genomes. *Bioinformatics* **20**:129–130 (2004).
12. Min, W. et al.: A fisheye viewer for microarray-based gene expression data. *BMC Bioinformatics* **7**:452 (2006).
13. Montgomery, S. B. et al.: Sockeye: A 3D Environment for Comparative Genomics. *Genome Research* **14**(5):956–62 (2004).
14. Pook, S., Vaysseix, G., and Barillot, E.: Zomit: biological data visualization and browsing. *Bioinformatics*, **14**(9):807–814 (1998).
15. Sinha, A. U. and Meller, J.: Cinteny: flexible analysis and visualization of synteny and genome rearrangement in multiple organisms. *BMC Bioinformatics* **8**:82 (2007).
16. Stone, M. C. et al.: The movable filter as a user interface tool. HCI'94 Human Factors in Computing Systems, ACM Press 306–312 (1994).
17. Wang, H. et al.: SynView: a GBrowse-compatible approach to visualizing comparative genome data. *Bioinformatics* **22**(18):2308–2309 (2006).

C: VisGenome: visualisation of single and comparative genome representations

Joanna Jakubowska^{a,*}, Ela Hunt^b, Matthew Chalmers^a, Martin McBride^c and Anna F. Dominiczak^c

^aDepartment of Computing Science, University of Glasgow, G12 8QQ, Scotland,

^bDepartment of Computer Science, ETH Zurich, 8092 Zurich, Switzerland,

^cBHF Glasgow Cardiovascular Research Centre, University of Glasgow, G12 8TA, Scotland

ABSTRACT

Summary: VisGenome visualises single and comparative representations for the rat, the mouse, and the human chromosomes at different levels of detail. The tool offers smooth zooming and panning which is more flexible than seen in other browsers. It presents information available in Ensembl for single chromosomes, as well as homologies (orthologue predictions including ortholog one2one, apparent ortholog one2one, ortholog many2many) for any two chromosomes from different species. The application can query supporting data from Ensembl by invoking a link in a browser.

Availability: <http://www.dcs.gla.ac.uk/~asia/VisGenome>

Contact: asia@dcs.gla.ac.uk

1 INTRODUCTION

VisGenome was designed to match the visualisation needs of the BHF Blood Pressure Group at the University of Glasgow which uses a rat model of cardiovascular disease. We analyse genomic data at different levels of detail, to dissect rat and human QTLs (McBride et al., 2006) in the search for candidate disease genes. QTLs are analysed in three species: the rat, the human and the mouse, and the work uses genotyping, and micro array and proteomics techniques. VisGenome supports QTL analysis by showing QTLs and the genes within each QTL in two species in one display, along with the supporting experimental data. It also shows data for a single chromosome in one display and supports zooming at an arbitrary level of detail. Current software release connects to Ensembl (Hubbard et al., 2007) and can be used as an alternative viewer for a subset of that data. Extensions are being tested and an ongoing user study will guide further development.

2 FEATURES

VisGenome loads QTLs, genes, micro array probes, bands, and markers, and displays pairs of homologies (orthologue predictions) from Ensembl. It welcomes the user with a view of all rat, mouse and human chromosomes, see Fig. 1. In the single chromosome representation, after choosing a chromosome of interest by clicking, a new view with detailed data about the chromosome is created. In the comparative representation the user clicks on two chromosomes from different species and a new view representing homologies between the chosen chromosomes is created. After choosing a chromosome, the users can manipulate the view by mouse and keyboard interaction.

Navigation. VisGenome offers mouse and keyboard interaction. The users choose the representation by selecting a menu item or by typing Ctrl+S (single) or Ctrl+C (comparative). A choice is offered for marking the chromosome region of interest. The users can drag or enlarge the box enclosing the region marked on the chromosome or enter the region coordinates in the top panel.

Zooming and Panning. We offer smooth zooming which supports the visual exploration of the chromosome space. Zooming and panning uses Piccolo (Bederson et al., 2004). The zooming technique allows the users to keep an area of interest in focus during interaction with the data. Zooming is manipulated by the right mouse button by moving it to the right (zoom in) or to the left (zoom out). Panning uses the left mouse button. Both interactions are easy to use and the users quickly become familiar with them.

Marking a region of interest. The users can choose a chromosome region of interest. Then, the main view shows only the data for this region and the users manipulate only the data in the selected area.

Additional Information. VisGenome supports the display of additional information about presented data in an info panel.

Supporting data. Access to Ensembl is provided via clicking on a feature of interest which invokes Ensembl web pages in the user's browser.

3 IMPLEMENTATION

VisGenome was written and tested on Windows XP with Java 1.5, on a 2.39 GHz Pentium 4 with 512 MB RAM. VisGenome connects via JDBC to Ensembl databases for the rat, the mouse, and the human. During the tests we visualised chromosome bands, micro array probe sets, QTLs and genes. Eclipse, www.eclipse.org, was used as a development environment. The application is packaged as a jar file. The user has to install Java prior to invoking the code. 1280 x 1024 screen resolution was used for testing.

4 DISCUSSION AND CONCLUSIONS

Visualisation of genome comparisons is an important research tool in biology and medicine. None of the tools we studied (Jakubowska et al., 2006) fulfilled the requirement of showing in detail and overview all the data needed for our work in one display, and allowed us to study QTLs and other relevant data. VisGenome extends SyntenVista (Hunt et al., 2004) with a single genome display and presents homologies alongside QTLs and micro array probes.

*to whom correspondence should be addressed

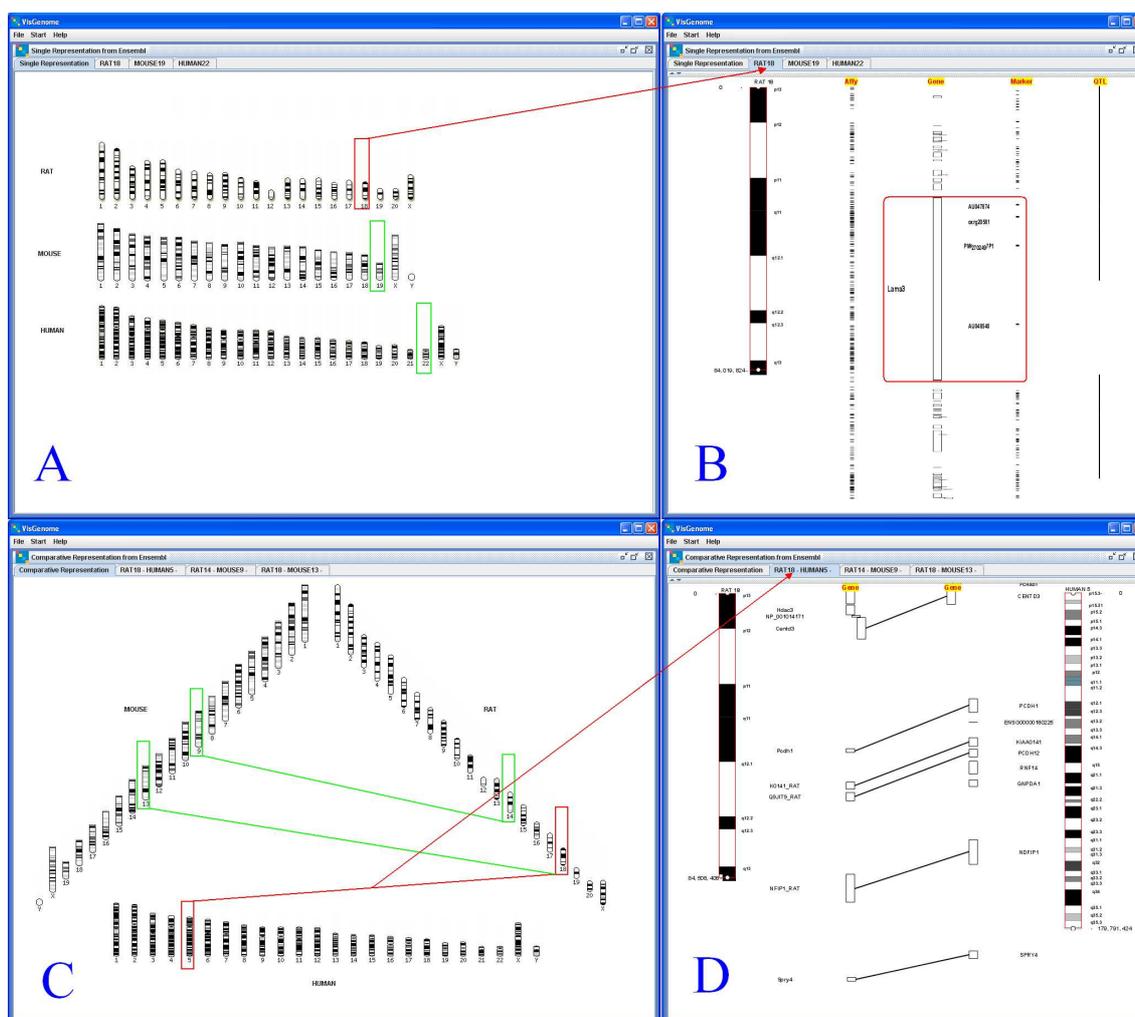


Fig. 1. A and B - VisGenome, single chromosome view. C and D - comparative view. A and C: chromosomes from the mouse, the rat and the human. B: an overview and detail for the rat chromosome 18. D: an overview and detail for the rat chromosome 18 and the human chromosome 5.

Our plans are as follows. We are conducting a user study comparing VisGenome and Ensembl, to identify future improvements and assess the usability of our solution. We will broaden the application of cartoon scaling which is not part of the current release, and requires algorithmic extensions. We will support the import of additional data from external files or web services, and local caching. We will add data import from DAS servers (Dowell et al., 2001). We are preparing a follow-on release supporting user-driven choice of colours and data sources. The data will be cached, so that the tool is usable even if access to Ensembl is slow, and the user will not have to re-import the same data. An intelligent caching solution is being tested.

ACKNOWLEDGEMENT

This work was funded by the BHF Programme (PG 02012) and Chair (CH98001) Grants and the Wellcome Trust

Cardiovascular Functional Genomics (WT066778/Z/01/Z) all to A.F.D., the MRC (J.J.), and by the MRC and an EU Marie-Curie Fellowship (E.H.).

REFERENCES

- Hunt, E., Hanlon, N., Leader, D., Bryce, H., and Dominiczak, A. F. (2004) The Visual Language of Synteny. *OMICS*, **8**(4):289–305.
- Jakubowska, J., Hunt, E. and Chalmers, M. J. (2006) Granularity of genomics data in genome visualisation. *Dept of Comp. Sci., University of Glasgow*, Tech. Report: TR-2006-221, <http://www.dcs.gla.ac.uk/publications/PAPERS/8212/AsiaElaMatthewCHI06.pdf>.
- Hubbard, T. J. P. et al. (2007) Ensembl 2007. *Nucleic Acids Res.* **35**, Database issue: D610-D617.
- Bederson, B. B., Grosjean, J., and Meyer, J. (2004) Toolkit Design for Interactive Structured Graphics. *IEEE Transactions on Software Engineering*, **30** (8):535–546.
- McBride, M. W., Graham, D., Delles, C., Dominiczak, A. F. (2006) Functional genomics in hypertension *Curr Opin Nephrol Hypertens* 2006 Mar;**15**(2):145-51.
- Dowell, R. D. et al. (2001) The Distributed Annotation System. *BMC Bioinformatics*, **2**:7.

D: Genome Visualisation

Joanna Jakubowska
University of Glasgow
Department of Computing Science
Glasgow G12 8QQ, Scotland
asia@dcs.gla.ac.uk

Abstract

In the visualisation of complex data there are a lot of unsolved problems. The complexity of data is increasing fast, but the users' ability to understand seems to be constant. The biological researchers we collaborate with would like to see the data in some user-friendly tools, but, the tools, computer monitors, and machines have their limitations and do not always precisely show the data under investigation.

We experiment with visualisation tools and develop new techniques which will more precisely express the complexity of data. We examined existing tools used by researchers and created a classification which will help to find a solution. We developed a prototype called VisGenome. VisGenome is an application for visualising single and comparative representations of the rat, the mouse, and the human chromosomes. During my PhD, I am going to conduct a user study and experiments with both existing and new applications for genome visualisation.

Keywords: genomics visualisation, genome browser.

1. INTRODUCTION

Large and complex data give rise to visualisation problems. Researchers try to solve the problems either from database or visualisation point of view. There is only a small group of people examining current work practices and studying Human-Computer Interaction with biological data. We began our work by studying existing visualisation solutions in order to find out what features they offer, which of those correctly support data analysis, and which are not helpful. The study will allow us to find a better solution for data analysis which overcomes cognitive problems.

In the area of genome visualisation, we found two groups of problems: visualisation and management of large amounts of data which should be shown at the same time. We would like to find a solution which will clearly present the information, including all relevant data the biologists wish to see. We are aiming to derive general principles of data representation and visualisation usability for genomics. We also would like to discover how best to compare data coming from various sources and experiments in biological setting.

The PhD work focuses on the use of visualisation to support the understanding of very large data sets. We would like to create an universal solution. We hope, VisGenome will solve both the visualisation problems and some of the data integration problems. We would like to offer a clear presentation of the data the biologists wish to see. We cooperate and make experiments with biologists where we study Human-Computer Interaction. We hope that our study will allow for deeper understanding of the problems and help in finding a solution.

2. PROBLEMS IDENTIFIED DURING RESEARCH

There are two main groups of problems, one related to visualisation and the other one to database integration.

We found out that there is no universal tool for genome visualisation. Each biologist group uses a different tool in their experiments. Some tools show all publicly available data, see Ensembl [3]. Other tools visualise only specific data from one experiment and provide no possibility to add any external data. There is no possibility to add data from a new experiment and compare the results. The biological data has a variety of formats and is situated in a lot of places. Genome browsers can read the data from special file formats or from a database. Frequently, the databases use different technologies and the users need to convert the data before it can be visualised.

The main visualisation problem is how to show all data and what kind of visualisation technique to use. The tools offer poorly designed zooming or panning. Some applications, for example DerBrowser [2] offers zooming, but it is not smooth zooming and is limited in depth. New genome browsers are often designed without respect for the principles of HCI. Therefore we decided to experiment with VisGenome. At the time, the application shows data in natural scaling, and we will extend it to use cartoon scaling [1]. We are developing an algorithm which represents different kinds of data in cartoon scaling.

3. PROPOSED SOLUTION / METHODOLOGY

The methodology will be based on prototyping with the users, combined with investigating existing genome browsers. In cooperation with the biologist group we tested some more popular genome browsers in order to find which one supports the interpretation of their experiments. We found out that none of the browsers fully supports the user requirements we identified. The experiments motivate us to define a genome browser classification and develop a new visualisation tool - VisGenome. We are going to conduct a study with the users – biologists from schizophrenia and hypertension groups at the University of Glasgow, London and Edinburgh. The experiments will combine studying data from the biologists' experiments with measuring the time, and counting the mouse clicks. A survey will be used to get users impression on the legibility of the display, aesthetic appeal, and the subjective ease of use.

4. CONTRIBUTIONS

The experimental work will be the main contribution. To our knowledge, nobody has carried out so far experiments with genome browsers. The deeper understanding of how the biologists work, what kind of information they need in their experiments, and the two-fold character of my research (the experiments are parallel with the prototyping of VisGenome and looking for an universal solution for representation the relevant biological data) is the next contribution.

5. CURRENT WORK

Initially, we studied the existing source code of two browsers SyntenyVista [1] and DerBrowser [2]. The aim of the work was to modify the existing implementation to implement Fisheye [5] and Excentric Labelling [6]. Then, we surveyed genomics visualisation software and defined a classification of genome browsers according to three dimensions: number of species, size of the objects shown, and representation complexity. The classification argues the need for a new genome browser which offers improved zooming functions. Therefore we develop an extension of SyntenyVista [1], VisGenome. The application allows for the addition of new data types to the display, and will be able to fully satisfy user requirements. So far, we developed the part responsible for single data representation. The tool offers smooth zooming and panning implemented using Piccolo [4]. The users can keep an area of interest in focus during zooming process. The solution allows the biologists to keep the context which help them not to get lost.

6. FUTURE DIRECTIONS AND CONCLUSION

Future work will use a number of prototypes which will be assessed with users. We will test not only VisGenome but also other tools with users.

Visualisation of genome comparisons is an important research tool in biology and medicine. There are variety of genome browsers which in practice should perform the same function – show the chromosomes of some species in detail. The differences in the view and also in functionality of genome browsing motivated me to create a classification of genome browsers and, in consequence, to develop new tool – VisGenome. We believe, that biologists still require new methods to visualise genomic data.

REFERENCES.

- [1] Hunt, E. et al. (2004) The visual language of synteny. *OMICS*, **8(4)**, 289–305.
- [2] Leser, U. et. al. (1997) IXDB, an X chromosome integrated database. *NAR*, **26(1)**, 108-111.
- [3] Ensembl, <http://www.ensembl.org>.
- [4] Piccolo Toolkit, <http://www.cs.umd.edu/hcil/piccolo/>.
- [5] Furnas, G. W. (1986) Generalized Fisheye Views. *Human Factors in Computing Systems CHI '86 Conference Proceedings*, 16–23.
- [6] Fekete, J. D. and Plaisant, C. (1999) Excentric Labeling: Dynamic Neighbourhood Labeling for Data Visualization. *CHI'99 Pittsburgh PA, USA*, 512–519.

E: Usability of VisGenome and Ensembl - A User Study

DCS Tech Report Number: TR-2007-244

corrected version, originally published on 14.03.2007

Joanna Jakubowska ^a, John McClure ^b, Ela Hunt ^c, and Matthew Chalmers ^a

^aDepartment of Computing Science, University of Glasgow, UK

^bBHF Glasgow Cardiovascular Research Centre, University of Glasgow, UK

^cDepartment of Computer Science, ETH Zurich, Switzerland

asia@dcs.gla.ac.uk, jdmc4w@clinmed.gla.ac.uk, hunt@inf.ethz.ch, matthew@dcs.gla.ac.uk

June 16, 2008

Abstract

We focus on the usability of genome browsers. In response to new requirements, we developed a browser which shows genetic data in overview and detail. An experiment involving biomedical researchers compared the new browser, VisGenome, and Ensembl. The study consisted of three search tasks and a questionnaire. It showed that neither of the visualisations is perfect. However, the subjects were more successful in VisGenome, and they liked the mouse manipulation offering improved navigation through biological data.

Chapter 1

Introduction

With the growth of biological data volumes it is becoming difficult to find the correct biological information and put it in the right context. One of the possible solutions to data contextualisation is visualisation. There are a number of genome browsers which show similar data differently [2], but, to our knowledge, their development was not accompanied by usability studies. Expression-view [7], for example, shows QTLs¹ and micro array probes and no other data. SyntenyVista [1] shows a comparative view of two genomes but is limited with regard to other data such as micro array probes. We reviewed the existing genome browsers and realised that they do not support fully the work of our collaborators from the British Heart Foundation Glasgow Cardiovascular Research Centre who study QTLs in the rat and the human, and want to compare them with known mouse QTLs [5]. Since we hope that new software will offer both usability and cognitive benefits, we study human-computer interaction to quantify those gains. Therefore, after a small survey [2], we developed a new prototype, and conducted an experiment which compared it to Ensembl [3]. VisGenome allows the researchers to see single and comparative views of chromosomes and uses smooth zooming and panning. It displays both large objects, such as chromosomes, and small areas such as Affymetrix probes, www.affymetrix.com, see Figure 1.1.

1.1 User study methodologies

In bioinformatics only a small number of user studies have been published. Stevens et al. [15] surveyed bioinformatics tasks undertaken by biologists. They investigated the biological nature and syntactic structure of queries and the tasks, and used a questionnaire to assess the bioinformatics knowledge and tool usage in the community. The authors report on a number of new requirements which could stimulate the development of future bioinformatics applications. The paper does not involve a user study and focuses on task classification.

Wu et al. [16] report on an electronic table that uses fisheye distortion. The table shows gene expression data and is a subject of a pilot user study. Five researchers took part and a Questionnaire for User Interface Satisfaction (QUIS) [17] was used. User impressions were positive. Yang and colleagues [18] studied biologists interacting with MetNet3D. Both students and researchers used the software to analyse experimental data, however, a formal study has not

¹A quantitative trait locus (QTL) is a part of a chromosome which is correlated with a physical characteristic, such as height or disease. Micro array probes are used to test gene activity (expression).

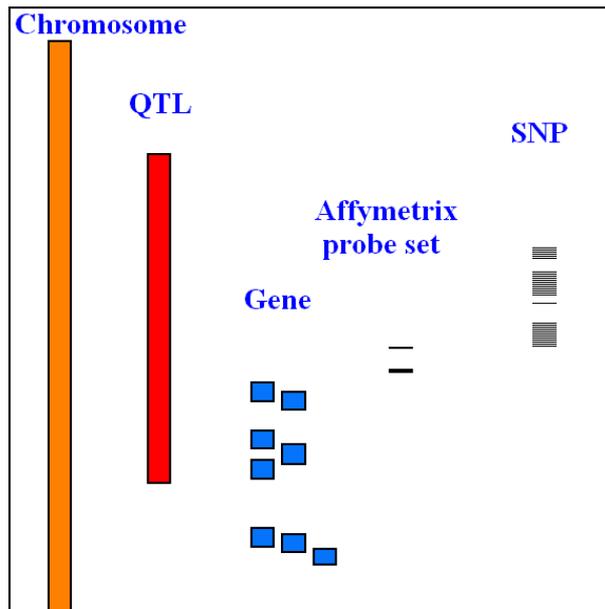


Figure 1.1: A genome browser represents chromosomes which are strings of letters: A, C, G and T, and contain between a few million and several hundred million letters (base pairs: bp). A QTL measures up to several million bp and an Affymetrix probe has 25 bp, while a single nucleotide polymorphism (SNP) is 1 bp long. Large differences in the size of the presented objects cause difficulties in the representation of data.

taken place.

The methodology of human-computer interaction encompasses a number of established techniques. For instance, Pausch et al. [8] compared head-mounted and stationary displays and as only important metric used task completion time. Dhamija and Perrig [9] compared *Déjà Vu* to traditional password and pin authentication and directly observed user interaction. We studied the functionality and representation of data in genome browsers [2] and then decided to conduct a user study comparing the usability of VisGenome and Ensembl, as no reports of the usability of such tools could be found.

1.2 VisGenome

VisGenome (VG), see Figure 1.3, shows single and comparative representations of the rat, the mouse and the human chromosomes at different levels of detail, and uses data from Ensembl. It offers an overview of all rat, mouse and human chromosomes. After choosing a chromosome of interest, the user sees it in a new view with detailed data. The view supports interaction by mouse and keyboard, such as smooth zooming and panning [4] which is more flexible than seen in other browsers. The users can keep an area of interest in focus and choose the chromosome region by dragging the box enclosing the region or typing in the coordinates in an info panel. Then only the data in the selected area is displayed. The aim is to provide the context and allow the researchers to navigate the data at the same time. VG retrieves supporting data from Ensembl

by invoking a link in a browser.

1.3 Ensembl

Ensembl (Ens) [3] is probably one of the most popular systems for genome analysis. It offers 17 different views, including ChromoView, ContigView, GeneView, MultiContigView, SNPView, and SyntenyView. In our experiment, biological and medical researchers used ContigView, MultiContigView (see Figure 1.2) and SyntenyView (see Figure 1.5). ContigView, see Figure 1.4, shows different views of a gene, from broad chromosome context to fine nucleotide detail. These views are in separate horizontal frames, one below the other. Data items are labelled and searching on names and coordinates is possible. Zooming uses buttons (panels C and D) and pop up menus (panels A, B and C). One needs to scroll to see the views, as only two views fit on a computer screen. MultiContigView is an extension of ContigView. It displays genome annotation for several species. In SyntenyView a clickable high-level view of chromosomes with blocks of conserved synteny is shown.

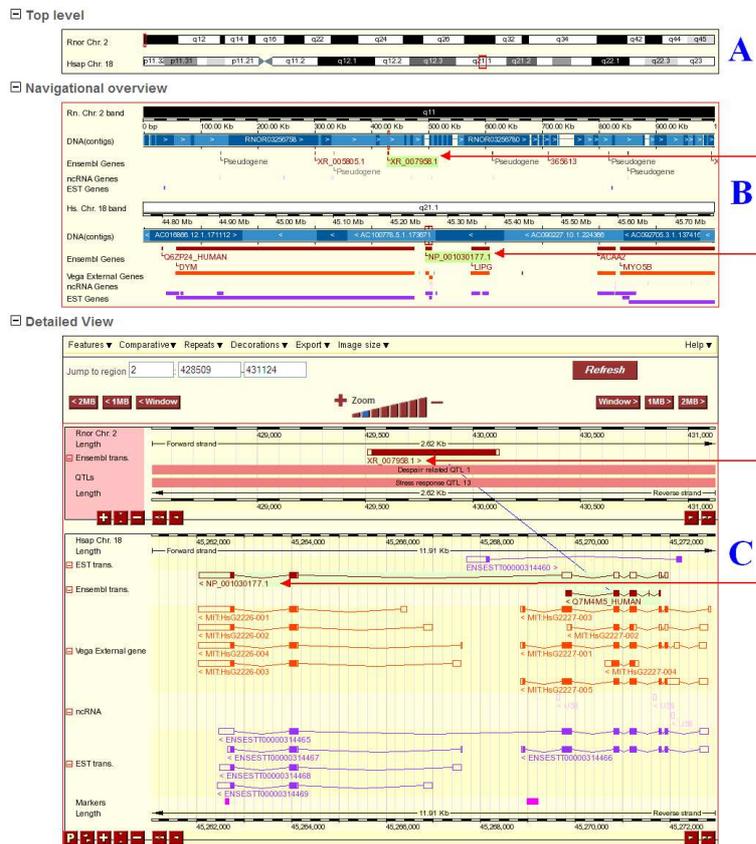


Figure 1.2: Ensembl MultiContigView for rat chr. 2 and human chr. 18. (A) shows the entire chromosomes, (B) is an ‘Overview’ of the homologous genes, (C) is the ‘Detailed View’ showing the homology between XR_007958.1 and NP_001030177.1, and the homology is the same as in VG in Figure 1.3.

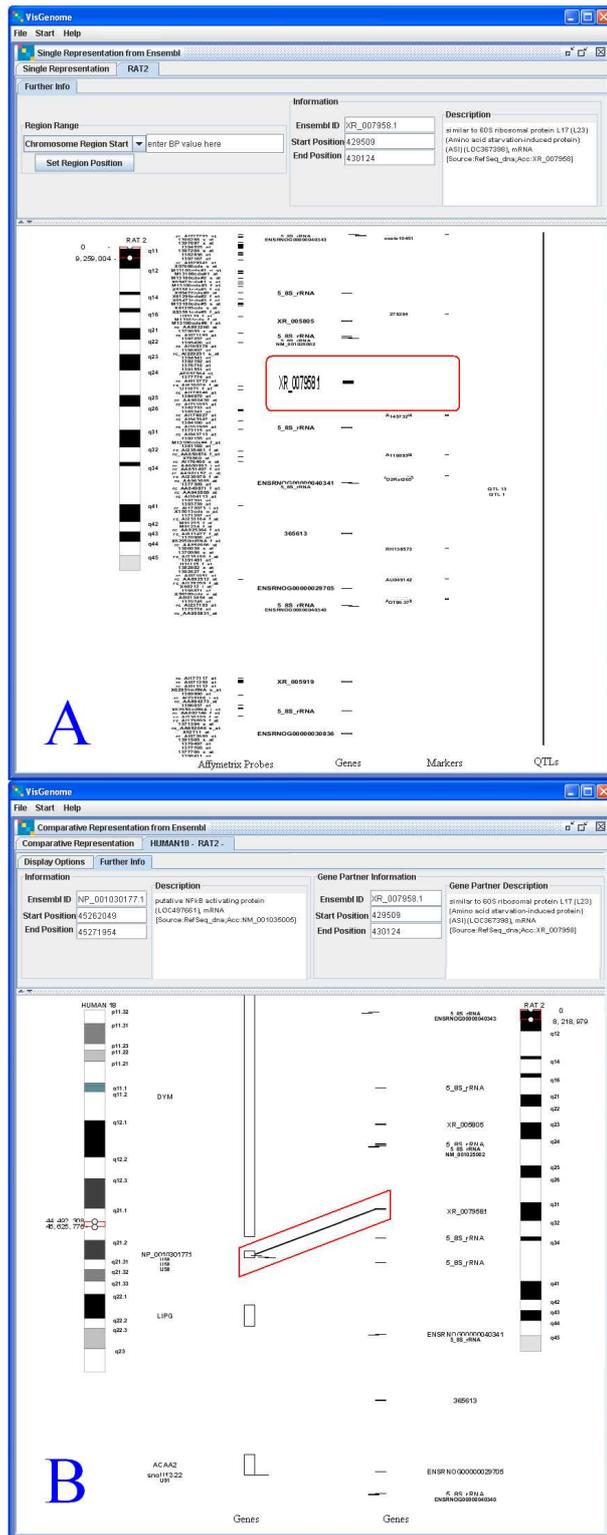


Figure 1.3: A: VisGenome, single chromosome view, an overview and detail of the rat chr. 2. B: VisGenome, comparative view, an overview and detail of the rat chr. 2 and the human chr. 18. Gene XR_007958.1 is framed in red in both views.

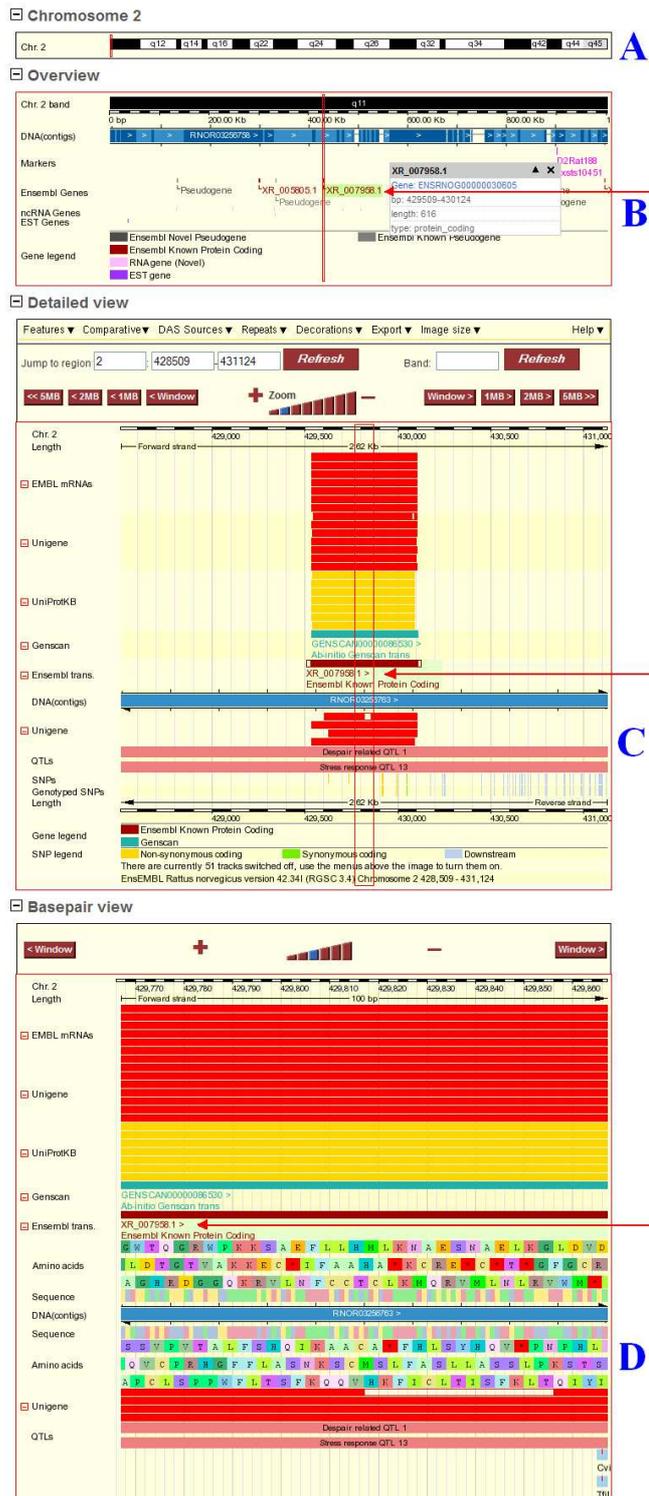
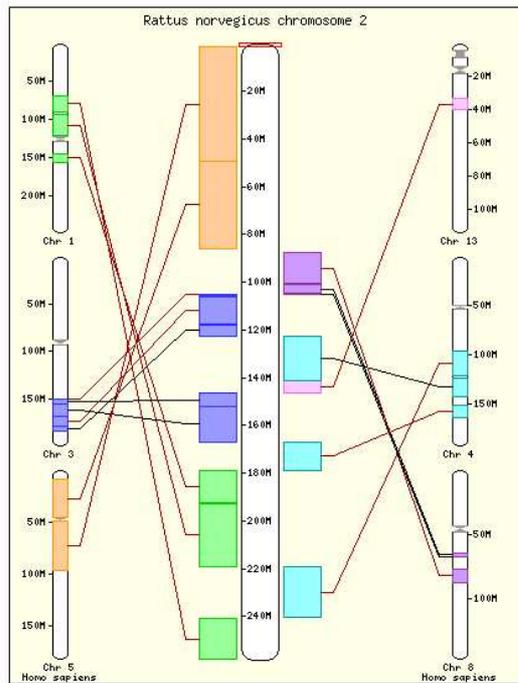


Figure 1.4: Ensembl ContigView for rat chr. 2. (A) shows the entire chromosome, (B) is an ‘Overview’ of a region of 1 Mbp, (C) is the ‘Detailed View’ showing markers and genes, and (D) is a ‘Basepair View’ showing letters. XR_007958.1 gene shown in B and C here is also shown in VG in Figure 1.3. Arrows point to XR_007958.1 in views B, C and D.



Homology Matches	
Rattus norvegicus Genes	Homo sapiens Homologues
5_8S_rRNA (134.44 Kb) [ContigView]	No homologues
ENSRNOG0000040343 (135.89 Kb) [ContigView]	No homologues
5_8S_rRNA (246.67 Kb) [ContigView]	No homologues
XR_005805.1 (300.51 Kb) [ContigView]	-> ENSG00000187544 (5: 78.86 Mb) [ContigView] [MultiContigView]
	-> XR_017017.1 (14: 69.42 Mb) [ContigView] [MultiContigView]
	-> SNORD36C (9: 135.19 Mb) [ContigView] [MultiContigView]
	-> ENSG00000175886 (18: 35.17 Mb) [ContigView] [MultiContigView]
	-> ENSG00000182281 (14: 67.88 Mb) [ContigView] [MultiContigView]
5_8S_rRNA (330.25 Kb) [ContigView]	No homologues
5_8S_rRNA (332.75 Kb) [ContigView]	No homologues
NM_001025002.1 (334.19 Kb) [ContigView]	No homologues
XR_007958.1 (429.51 Kb) [ContigView]	-> NP_001030177.1 (18: 45.26 Mb) [ContigView] [MultiContigView]
	-> RL17_HUMAN (15: 69.42 Mb) [ContigView] [MultiContigView]
	-> ENSG00000187096 (3: 199.06 Mb) [ContigView] [MultiContigView]

Figure 1.5: Ensembl SyntenyView for rat chr. 2. We see the entire rat chromosome, and small icons of syntenic human chromosomes. XR_007958.1 and NP_001030177.1 gene are framed by a box. Note that A does not show synteny with the human chromosome 18 while B does.

Chapter 2

A User Study

We studied the usability of Ens and VG. Although the tools offer similar functions, Ens shows more data types than VG, as VG does not show sequence level data (view D in Figure 1.4) or gene structure (view C). In both cases, we used the same tasks and measured search time in minutes and the number of mouse clicks (NoMC).

2.1 Participants

We first carried out a pilot experiment with two subjects from the Bioinformatics Research Centre (BRC) and five from the Western Infirmary (WI) in Glasgow. Finally, in the experiment we had 15 participants from the WI and the BRC. Six of them use Ens often (Ens Experts: Ex). Nine of them use different tools, such as BugView [12], UCSC GenomeBrowser [13] or AtIDB [14], or were from BRC and do not use genome browsers but know them from presentations (NonExperts: NEx). Three of the participants previously took part in a one day Ens course.

2.2 Methods

None of the biologists have used VG before the experiment. We gave a short presentation of VG to all subjects. Several researchers asked us to remind them first how Ens works and where to find information (three participants - NEx). We gave them a short introduction to Ens. Before the experiment we offered the subjects the opportunity to carry out an experimental task in VG (for NEx also in Ens).

Prior to the study, two WI subjects had asked to see their experimental data. To that end, we created one version of VG for the majority of participants and two specific versions with private data. In those versions micro array probes were coloured in both Single and Comparative Representations. The aim was to receive more feedback from those subjects.

The experiment was divided into two parts (Ens and VG). Afterwards, the subjects filled in a questionnaire and participated in an interview. We explained to the participants what we understand by Single and Comparative Representation and that VG offers Single and Comparative Representations, but in Ens the subjects have to decide if they would like to use MultiContigView or SyntenyView as Comparative Representation, and ContigView or any other Views as Single Representation. Some of the participants asked us if they can use BioMart [11] or

RGD [10] (2 users) during the execution of Ens task. They could use all tools available from Ens pages. During the experiment the participants could give up if they thought that it was not possible to complete the task. The majority of the subjects attempted the tasks and only one person gave up and abandoned tasks T2 and T3, see below.

2.2.1 Search Tasks

We asked our biological collaborators to recommend some common search tasks. After some discussion and analysis we defined three tasks, as follows.

T1. Single Representation. Choose one of the rat, mouse or human chromosomes. Mark the whole chromosome and show all available data. Then choose the region between 100bp and 10,000,000bp and note the name of the first gene and the last Affymetrix probe inside the region.

T2. Comparative Representation. Choose rat chr. 18 and human chr. 5. Zoom in and out to find any homologies between genes. Then choose one of the homologies and read out the gene names of the homologous genes.

T3. Single Representation. Choose one of the rat chromosomes. Find the longest QTL. Then zoom on it and write down the names of the genes which are the closest to the beginning and the end of the QTL.

2.2.2 Questionnaire

We offered all subjects a questionnaire in order to study their perception of the mental and physical demands of the tasks. The scale used had 21 points, from -10 (low/poor) to +10 (high/good). All 15 subjects answered the following questions, once with regard to Ens and then with reference to VG.

Q1. How much mental and visual activity was required?

Q2. How psychically demanding did you find this experiment?

Q3. How much time pressure did you feel because of the rate at which things occurred or the time limit imposed on the task?

Q4. How hard did you work?

Q5. How successful do you think you were in doing the task set by the experimenter? How satisfied were you with your performance?

Q6. How much frustration did you experience?

Q7. How annoying did you find the mouse manipulations used in the experiment?

Q8. Which of the systems do you prefer?

2.2.3 Interview Questions

During an interview after the experiment we asked the participants to answer the following questions.

Q9. How often do you use a computer during your work?

Q10. How often do you use a genome browser during your daily work?

- Q11.** If you use a genome browser, please give the name of the one you use the most frequently.
- Q12.** What do you like/dislike about Ensembl?
- Q13.** How often do you use Ensembl in your daily work?
- Q14.** What do you like/dislike about VisGenome?
- Q15.** Do you think the fisheye visualisation technique is useful? (Do you like it?)
- Q16.** Do you think excentric labeling is useful? (Do you like it?)
- Q17.** Do you use panning? (Do you like it?)
- Q18.** Do you use zooming? (Do you like it?)
- Q19.** Is zooming via buttons, for example in Ensembl, better than via mouse action?
- Q20.** Do you use scroll bars, for example in Artemis od UCSC Browser?
- Q21.** Which other visualisation techniques in VisGenome and Ensembl seem to be helpful?
- Q22.** Are the colours in the visualisation meaningful for you?
- Q23.** If you use the colours at all in the visualisation, please say how you use them?
- Q24.** Which of the representations of chromosomes do you prefer (karyotypes in three rows, karyotypes in a triangle, coloured histogram), see Figure 3.13, page 43.
- Q25.** Is it important for you to have any additional information about the genes such as presented in VisGenome in Panel Info? What would you like to see on it?

2.3 Task Benchmark

We first carried out a test benchmarking the tasks. For VG we observed: for T1, 30 NoMC, 2min 22s; for T2, 16 NoMC, 1min; and for T3: 38 NoMC, 2min 40s. On our first attempt for T2 in Ens we obtained 6 NoMC, 1min, for T1 the browser crashed a few times, and then during T3 in Ens we abandoned the test, as Ens became unavailable, see figure 2.1.

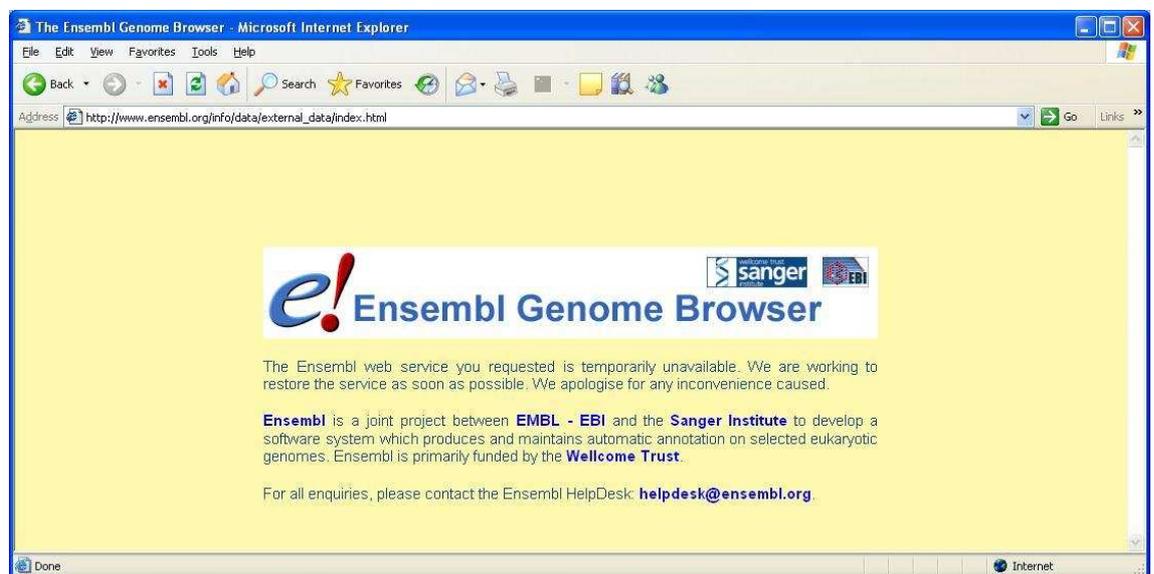


Figure 2.1: Ensemble not available.

On another day, we carried out T1 and T3 in Ens successfully. T1 took 3min 40s and 30

NoMC, and T3 took 15min 10s and 51 NoMC. We summarise this benchmark test in Table 2.1.

Table 2.1: The developer time and NoMC for T1, T2, and T3 in VG and Ens.

	VisGenome		Ensembl	
	Time	NoMC	Time	NoMC
T1	00:02:22	30	00:03:40	30
T2	00:01:00	16	00:01:00	6
T3	00:02:40	38	00:15:10	51

2.3.1 T1 in VG

We expected the users to choose the Single Representation from the menu and then drag one of the chromosomes into the lower panel, see Figure 2.2-1, where rat chromosome 18 was chosen. After clicking on the selected chromosome in the lower panel, the user sees Figure 2.2-2. She has to mark the whole chromosome by stretching the red box to the top and to the bottom of the karyotype image, see Figure 2.2-2. Subsequently, the user has to zoom out to see all genes, Affymetrix Probe Sets and QTLs, Figure 2.2-2. Then, in the top panel, the user chooses the region between 100 bp and 10,000,000 bp, see Figure 2.3-3. and notes the name of the first gene, see Figure 2.3-4, and the last Affymetrix probe inside the region, see Figure 2.3-6.

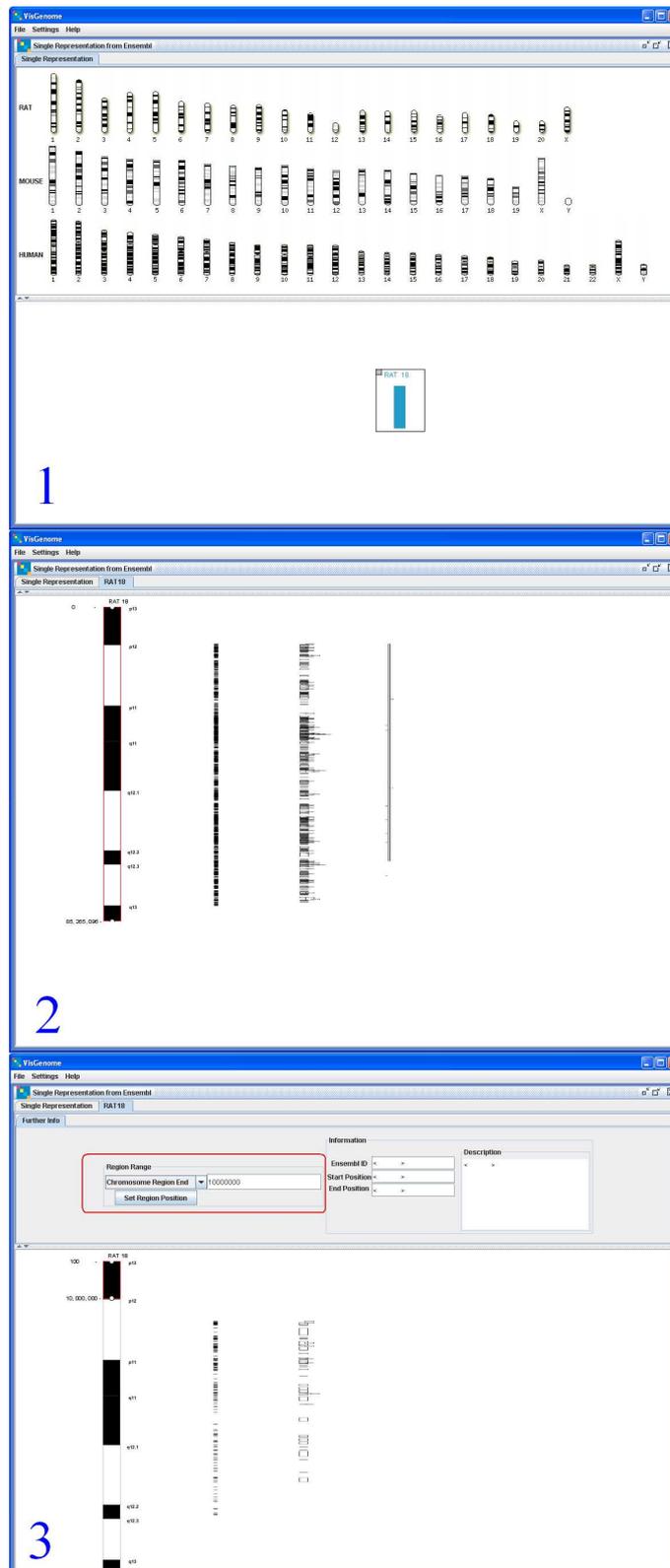


Figure 2.2: T1 in VG. 1 shows selecting rat chromosome 18 from the top panel. In 2 the user marks the whole chromosome and in 3 uses top panel to choose the region between 100 bp and 10 Mbp.

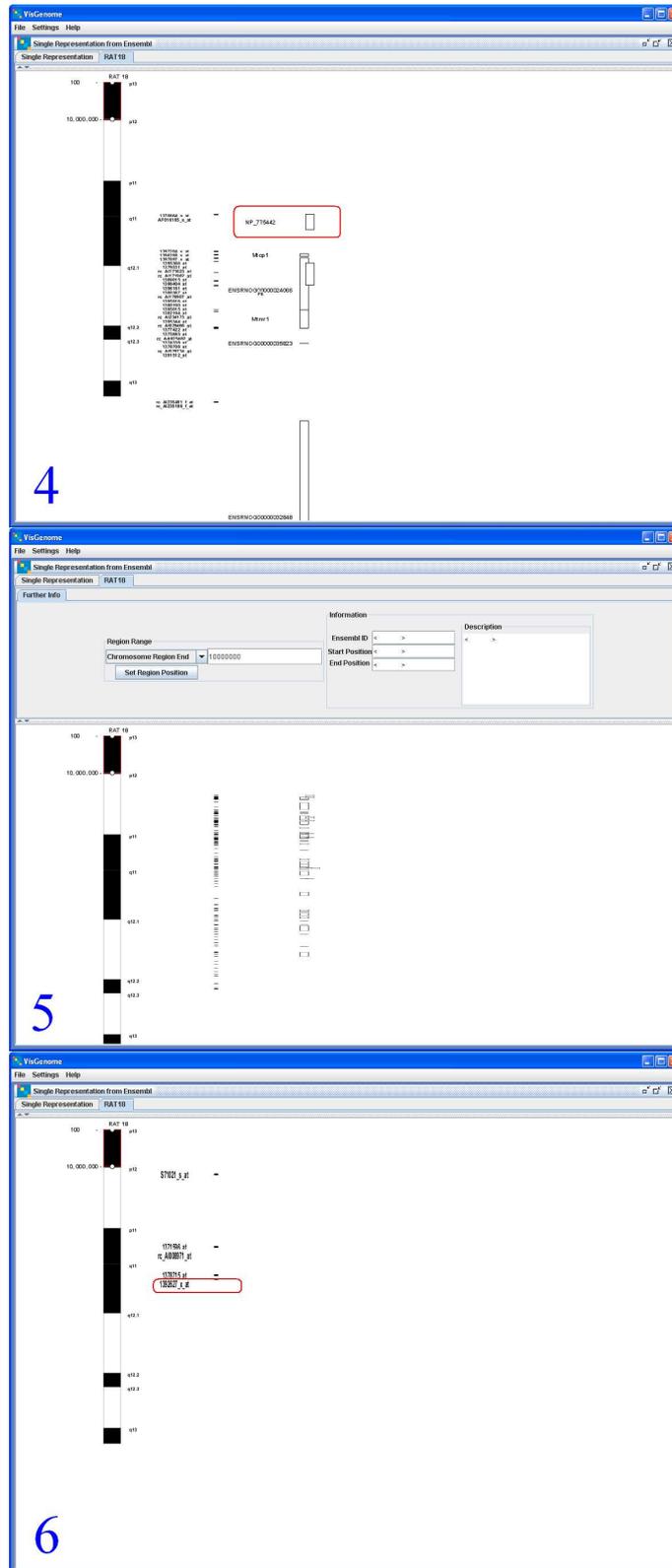


Figure 2.3: T1 in VG, continued. 4 shows the name of the first gene, NP_775442. In 5 the user zooms out to see the end of the marked region and in 6 the last Affymetrix probe, 1392627_x_at, is shown.

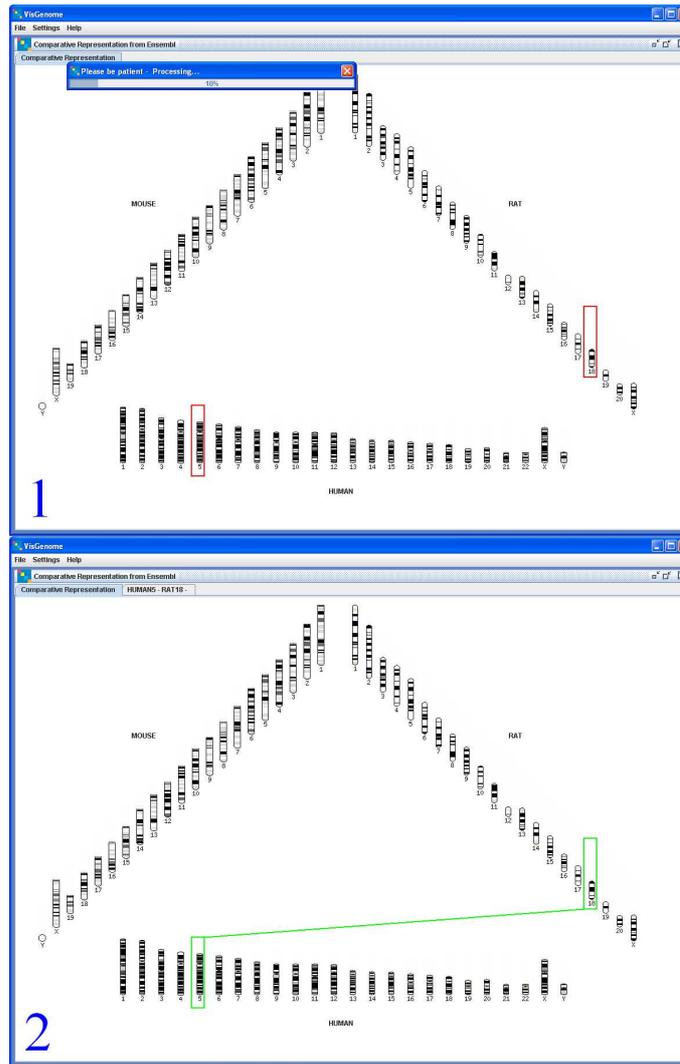


Figure 2.4: T2 in VG. 1, selecting rat chr. 18 and human chr. 5. 2, both chromosomes have been selected.

2.3.2 T2 in VG

In T2 the users choose the Comparative Representation and select the rat chr. 18 and the human chr. 5, see Figure 2.4-1,2. They zoom in and out to find any homologies between genes, see Figure 2.5-3, and then choose one of the homologies and show the gene names of the homologous genes, see Figure 2.5-4.

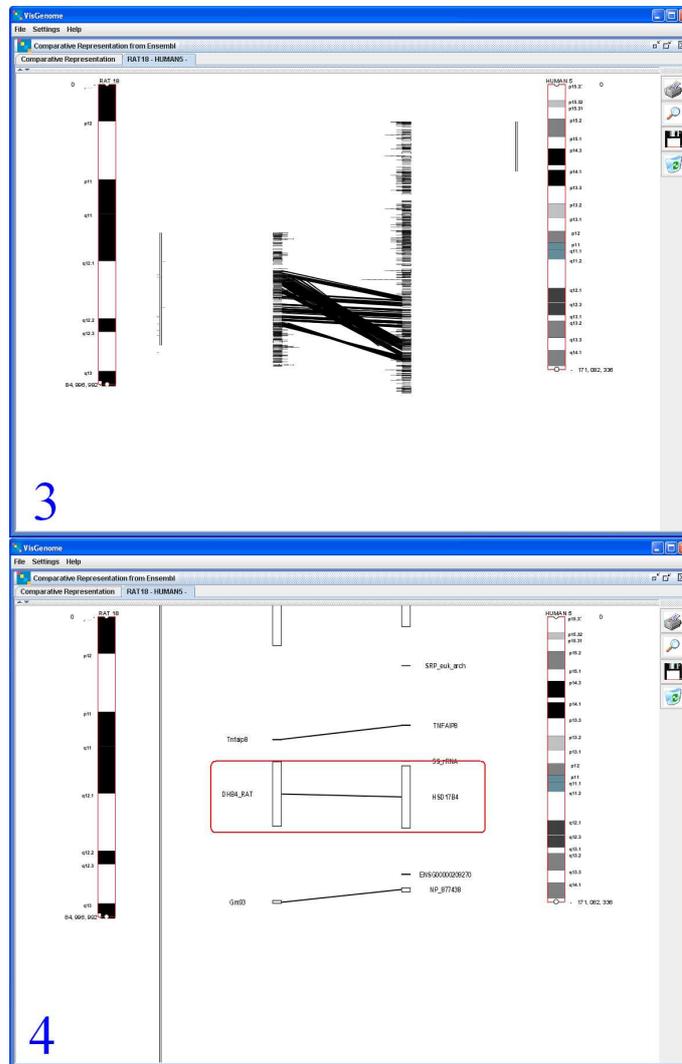


Figure 2.5: T2 in VG, continued. In 3 the user marks a large region on both chosen chromosomes (or even entire chromosomes) to find homologies. In 4 the user chooses one homology, here CHB4_RAT and HSD17B4, where a link symbolises a homology.



Figure 2.6: T3 in VG. 1, selecting rat chr. 18 from the top panel. 2, a user marks the whole chromosome to find the longest QTL. 3, the user zooms to see the beginning of the longest QTL and shows NP_775442 which is the closest gene to the beginning of the QTL.

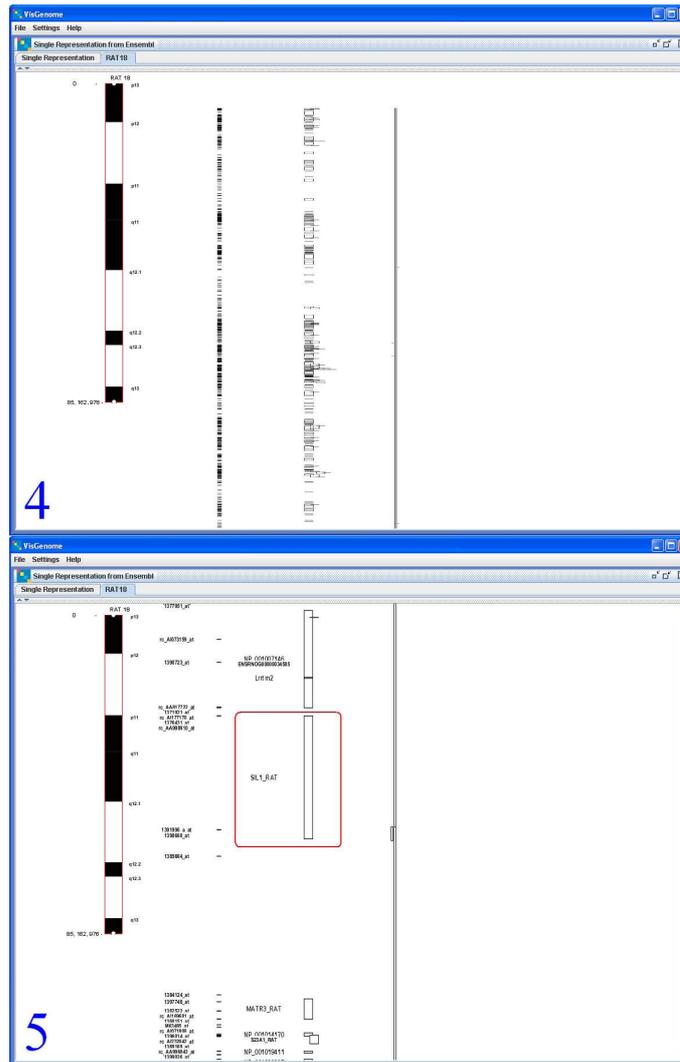


Figure 2.7: T3 in VG, continued. In 4 the user zooms out to see an overview and find the end of the longest QTL. 5 shows SIL1_RAT which is the gene closest to the end of the longest QTL.

2.3.3 T3 in VG

In T3 the participants select the Single Representation and choose one of the rat chromosomes. In Figure 2.6, rat chromosome 18 was chosen. To find the longest QTL, we marked the whole chromosome and zoomed out, see Figure 2.6-2. Then we zoomed in to the beginning of the longest QTL and showed the name of the gene which was the closest, see Figure 2.6-3. We zoomed out to see the end of the chromosome and finally we zoomed on the end of the QTL and showed the name of the gene which was the closest to the end of the longest QTL, see Figure 2.7-5.

2.3.4 T1 in Ensembl

All the tasks in Ens start from the main Ens web page, see Figures 2.8, 2.10, and 2.13-1. In T1 we chose mouse chr. 4, see Figure 2.8-3. Then we chose the region between 100bp and 10,000,000bp and show the name of the first gene, see Figure 2.9 - 4, and the last Affymetrix probe inside the region, which in Ensembl is marked as Oligo, Figure 2.9 - 5 and 6. We used in T1 MapView, CytoView and FeatureView offered by Ensembl.

2.3.5 T2 in Ensembl

In T2 we chose the rat chr. 18 and went to the SyntenyView for human chromosomes, see Figure 2.10-3 and Figure 2.11-4. On the right part of the image in Figure 2.11-4 we see two data columns. The first column shows all rat genes and the second column their human homologues. In the experiment it was enough to find one homology, but some users became interested in the data and added extra steps to the test. Figures 2.11-5 and 6, and 2.12-7 and 8 show additional data about the homologous genes.

2.3.6 T3 in Ensembl

In T3 the participants choose one of the rat chromosomes, see Figure 2.13 showing the choice of the rat chromosome 18. Then, to find the longest QTL we mark the whole chromosome by entering the region between 1 and 59,218,465bp, see Figure 2.13-3. We can easily find the longest QTL in CytoView, see Figure 2.14-4. Because Ens does not show detailed data for large regions, we marked a small region close to the beginning of the longest QTL, *Urinary albumin excretion QTL 19*, and showed the name of the gene which was the closest to the beginning, see Figure 2.15-6. Then we go back one step and zoom into the end of the longest QTL and show the name of the gene closest to the end of the QTL, see Figure 2.15-7.

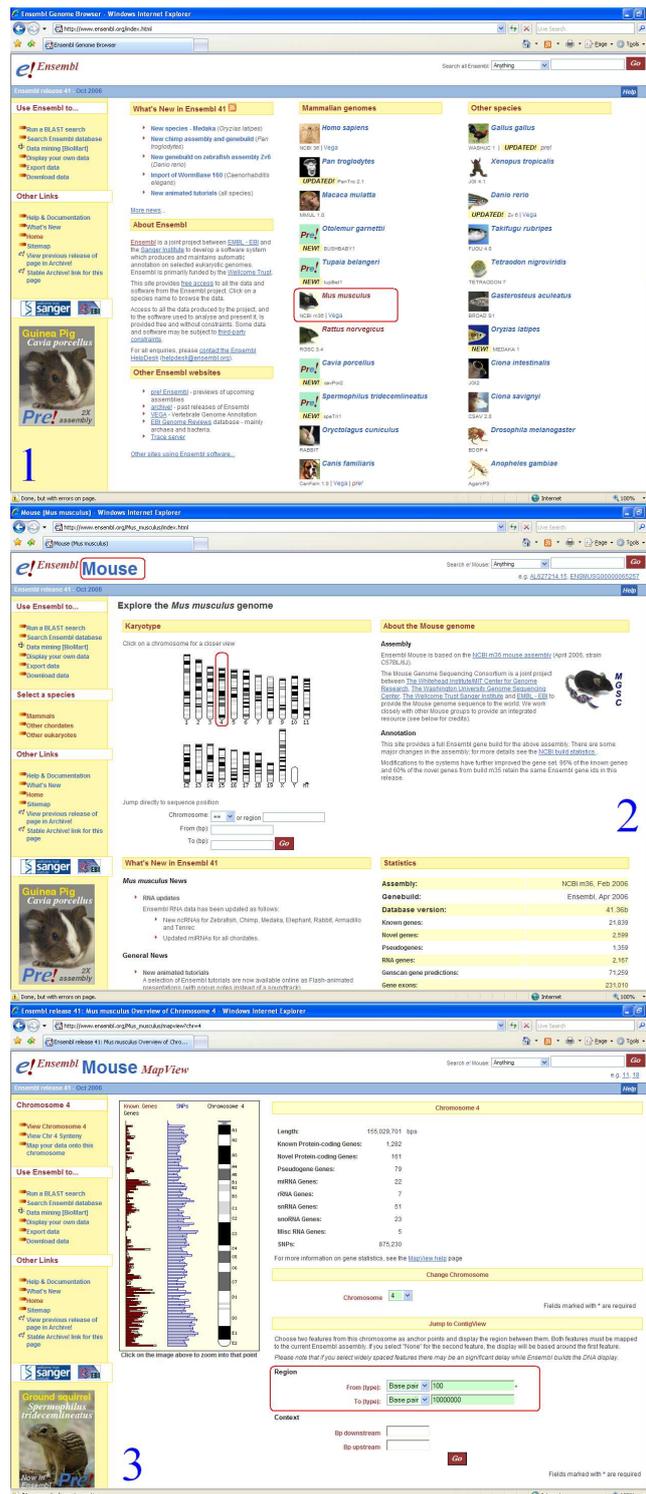


Figure 2.8: T1 in Ens. 1 shows the species display. In 2 the user chooses the mouse, and in 3 enters the coordinates for mouse chromosome 4.

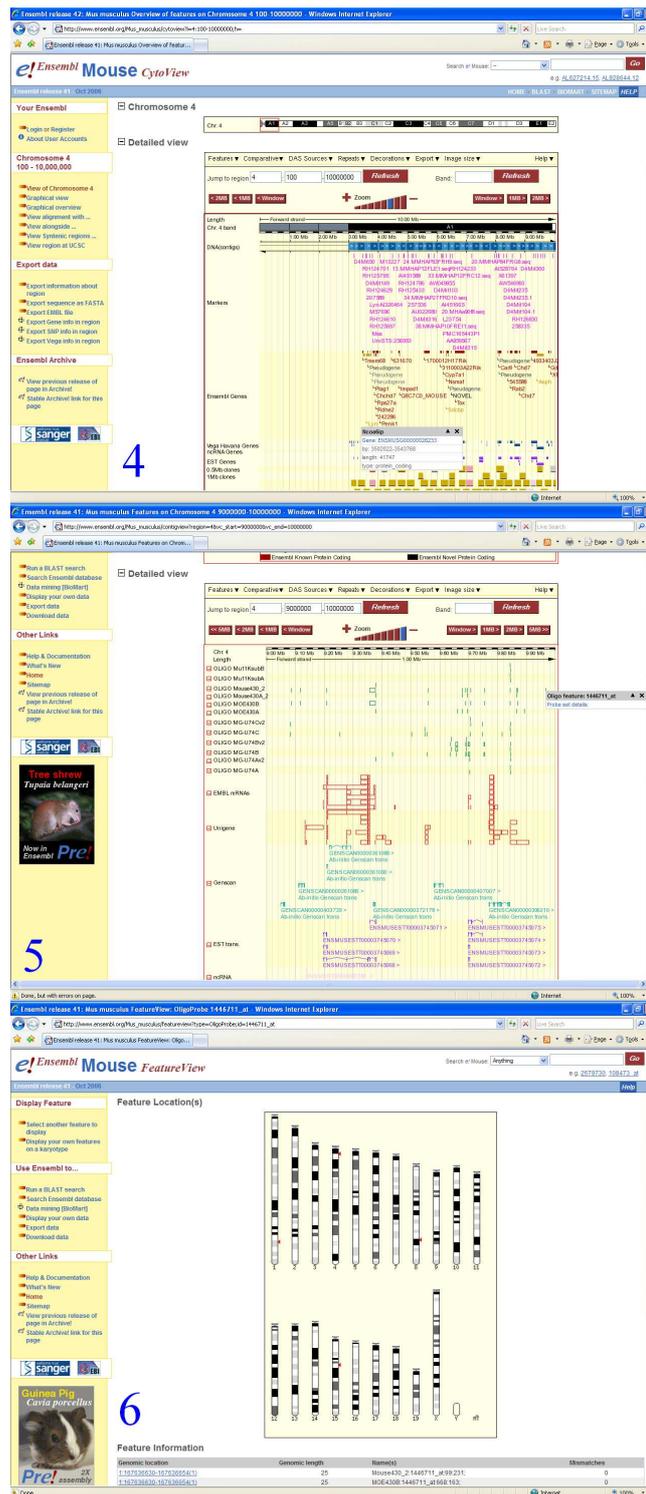


Figure 2.9: T1 in Ens, continued. In 4 the user finds *Nco6ip*, the first gene in the marked region. 5 shows 1446711_at, the last Affymetrix probe in the region. This was the end of T1 in Ens. However, some users continued and executed step 6 to show the FeatureView with the Affymetrix probe set.

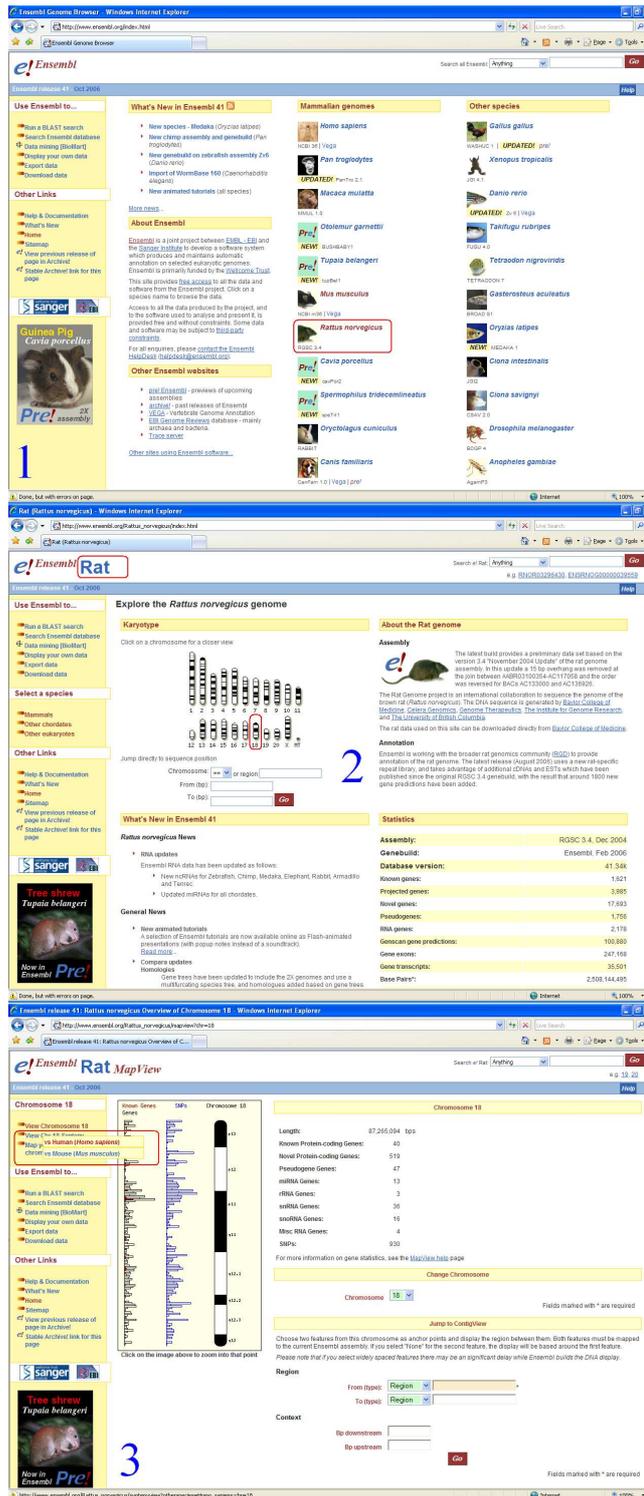


Figure 2.10: T2 in Ens. 1 shows the species. In 2 the user chooses rat chromosomes and in step 3 the rat chromosome 18, and chooses SyntenyView for the human genome.

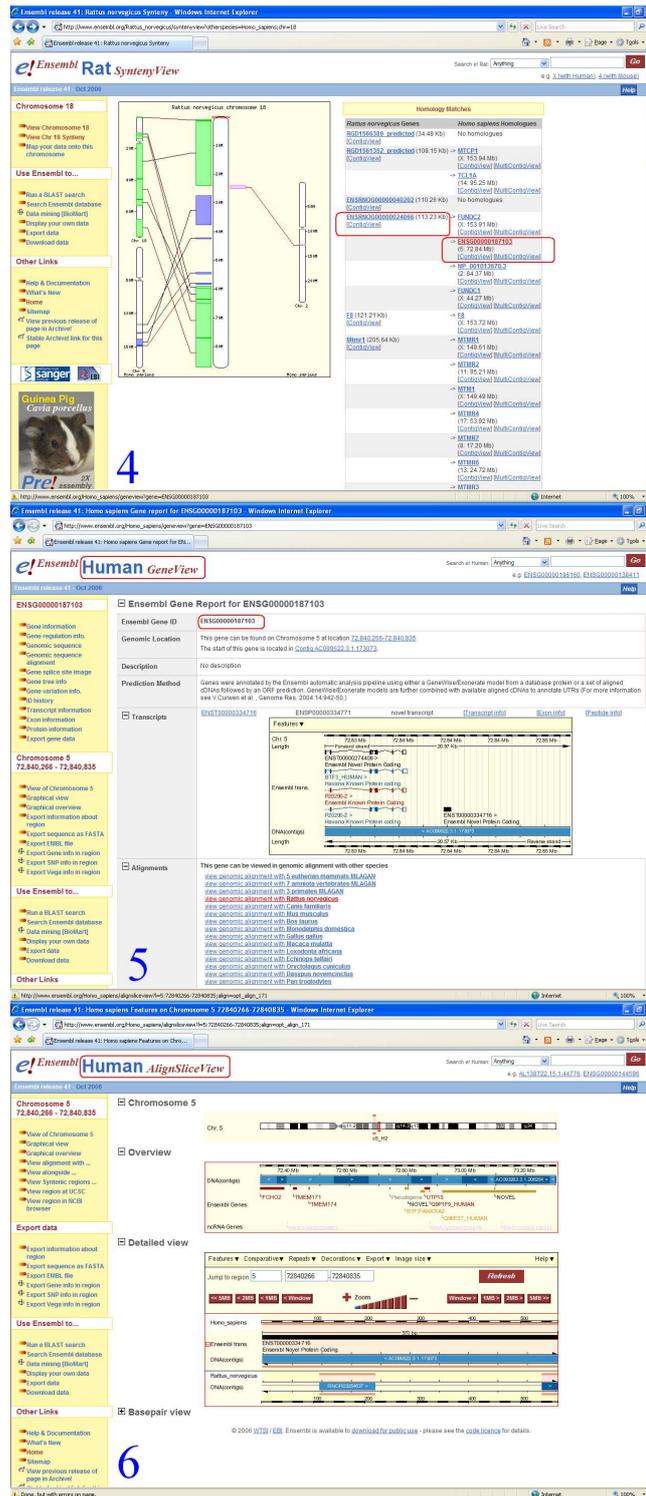


Figure 2.11: T2 in Ens, continued. In 4 the user finds ENSG00000187103, human gene (on chr. 5) in homology with ENSRNOG00000024066 (rat). This was enough to complete T2 in Ens. However, some users added step 5, showing GeneView for the selected human gene and step 6, AlignSliceView, showing even more information about the gene.

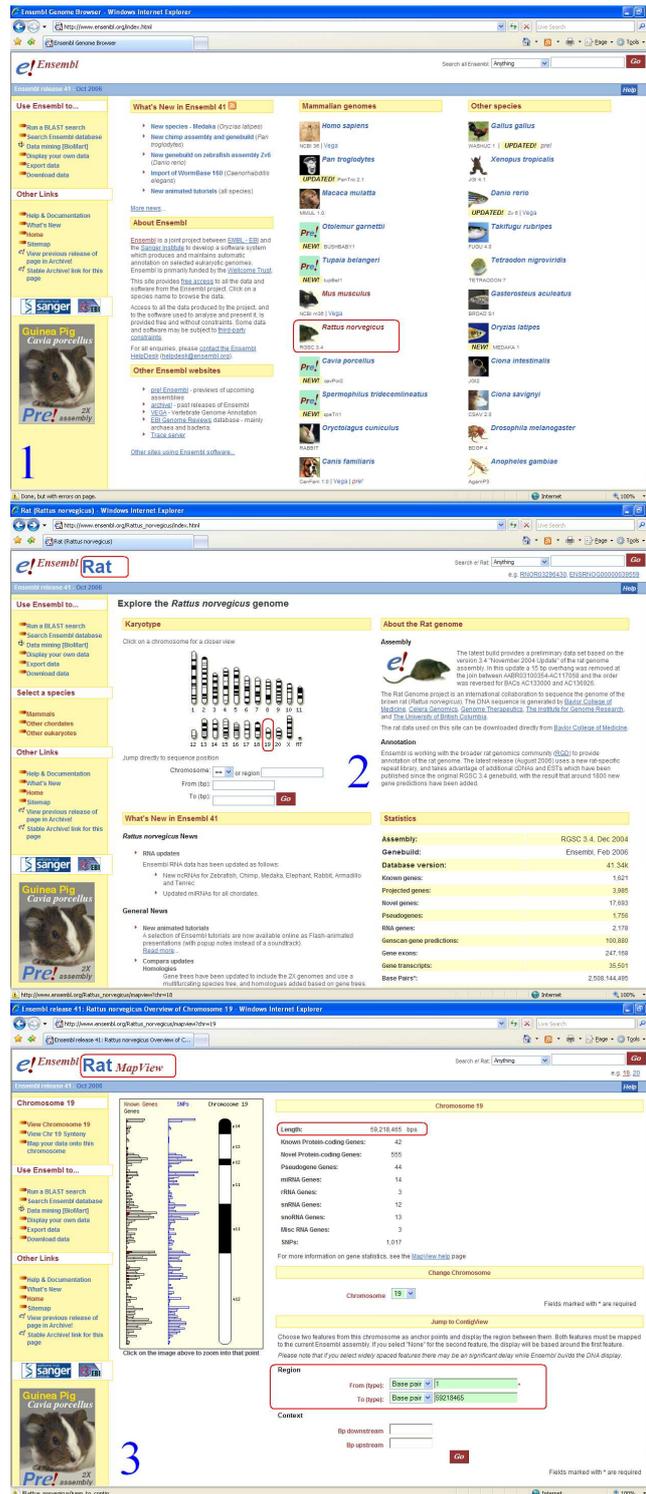


Figure 2.13: T3 in Ens. 1 shows all available species in Ens. In 2 the user chooses the rat and chr. 19, and in 3 enters the coordinates for the rat chr. 19.

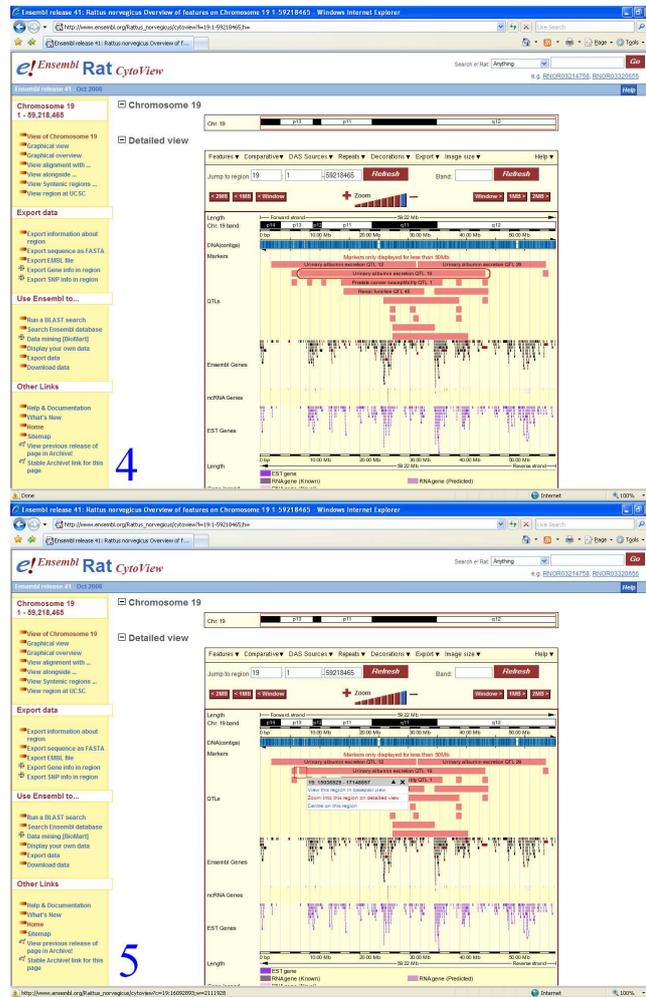


Figure 2.14: T3 in Ens. In 4 the user finds the longest QTL, *Urinary albumin excretion QTL 19* in CytosView. In 5 she zooms into the start of the QTL.

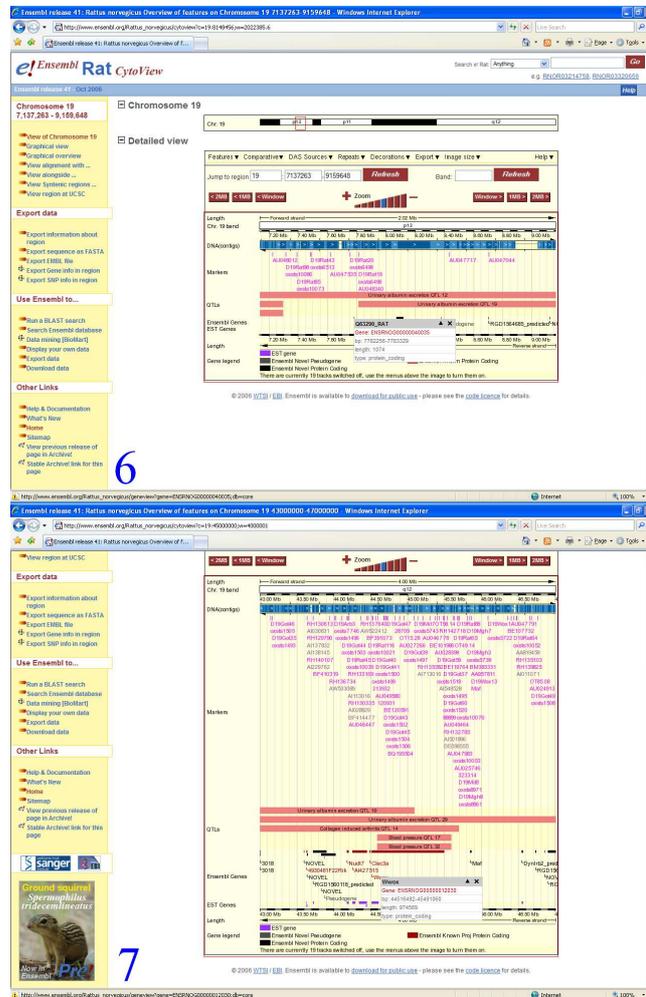


Figure 2.15: T3 in Ens. In 6 the user finds Q63290_RAT, the gene closest to the beginning of the QTL. In 7 the user zooms into the end of the region and shows Wwox, the gene closest to the end of the QTL region.

Chapter 3

Results

3.1 Raw Experimental Data

Table 3.1: Time, NoMC and success (SC) for T1 in VG and Ens.

	VisGenome			Ensembl		
	Time	NoMC	SC	Time	NoMC	SC
1dr1	00:07:28	59	0	00:05:34	54	0
2phd1	00:03:55	36	0	00:04:30	54	0
3phd2	00:08:50	89	0	00:16:18	167	0
4phd3	00:06:07	77	0	00:00:00	0	0
5phd4	00:04:09	25	0	00:03:43	60	0
6dr2	00:04:29	52	1	00:14:39	148	0
7dr3	00:08:37	147	0	00:06:04	71	0
8phd5	00:08:04	87	0	00:04:54	70	0
9phd6	00:04:37	63	0	00:04:49	58	0
10phd7	00:05:23	44	1	00:02:10	16	0
11dr4	00:04:19	43	0	00:05:50	48	0
12phd8	00:04:02	62	0	00:08:30	33	0
13dr5	00:06:28	141	0	00:03:30	40	0
14phd9	00:05:05	44	0	00:05:27	52	0
15dr6	00:07:13	63	1	00:32:00	45	0

We first present the raw data recorded for the 15 participants. Subject were named with a number and letter combination, where `dr` stands for post-doctoral researchers and `phd` for PhD students. We had 6 postdocs and 9 PhD students. For T1 to T3 in VG and Ens we show time (Time = HH:MM:SS), number of mouse clicks (NoMC) and 1 if the task was successful, and 0 if it was not (SC), see Tables 3.1, 3.2, and 3.3.

Table 3.2: Time, NoMC and success (SC) for T2 in VG and Ens.

	VisGenome			Ensembl		
	Time	NoMC	SC	Time	NoMC	SC
1dr1	00:04:28	52	1	00:05:57	50	0
2phd1	00:02:30	34	1	00:05:11	48	1
3phd2	00:03:15	45	1	00:09:52	62	0
4phd3	00:04:40	42	1	00:03:07	21	0
5phd4	00:02:59	38	1	00:02:44	40	0
6dr2	00:02:10	35	1	00:06:16	60	1
7dr3	00:05:02	62	1	00:02:19	19	0
8phd5	00:02:59	33	1	00:02:00	22	1
9phd6	00:02:16	42	1	00:01:01	8	1
10phd7	00:04:00	58	1	00:01:43	9	1
11dr4	00:05:04	69	1	00:02:40	24	1
12phd8	00:03:05	47	1	00:01:28	11	1
13dr5	00:05:45	138	1	00:02:17	19	1
14phd9	00:02:11	27	1	00:02:52	6	1
15dr6	00:03:19	44	1	00:16:00	56	0

Table 3.3: Time, NoMC and success (SC) for T3 in VG and Ens.

	VisGenome			Ensembl		
	Time	NoMC	SC	Time	NoMC	SC
1dr1	00:06:26	54	1	00:01:21	6	0
2phd1	00:04:04	60	1	00:11:56	157	0
3phd2	00:09:43	136	0	00:04:32	53	0
4phd3	00:06:02	97	1	00:00:00	0	0
5phd4	00:08:40	106	0	00:01:26	15	0
6dr2	00:04:50	74	1	00:01:58	30	0
7dr3	00:05:24	56	0	00:05:26	17	0
8phd5	00:04:40	78	0	00:09:00	131	0
9phd6	00:03:51	99	0	00:05:29	64	0
10phd7	00:04:02	37	0	00:02:50	16	0
11dr4	00:05:07	83	1	00:26:48	239	0
12phd8	00:04:01	76	1	00:08:30	40	0
13dr5	00:06:14	101	0	00:02:27	24	0
14phd9	00:06:27	80	1	00:03:05	18	0
15dr6	00:05:25	71	1	00:10:00	50	0

Table 3.4 shows the results of queries Q1 to Q7.

Table 3.4: The data for subjective questionnaire in VG and Ens. Queries are numbered 1-7 and correspond to Q1 to Q7.

	VisGenome							Ensembl						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
1dr1	0.0	0.0	-2.0	0.0	6.0	2.0	-4.0	0.0	-6.0	2.0	-2.0	0.0	2.0	0.0
2phd1	-4.5	-8.5	-9.0	-8.5	2.5	0.0	-7.0	-1.0	-6.0	-8.0	1.0	-3.0	2.0	-1.0
3phd2	2.0	-8.0	-8.0	-2.0	-2.0	-4.0	4.0	2.0	-8.0	-8.0	-2.0	-4.0	-4.0	-4.0
4phd3	-5.0	-10.0	-4.0	-1.0	7.0	-8.0	-3.0	4.0	-10.0	0.0	4.0	-8.0	-1.0	4.0
5phd4	2.0	-10.0	-9.0	-6.0	-2.0	4.0	4.0	-4.0	-9.0	-9.0	-4.0	-3.0	1.0	1.0
6dr2	-8.0	-8.0	-8.0	-8.0	8.0	-8.0	-8.0	8.0	0.0	0.0	8.0	0.0	2.0	2.0
7dr3	-3.0	-10.0	2.0	-6.0	-2.0	-8.0	6.0	4.0	-10.0	2.0	-2.0	-7.0	-6.0	8.0
8phd5	0.0	-6.0	-2.0	2.0	2.0	2.0	2.0	6.0	2.0	10.0	10.0	-8.0	8.0	6.0
9phd6	2.0	-9.0	-6.0	0.0	4.0	-2.0	-4.0	7.0	-9.0	-6.0	4.0	0.0	5.0	0.0
10phd7	-6.0	-8.0	-8.0	-6.0	4.0	-8.0	-8.0	0.0	-8.0	-8.0	-2.0	0.0	4.0	0.0
11dr4	-6.0	-9.0	-9.0	-4.0	4.0	2.0	-1.0	4.0	-8.0	-8.0	4.0	-4.0	6.0	8.0
12phd8	-7.5	-5.5	-8.5	-8.5	8.5	-9.5	-9.5	-4.5	-6.5	-8.5	-8.5	0.5	-7.5	-9.5
13dr5	-8.5	-1.5	-7.5	-1.5	4.5	-1.5	-7.5	-1.5	1.5	-2.5	-5.5	5.5	-8.5	1.5
14phd9	-4.0	-6.0	-8.0	-4.0	6.0	-6.0	-2.0	-4.0	-6.0	-6.0	-6.0	4.0	-6.0	-2.0
15dr6	-2.0	-6.0	-1.0	-4.0	0.0	-6.0	-4.0	4.0	-1.0	3.0	4.0	-6.0	-7.0	-8.0

3.2 Analysis

3.2.1 Overview

The results are quite surprising. The researchers who use Ens frequently are often unsuccessful in task execution. The experts encounter no problems in their everyday work which focuses on a chromosome fragment. However, when they examine similar data in a different part of the chromosome, they encounter problems. We also found that some of the zooming mechanisms in VG were hard to use and that the subjects prefer mouse clicking to dragging. The researchers want to see large amounts of data, but when they are looking for a particular object, they prefer to see only a small part of the data under investigation.

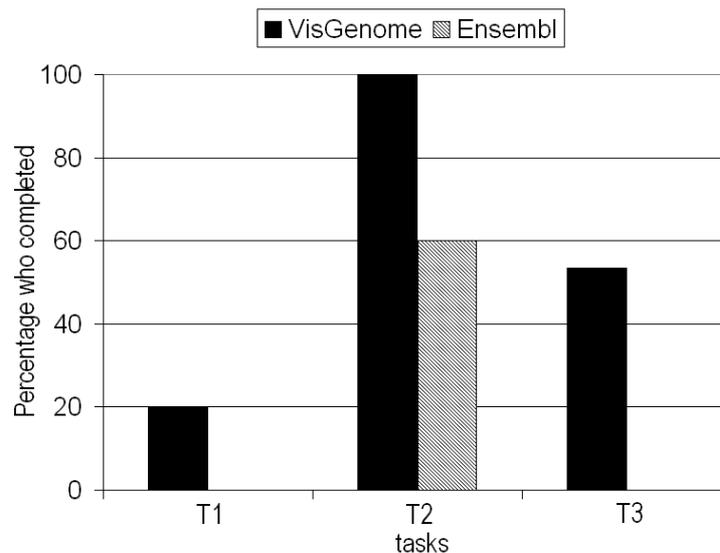


Figure 3.1: Percentage of subjects (out of 15) who completed each task.

3.2.2 Accuracy and Task Completion

Figure 3.1 shows that T2, the only task involving comparative genome representation, was more successful with VG (100%) than with Ens (60%, 9 subjects, see also Figure 3.2¹ for T2). In T3 53% of attempts were successful in VG (8 subjects), while in Ens the success rate was 0. In T1 we note 20% success rate in VG and 0% in Ens. Using the two-sided sign test (where 0=both/neither successful; 1=VG success but Ens not; -1=Ens success but VG not) as an alternative to McNemar's test the success rate for VG was significantly greater for both T2 ($P=0.0313$) and T3 ($P=0.0078$), but not for T1 ($P=0.25$). Completion rates were higher in VG than in Ens for all tasks, particularly for T2 and T3. This may be due to the fact that Ens is a much richer interface, with many more options and controls and represents more data. Possibly, the subjects

¹Survival analysis is a branch of statistics which deals with death in biological organisms and failure in mechanical systems. The presented plot is an example of a Kaplan-Meier plot for two conditions: task finished with success or without.

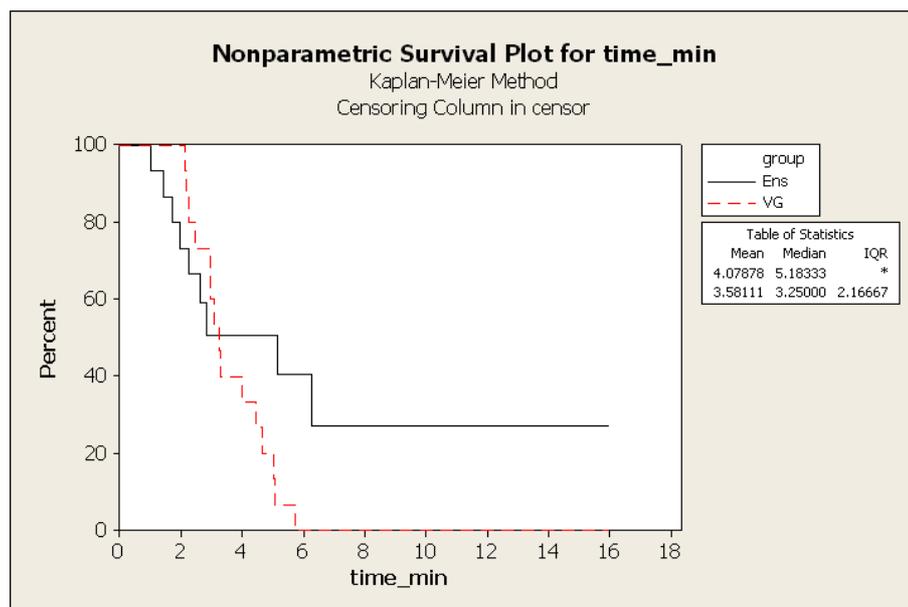


Figure 3.2: Survival plot for time in T2. 60% of the subjects finished VG T2 in 4 min or less. See that Mean and Median on the graph are presented for all subjects not only for successful like it is in the paper. We calculated the curves as follows: VG curve has 15 steps, because all the users (15) succeeded in T2. In Ens we see only 9 steps. It is because some users “disappear”, i.e. they finished but did not succeed. The “steps” are calculated as follows: at the beginning we have 15 users and they have 100% survival probability not to finish the task T2 (in medicine 100% survival probability not to die), see top left. Then when one user “died” (succeeded) we calculate probability not to finish the task for the remaining 14 users (in the new situation) as follows: $(1-(1/15)) = (14/15) = 0.93333$ (see 1st step). Then we have only 14 users and again one of them “died” (succeeded) so we calculate probability for the rest (13 users) as follows $(1-(1/14))*0.93333 = 0.866667$ survival probability not to die. Finally, we see 0.270899 survival probability not to die in Ens (for the 2 users who in that situation are still alive - anyway they finished unsuccessfully) and 0.0 in VG.

were not able to find out how to generate comparative genome views, or were getting lost while learning to use the system.

3.2.3 Time to finish

Time was measured in minutes. The biologists who completed the tasks had mean of T1=5.69’ (StDev=1.39’), T2=3.58’ (StDev=1.17’), and T3=5.29’ (StDev=0.97’) in VG and mean T2=2.83’ (StDev=1.76’) in Ens. As no one completed T1 and T3 in Ens, statistics were calculated only for T2, see Figure 3.3. In T2 in Ens and VG 9 researchers correctly completed both tasks. Sign Test for the differences in times is not significantly different for T2 (P=0.5078). We realised that Ex used both tools differently than NEx. Ex usually wanted to see more information, got interested in the data, while NEx subjects just wanted to complete the task. Ex tried to find and

show all possible answers they knew, and explore while doing the task. If there were several ways of doing the tasks in Ens they wanted to show all the solutions. In T2, for example, it was enough to show two genes in VG and Ens, and most NEx did that and finished quickly. Most Ex performed T2 and then explored MultiContigView to see more information about homologous genes, which took more time. Users behaved similarly in T3, however nobody succeeded in Ens. NEx showed Affymetrix probes in ContigView, while Ex used FeatureView and looked at the detail. There were also slight differences in server response times for Ens which might have influenced the speed of data analysis. Overall, in T2 there was little difference in task execution time between Ens and VG.

3.2.4 Mouse Clicks

Those who completed the tasks had the means of T1=53 (StDev=9.54), T2=51.07 (StDev=26.65), and T3=74.38 (StDev=13.38) NoMC in VG, and the mean for T2=23 (StDev=18.93) NoMC in Ens. Only T2 mouse clicks were analysed, due to non-completion in Ens for T1 and T3, see Figure 3.4. 9 subjects completed T2 with both VG and Ens, and despite the mean number of clicks being larger in VG than in Ens, there was no significant difference in NoMC, possibly due to the small sample size. One Ex had a very large NoMC (138) for VG, and only 19 for Ens. This shows that mouse manipulation in VG needs getting used to, as panning and zooming require keeping the left/right mouse button down and moving the mouse at the same time left/right or up/down, and the left/right movement is not offered by many similar applications where clicking on zoom bars is used instead, and smooth zooming is not widely used. This is a potential problem, however, most subjects learned how to use the mouse quickly. On the other hand, Ex often clicked to see additional information and some of NEx clicked because they wanted to find the solution and they were not sure where they had to look for it. This contributed to a large NoMC in some Ex as well as NEx.

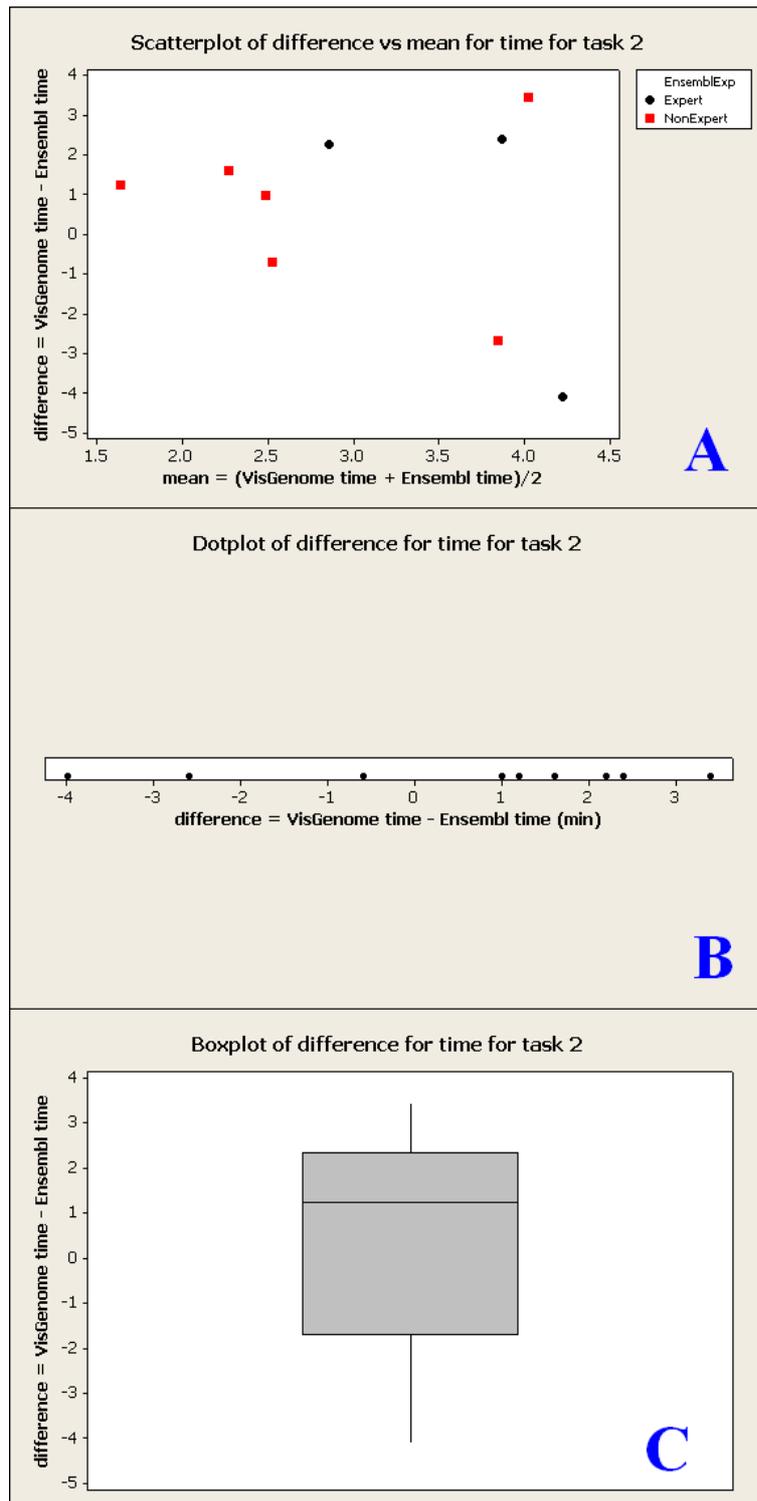


Figure 3.3: The Figure shows scatterplot (A), dotplot (B) and boxplot (C) for time differences for the 9 successful users in T2. Graph A clearly presents that there were 6 NEx and 3 Ex. The graphs show that 4 NEx had similar time in VG and Ens, there were 1 Ex and 1 NEx who executed T2 in Ens much faster than in VG, and 1 Ex and 1 NEx who completed T2 in VG much more quicker than in Ens.

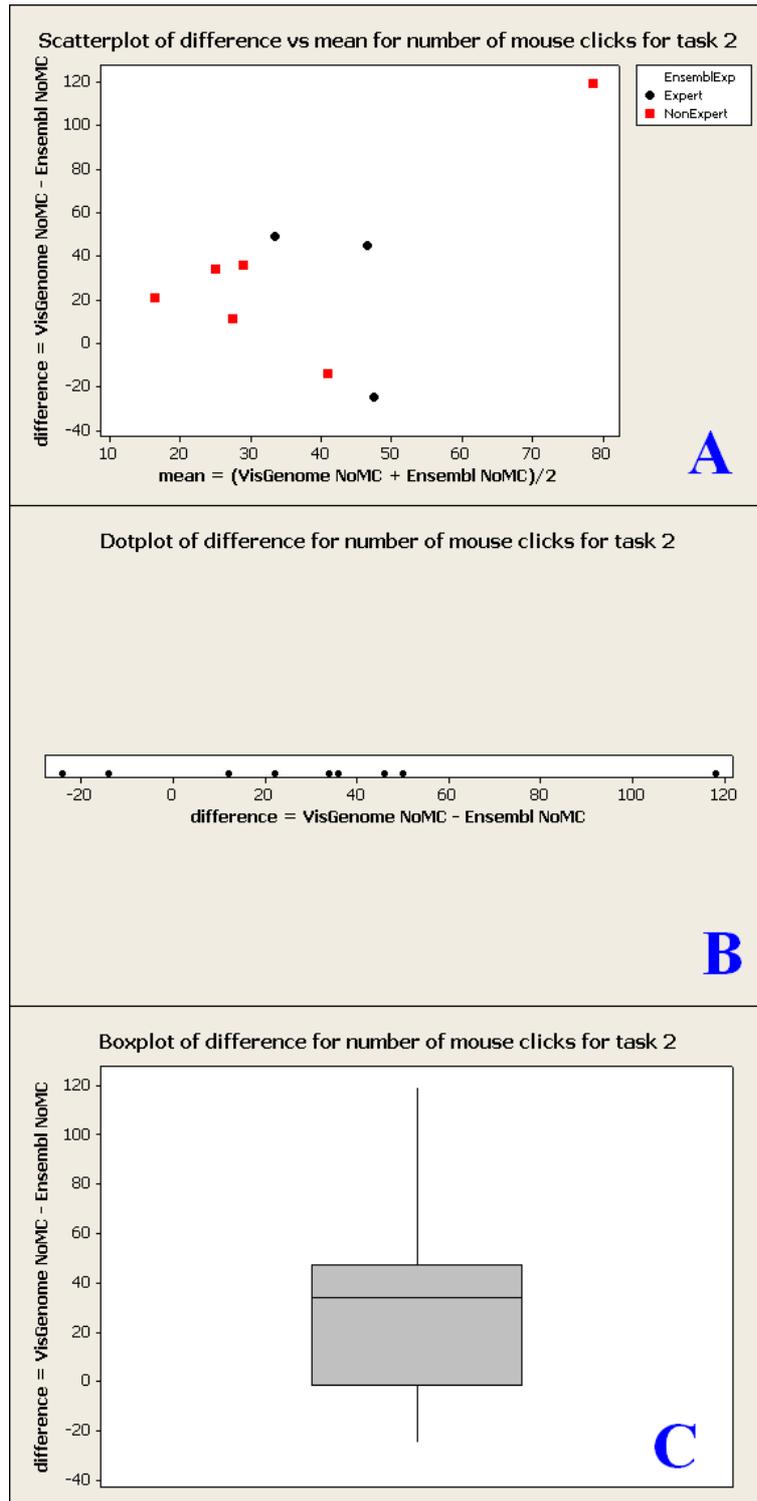


Figure 3.4: The Figure shows scatterplot (A), dotplot (B) and boxplot (C) for NoMC differences for the 9 successful users in T2. There were 6 NEx and 3 Ex. The graphs show that there was 1 NEx who used in VG more mouse clicks than in Ens - see A (78.5, 119) and B at 119.

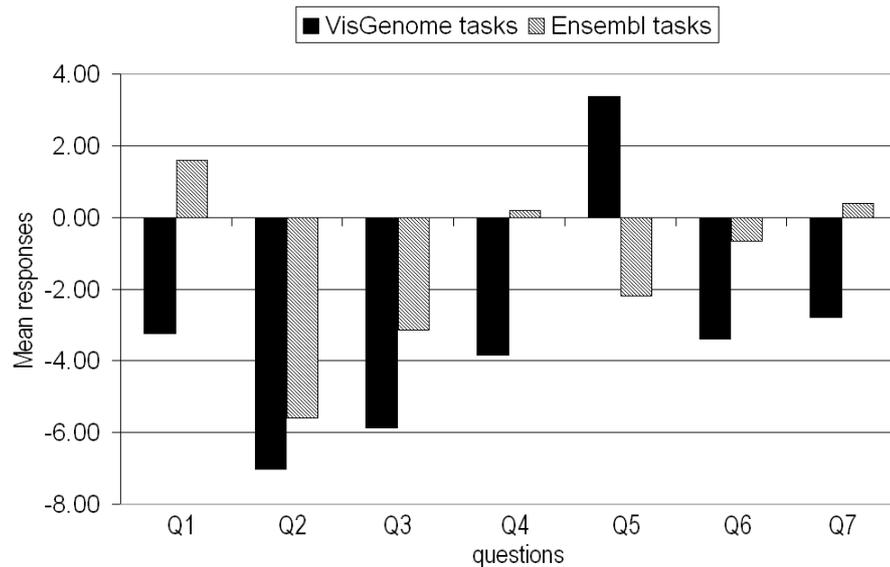


Figure 3.5: Mean response values for questions Q1-Q7 for 15 subjects. The subjects felt higher mental and physical demand during Ens tasks. They thought they worked harder and under higher time pressure in Ens. Subjectively, VG tasks were more successful and less frustrating. Mouse manipulation was more annoying in Ens tasks.

3.2.5 The user questionnaire results

The results of the user questionnaire are summarised in Figure 3.5 (note that means are given for Ens and VG separately, ignoring any pairing). All 15 subjects answered the following questions, once with regard to Ens and then with reference to VG.

Paired t-tests on the pairwise differences between Ens and VG gave significant results for Q1, Q3, Q4, Q5, and Q7. In Ens the subjects reported more mental and visual activity than in VG (Q1, see Figure 3.6). Population mean difference lies between -7.7 and -1.9 with probability 95% ($P=0.003$). Answers to Q3, see Figure 3.8, indicate that the subjects felt more rushed in Ens trial ($P=0.010$, 95% CI: (-4.7,-0.7)). Ens was perceived as being significantly harder in Q4, see Figure 3.9 ($P=0.011$, 95% CI: (-6.9,-1.1)). In Q5, see Figure 3.10, the subjects thought they were less successful in Ens than in VG. Population mean difference for Q5 lies between 3.3 and 7.7 with probability 95% ($P=0.000$). In Q6, see Figure 3.11, the subjects felt on average more frustrated with Ens than with VG, but this was not statistically significant. We have an additional observation here, gathered directly, that subjects were frustrated by the need to learn how to use the mouse in VG, but in Ens they were equally frustrated by the pop-up menus which suddenly appear and obscure the view, see Fig. 1.4, in panel B for gene XR.007958.1. Those menus may be one of the annoyance factors in the system. Population mean difference for Q7 lies between -6.1 and -0.2 with probability 95% ($P=0.036$), see Figure 3.12. Additionally the subjects were asked to state which of the two applications they prefer (Question 8, Q8). Significantly more subjects preferred VG to Ens ($P=0.013$; 1-proportion test). One subject preferred Ens, one did not answer Q8, two said that both tools were equal, while 12 preferred VG.

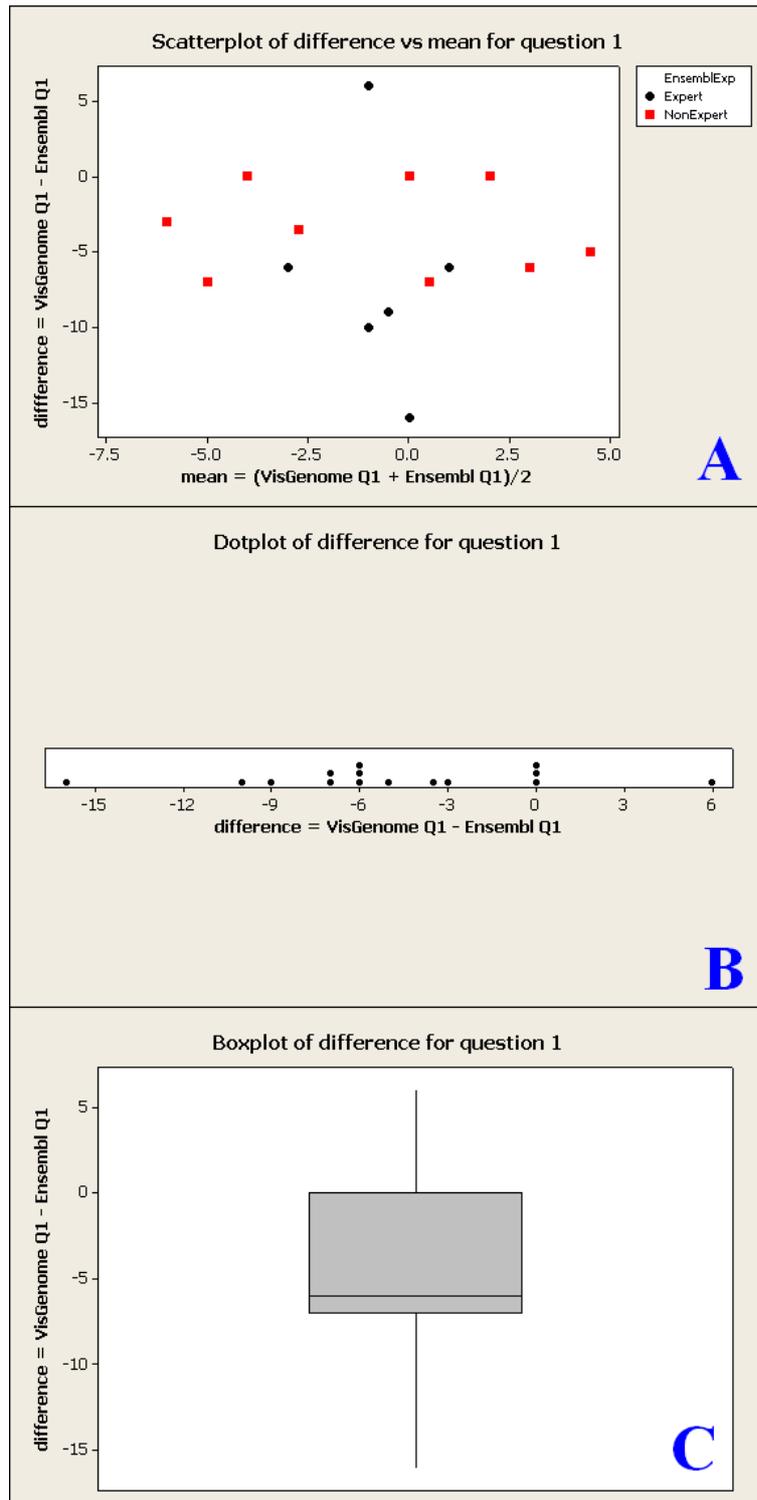


Figure 3.6: The Figure shows scatterplot (A), dotplot (B) and boxplot (C) for Q1 differences for the participants. 1 Ex thought that he performed more mental activity in VG, and 1 Ex thought he carried out more mental activity in Ens.

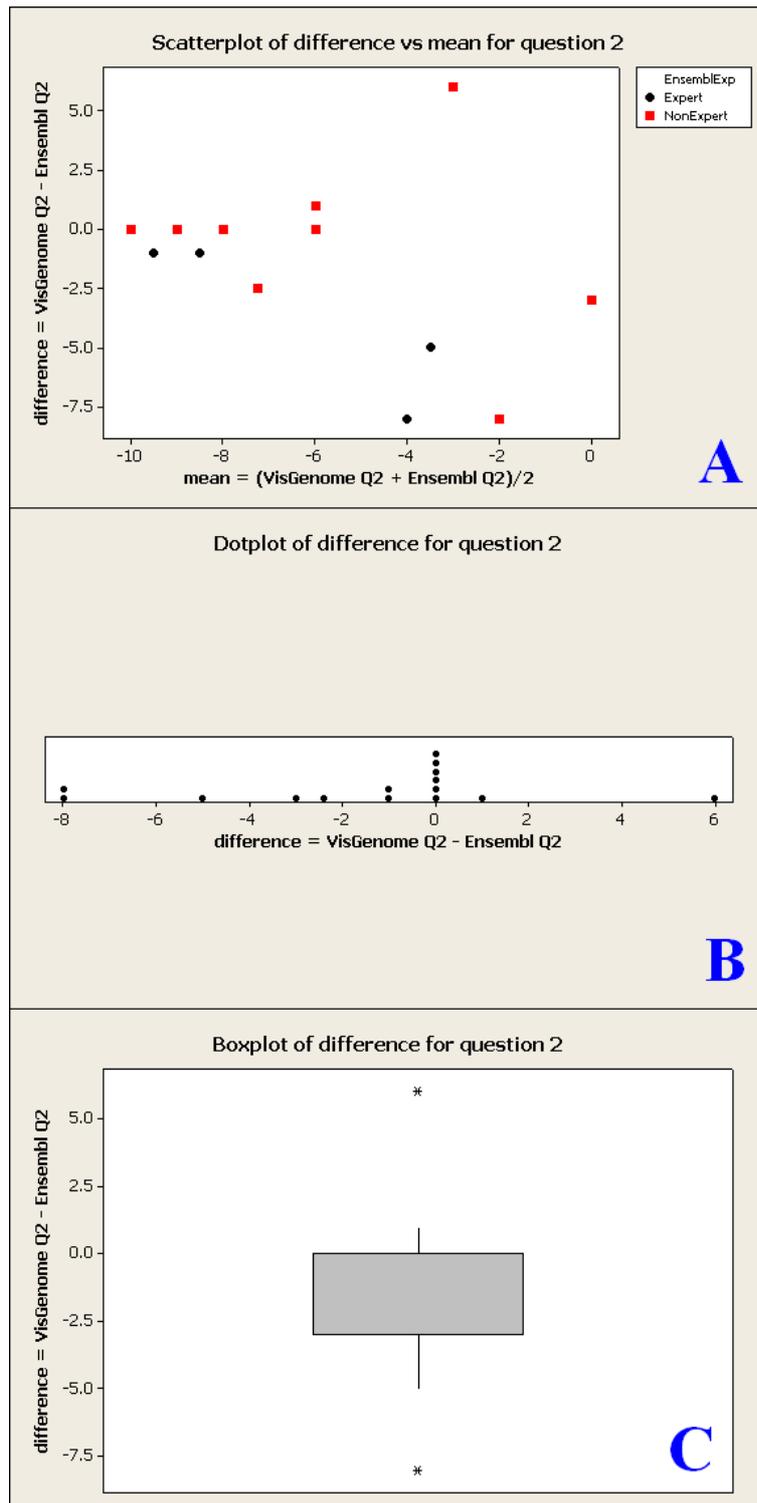


Figure 3.7: The Figure shows scatterplot (A), dotplot (B) and boxplot (C) for Q2 differences for the participants. 1 NEx found VG to be more physical demanding than Ens, and 1 NEx and 1 Ex found Ens to be more physical demanding than VG. Only 4 distinct values for Ex are visible in A, as they overlap two further values, however, B shows all 15 values recorded.

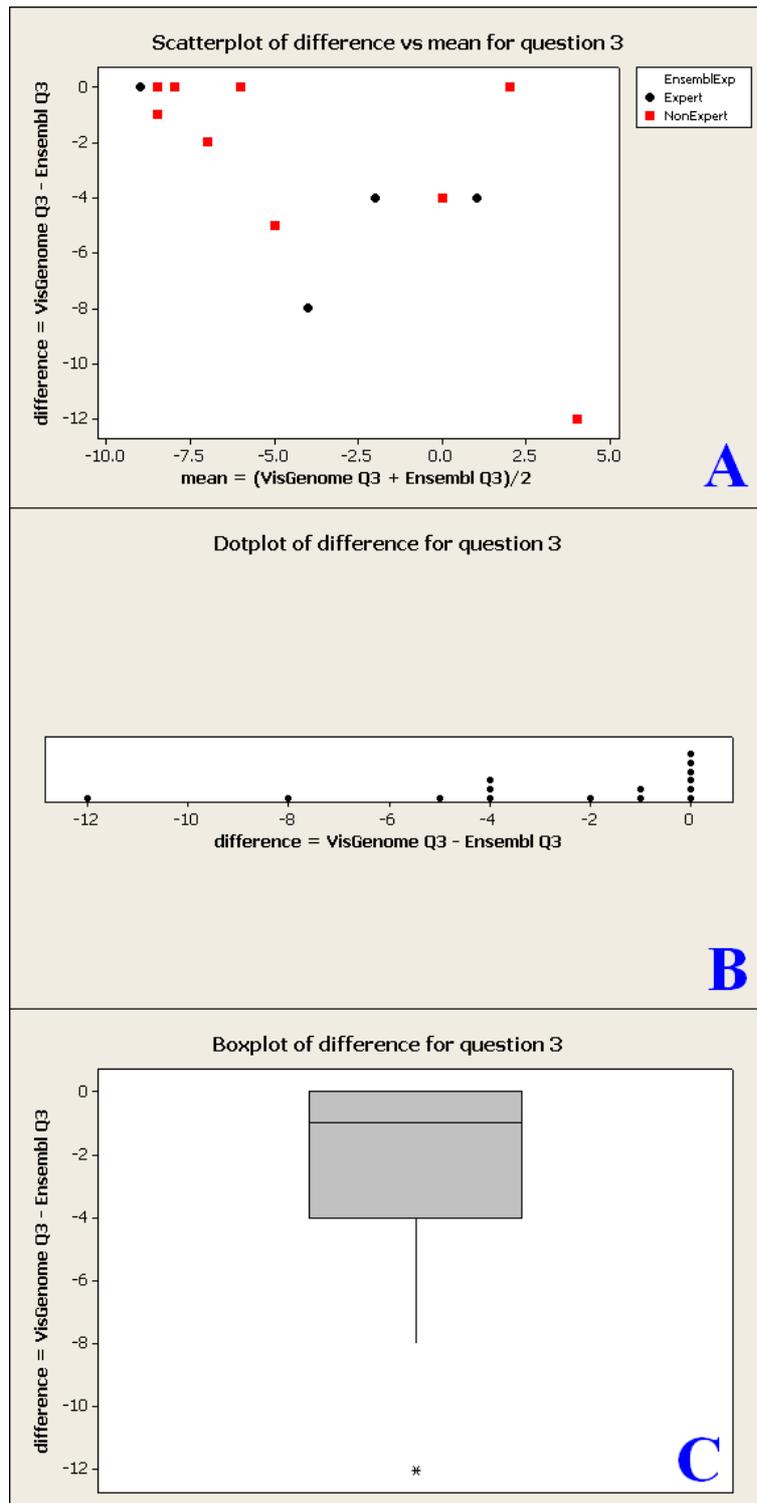


Figure 3.8: The Figure shows scatterplot (A), dotplot (B) and boxplot (C) for Q3 differences for the participants. No participant felt stronger time pressure in VG than in Ens. 1 NEx had an extremely large difference between VG and Ens for Q3 with respect to time pressure. Note that A shows 4 points for Ex, as those overlap two further points. B shows all 15 participants more clearly.

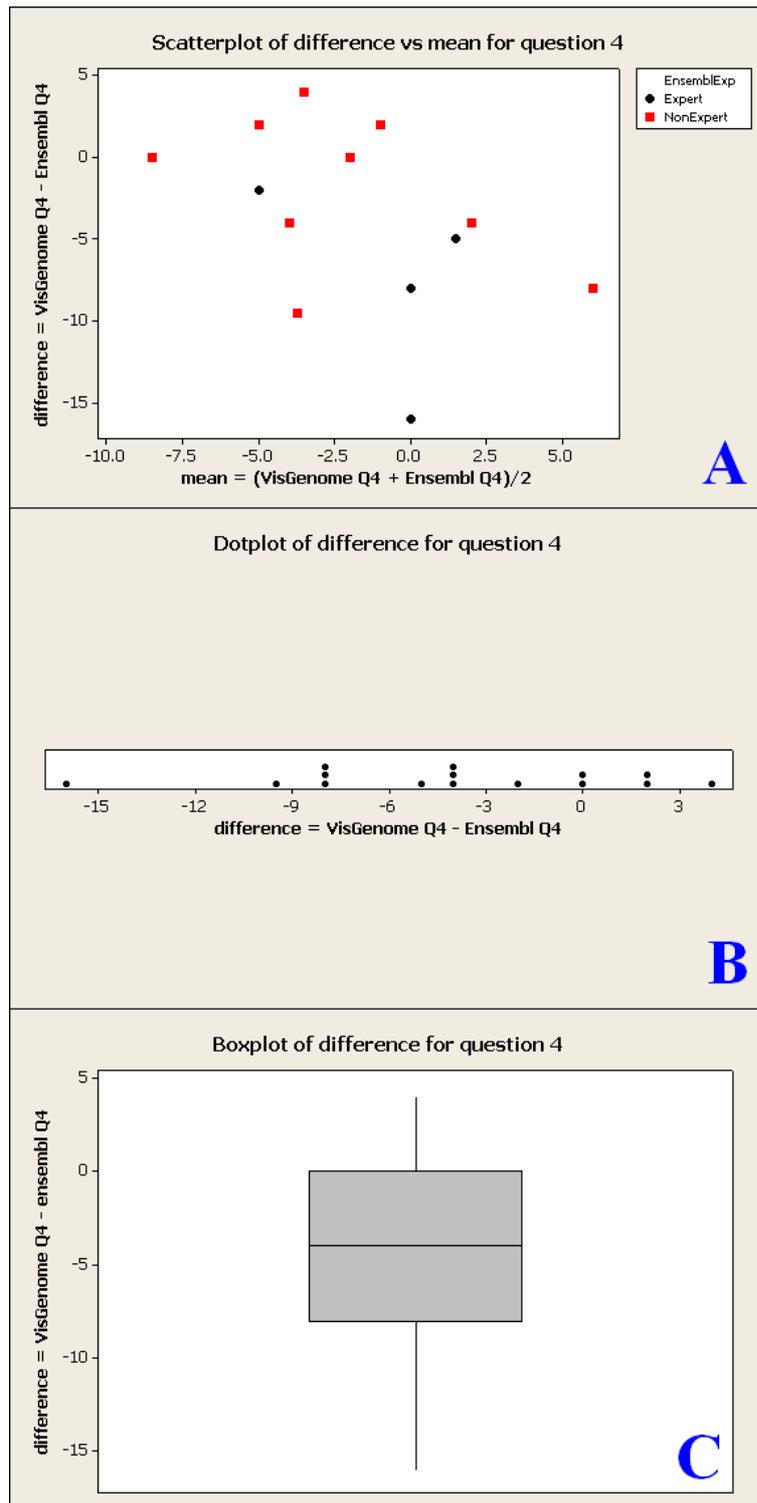


Figure 3.9: The Figure shows scatterplot (A), dotplot (B) and boxplot (C) for Q4 differences for the participants. 1 Ex thought he worked very hard in Ens - the black dot in A at (0, -16) corresponds to the extreme left dot on B at -16. A shows 4 distinct Ex values, representing 6 observations, while B shows all 15 participants.

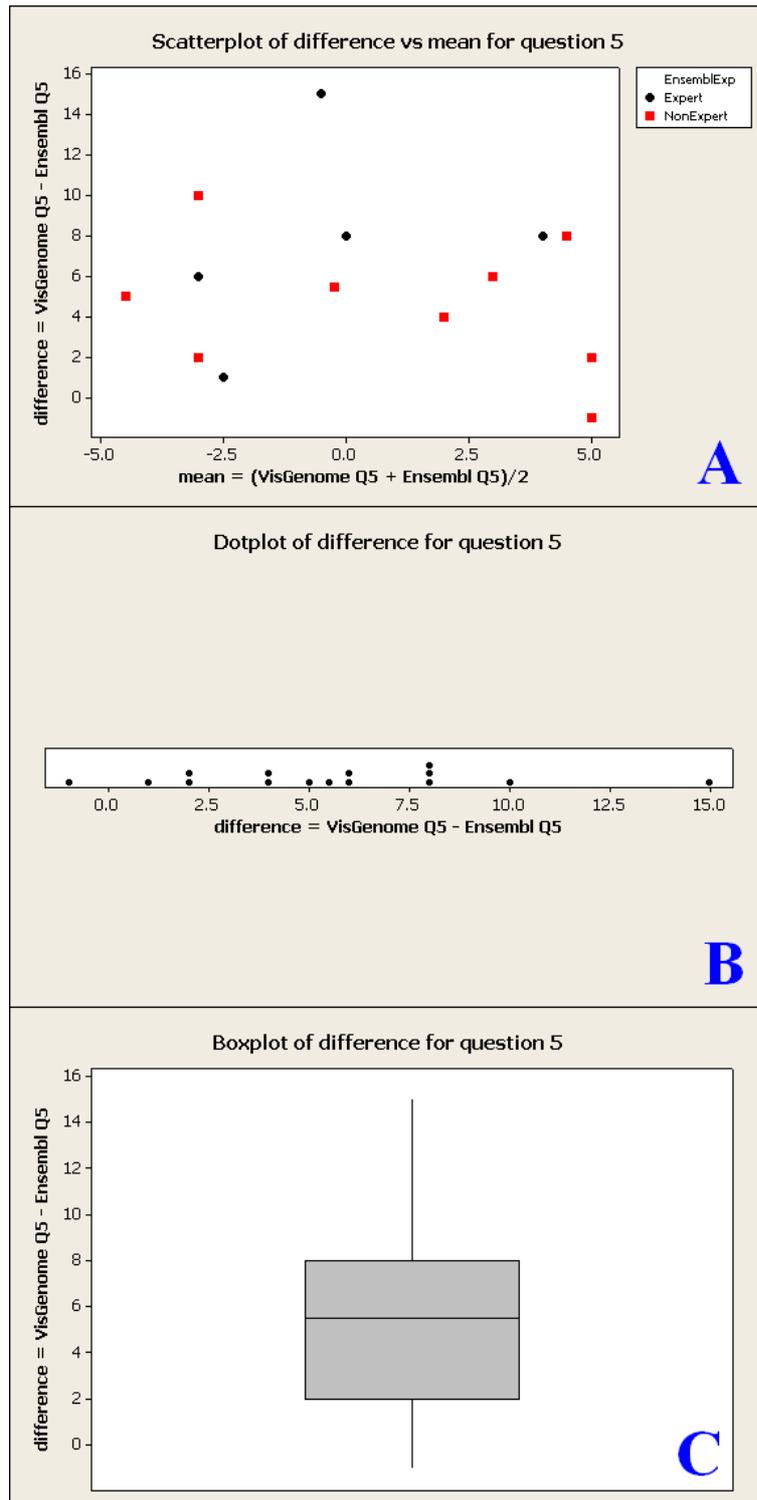


Figure 3.10: The Figure shows scatterplot (A), dotplot (B) and boxplot (C) for Q5 differences for the participants. 1 Ex was more successful in VG than in Ens - see A (-0.5,15) and B at 15. A shows 5 Ex, as one value represents two users, however, B shows all 15 participants.

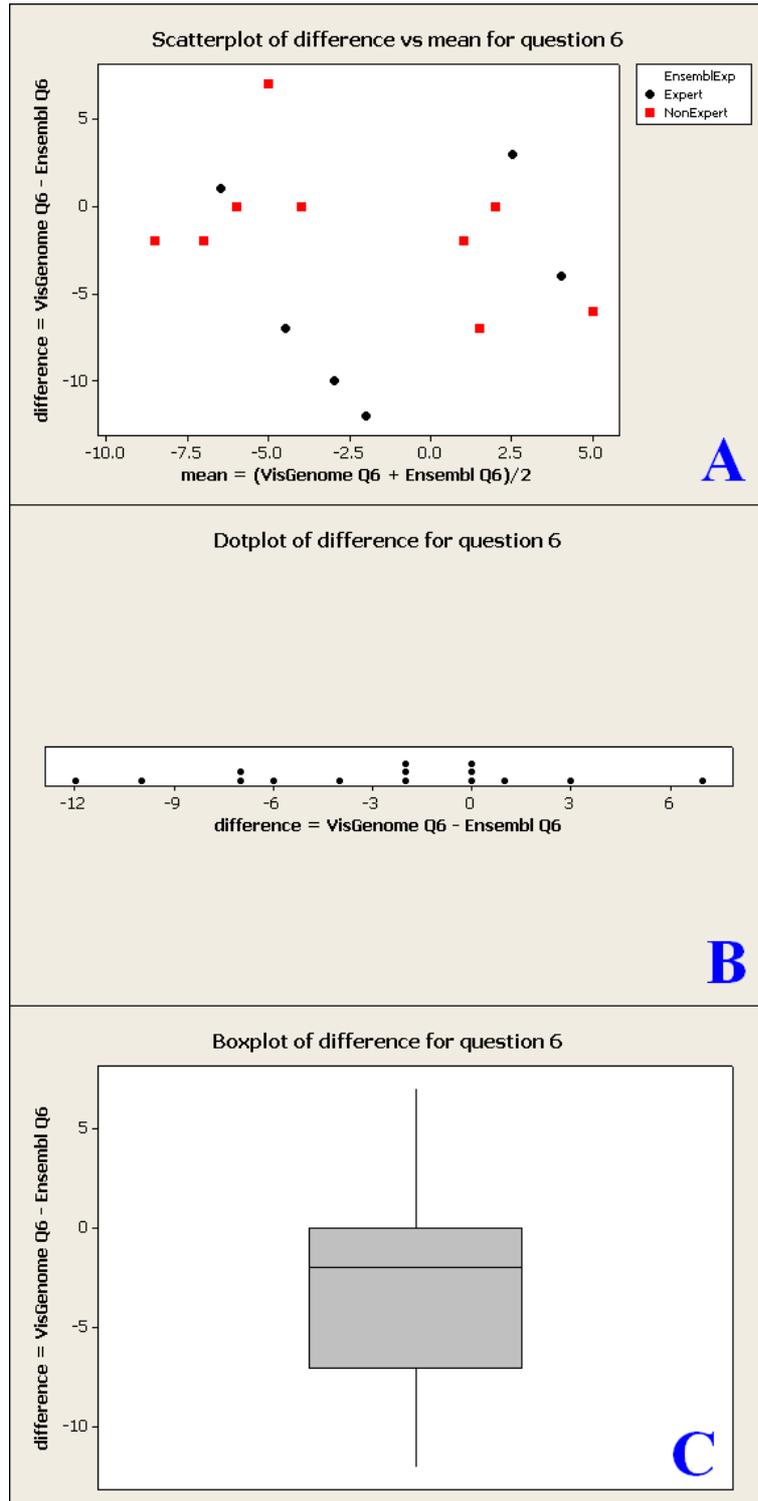


Figure 3.11: The Figure shows scatterplot (A), dotplot (B) and boxplot (C) for Q6 differences for the participants. The majority of the users were more frustrated in Ens, and 1 NEx was very frustrated in VG - shown in A at (-5,7) and in B at 7.

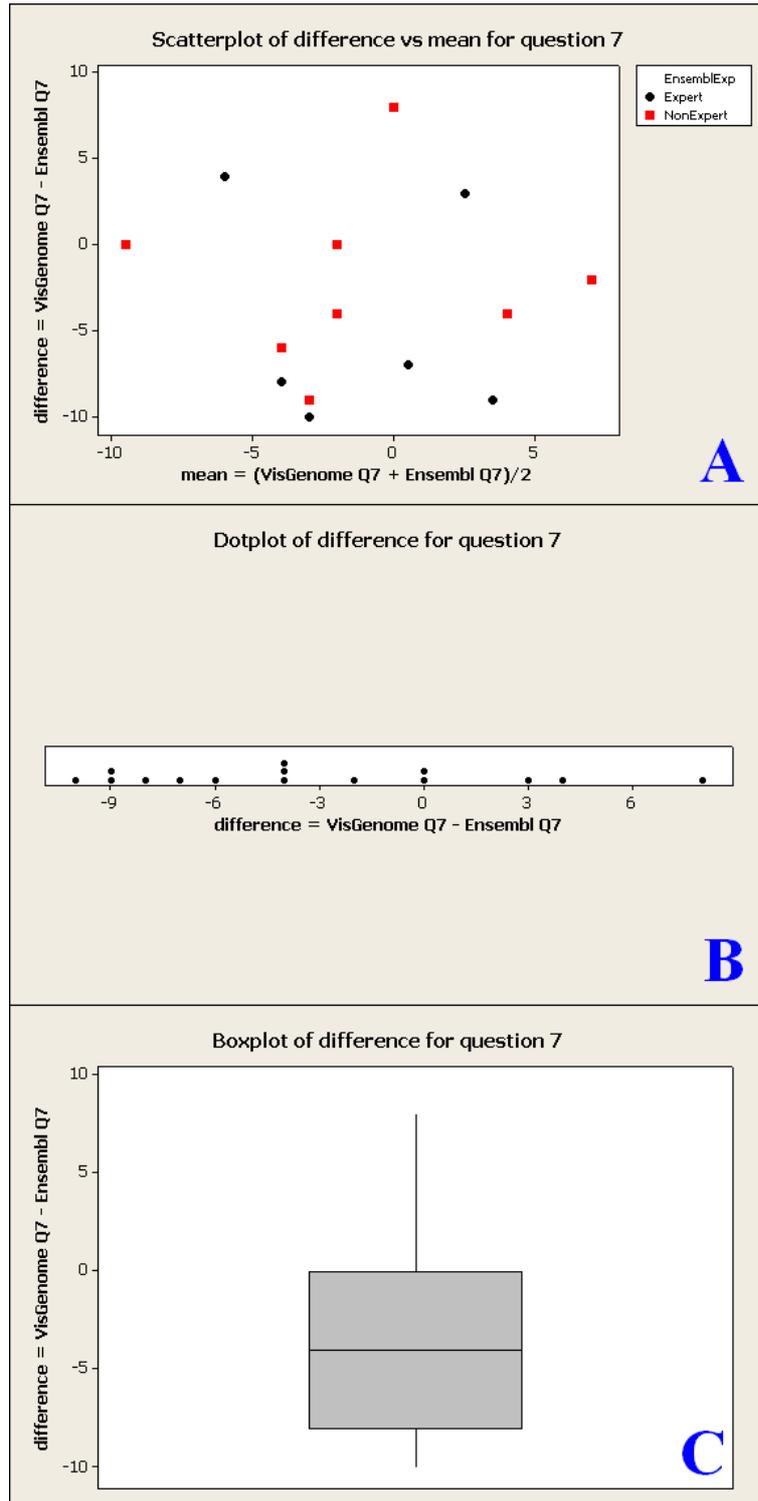


Figure 3.12: The Figure shows scatterplot (A), dotplot (B) and boxplot (C) for Q7 differences for the participants. No users found the mouse manipulations to be extremely annoying in VG or Ens. Some of them considered Ens and some VG to be annoying.

3.3 Additional Interview Questions

At the end of the experiment we briefly interviewed all the participants about visualisation techniques they know and they like in genome browsers. 12 subjects know Fisheye [19] and see it as useful and 11 users like it (Q15). In response to (Q16) 11 participants said that excentric labeling [20] is useful and 10 like the technique. Zooming and panning (Q17 and Q18) are common in genome browsers and the users like them, while 14 subjects use zooming and 13 like it, and 14 users use panning and 14 like it. Only two persons preferred zooming via buttons than mouse action (Q19).

We asked the users about the use of colour and if it has any meaning for them (Q22 and Q23). This is an interesting issue, as Ens offers a lot of colours and VG has a monochromatic display for the karyotypes, and all other data is coloured white in the current version of the application, with the exception of two participants where private gene expression data were coloured red and green. Only 8 persons answered that colours were meaningful. They would like to have the option to change the colour to mark interesting data. The subjects stressed as well that for them Ensembl colours have no meaning (only for one participant Ens colouring is meaningful). The subjects believe that colour only shows the grouping of data items, but if Ens offered horizontal lines instead of colours, this would be also a good solution. Some of the participants, specially from BRC, touched upon the problem of colour blindness where some colours may have no meaning at all.

In (Q24) we showed the users Figure 3.13 and asked them which karyotype representation they prefer. 6 participants preferred A, 7 liked B better (2 under the condition that they can click on the chromosomes and not drag them) and 1 person was not sure if they prefer B or C. The one user who liked view C motivated the choice by saying that it takes less screen space and in the future can allow the developer to add more species.

The experimental version of VG used dragging in single representation and clicking in comparative representation. We wanted to check what is preferable. During the experiment we observed that the participants prefer clicking on a chromosome to dragging it into a display. Only one user preferred dragging because in VG it allowed him to see that data is being downloaded. The majority of the participants clicked on a chromosome in the single representation in VG and waited, and when nothing happened, the subjects recognised that they ought to drag instead of click. The participants liked the info panel (8 users strongly recommended it - Q25) instead of keeping everything in the main view.

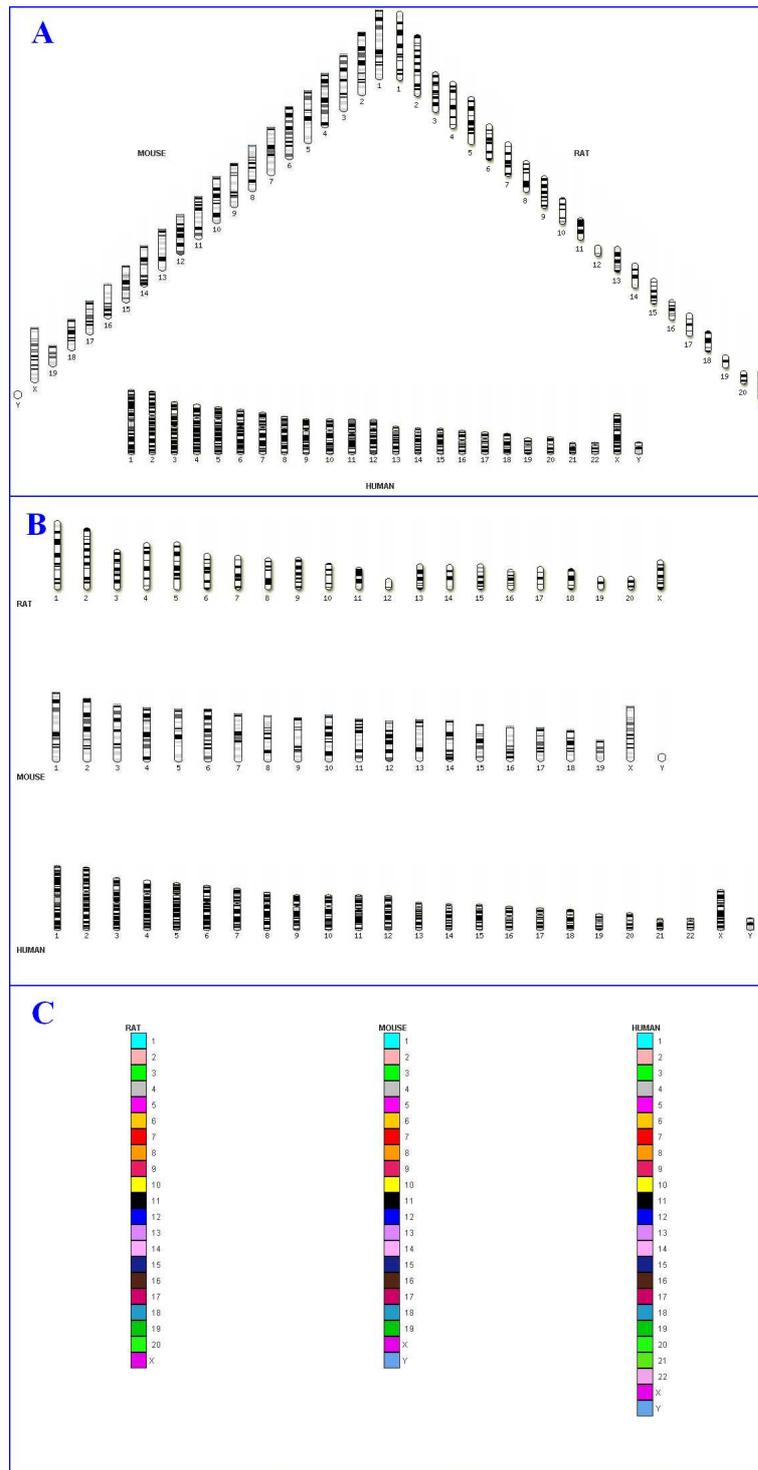


Figure 3.13: The three views for marking which chromosome the users would like to see in detail, part of Q24.

3.4 Discussion

In T1 we saw that the participants were looking for Affymetrix probes and couldn't find them. However, the main cause of failure in T1 was that the subjects made mistakes and typed 1 Mbp instead of 10 Mbp. In VG the subjects frequently forgot to mark the whole chromosome to show all available data or marked half of the chromosome instead of the whole. In Ens a number of users entered the coordinates and marked 'Region' instead of 'Base pair', and some did not use the overview offered by Ens but tried to mark the whole chromosome in ContigView. This usually crashed the web browser and required a restart. As nobody was successful in T1 in Ens, we created a 'light' version of T1, named T4. T4 required showing the last gene in the region instead of the last Affymetrix probe, and it did not require an overview. 53% of subjects succeed in T4 in Ens and 73% in VG, see Figure 3.14 and Figure 3.15. As VG was designed to enable this type of visual query, this confirms that our viewer fulfilled this requirement.

T3 required showing the longest QTL. In a chromosome with many small QTLs, the subjects could not decide which QTL to choose (four subjects). We suggested that they carry out the task for any of the QTLs. The same solution was suggested where several long QTLs appeared to be of similar length. 8 researchers were successful in T3 in VG. The most frequent mistake in the unsuccessful attempts in VG was choosing a complex of QTLs instead of one QTL. In Ens the subjects usually attempted to mark the entire chromosome, and only one person succeeded without crashing the browser. Some subjects tried viewing the chromosome in units of one 1 Mbp but gave up after recognising that this would take too long. One user tried to use BioMart and RGD, but this did not help. Most subjects did not realise that the view shown in Ens is not the whole chromosome but a small part of it. Several subjects chose a chromosome, clicked on it, viewed ContigView, looked down the screen to find QTLs and saw that they were all longer than the area shown in the browser, and did not know what to do to see the entire length of the QTLs.

Our experiments prove that Ensembl was designed for local data analysis, while VG supports a different requirement, that is cross-species QTL analysis. We note user training is required for both VG and Ens. Zooming and panning by mouse manipulation was classified as something very intuitive and natural, but at the beginning of the VG experiment, most subjects were confused and disappointed that they had to remember which button and which direction to use to zoom or pan. Some suggested that new visualisation techniques could be bad because biologists are not familiar with them, however they said that acceptance depends on the implementation. Two subjects suggested zooming with buttons instead of mouse manipulation and were disappointed because of the lack of scrolling.

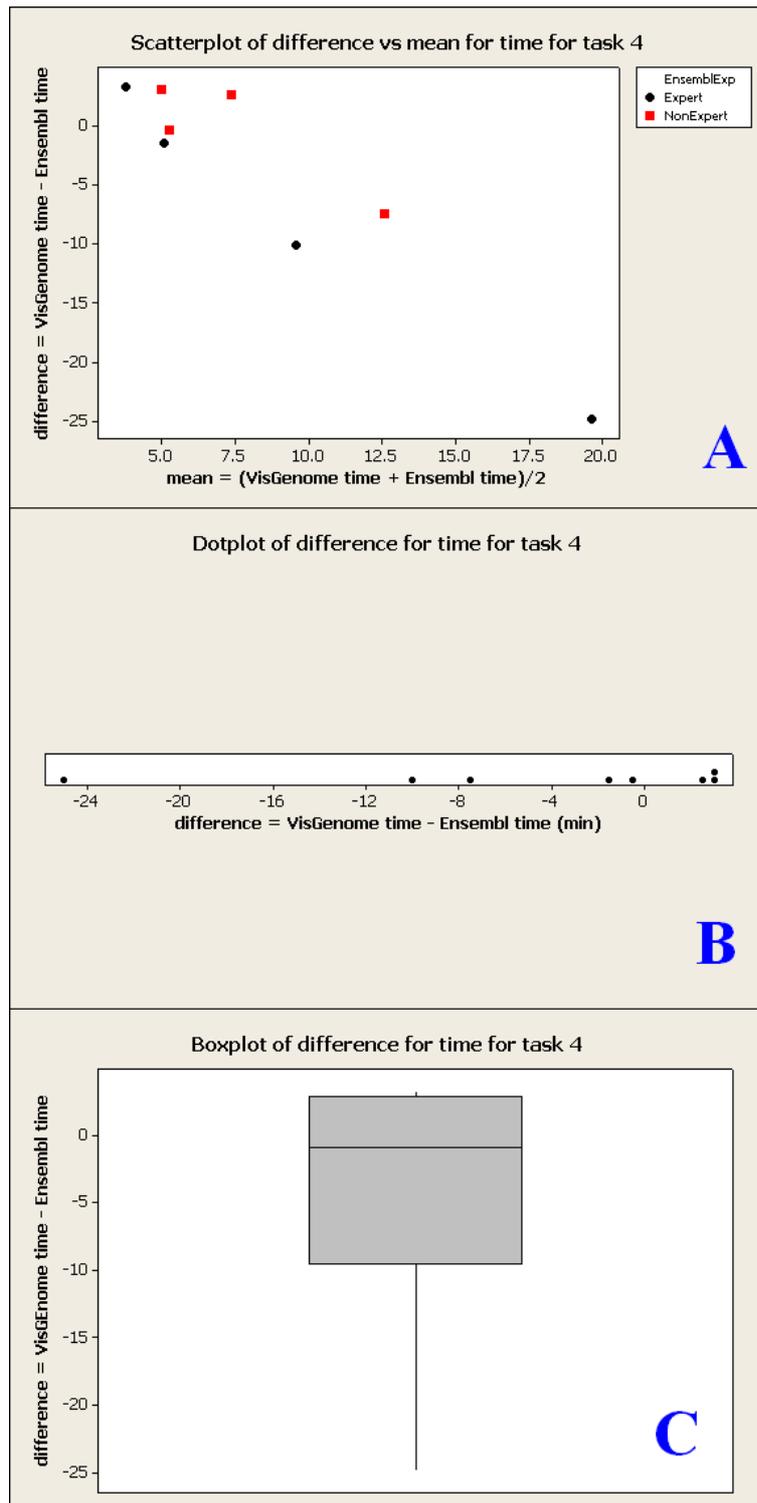


Figure 3.14: The Figure shows scatterplot (A), dotplot (B) and boxplot (C) for time differences for the 8 successful users in T4. A shows 4 NEx and 4 Ex. The data points divide into 3 clusters: 2 Ex and 3 NEx with similar time in Ens and VG (see A (7.3, 2.5), (3.7, 3.2), (5.1, -1.5), (4.9, 2.9), and (5.2, -0.4) and B at 2.5, 3.2, -1.5, 2.9, and -0.4), 1 NEx and 1 Ex with longer time in Ens (see A (12.5, -7.5) and (9.4, -10.5) and B at -7.5 and -10.5), and 1 Ex with an extremely long execution time in Ens (see A (19.6, -24.8) and B at -24.8).

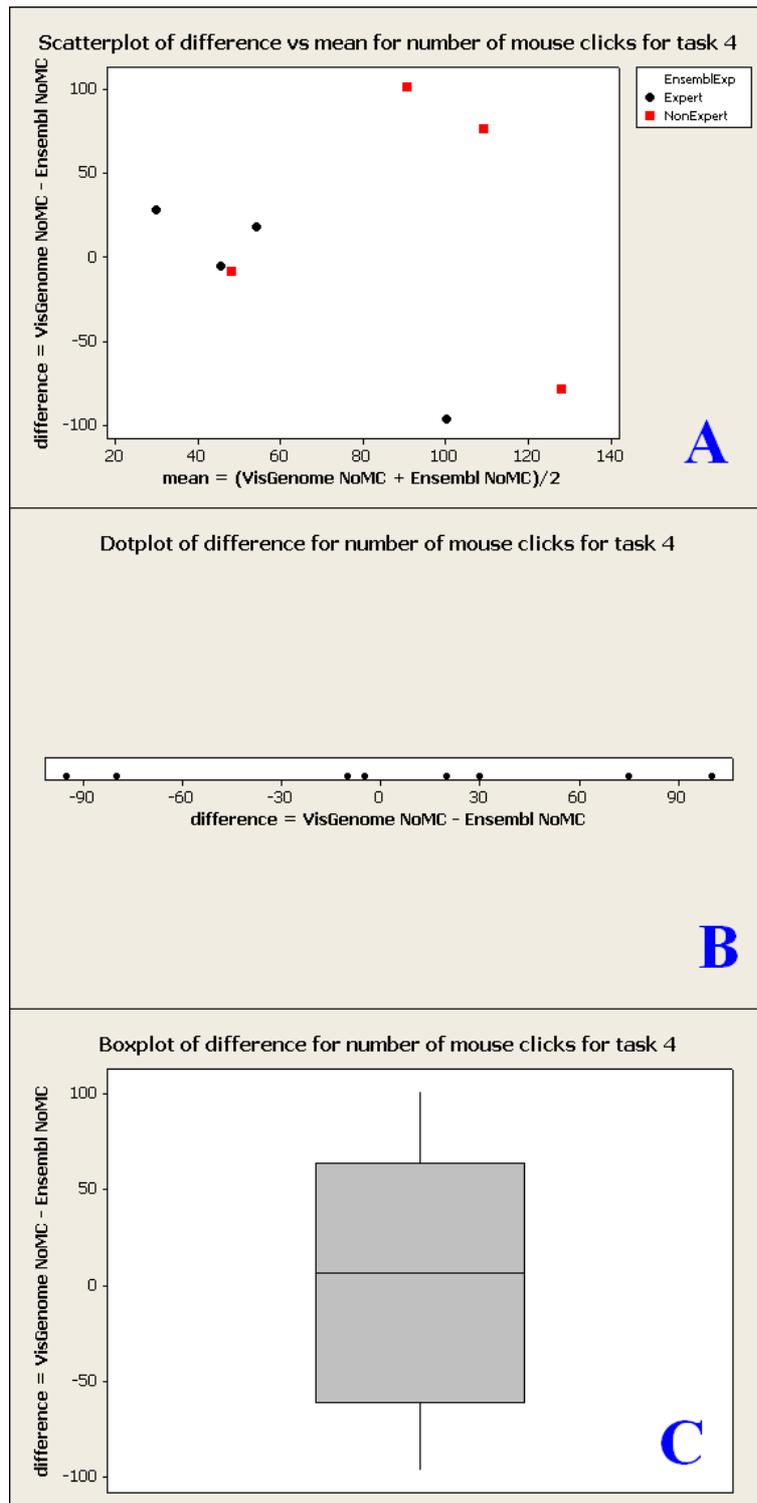


Figure 3.15: The Figure shows scatterplot (A), dotplot (B) and boxplot (C) for NoMC differences for the 8 successful users in T4. A shows 4 NEx and 4 Ex. The data form 3 clusters: 3 Ex and 1 NEx with similar NoMC in Ens and VG (see A (30, 28), (45.5, -5), (48, -8), and (56.5, 18) and B at 28, -5, -8 and 18), 2 NEx with a larger NoMC in VG (see A (90.5, 101) and (109, 76) and B at 101 and 76), and 1 Ex and 1 NEx with a larger NoMC in Ens (see A (128, -78) and (100, -96) and B at -78 and -96).

Conclusions

Biological data is difficult to visualise and analyse due to the large amount of information that needs to be represented. User studies are needed to find out what the success factors are, as success is domain bound, and involves factors such as genomic coordinates or relative object size, and also reflects the research focus. Our work compared Ensembl and VisGenome in three data localisation tasks, and gathered user impressions after the experiment. We found out that in our experimental setup the subjects were more successful in using VG than in Ens. VG was preferable in some aspects, as it showed less data and had fewer controls. Both Ens and VG require training to support efficient tool use in research. All subjects liked techniques they know, such as scrolling and panning, and need time to adapt to new solutions. However, they were also receptive to new methods. The subjects provided a number of useful suggestions which will be used to improve VG functionality. The next version of VG will be evaluated via a user study. We may also explore ways of presenting additional information, such as data digests from other sources, alongside the genes and other objects on the map. VisGenome is available at www.dcs.gla.ac.uk/~asia/VisGenome.

Acknowledgements

We thank all the participants for their contribution to this work and Helen Purchase for advice in user study design. EH is an EU Marie Curie fellow and JJ is funded by the MRC, UK.

Bibliography

- [1] E. Hunt, et al. *The Visual Language of Synteny*. OMICS, **8**(4):289-305, 2004.
- [2] J. Jakubowska, et al. *Granularity of genomics data in genome visualisation*. University of Glasgow, Tech. Rep.: TR-2006-221, 2006. www.dcs.gla.ac.uk/publications/PAPERS/8212/AsiaElaMatthewCHI06.pdf
- [3] T. Hubbard, et al. *Ensembl 2005*. NAR**33**, DB issue:D447-D453, 2005.
- [4] B. B. Bederson, et al. *Toolkit Design for Interactive Structured Graphics*. IEEE Trans. Soft. Eng., **30** (8):535-546, 2004.
- [5] M. W. McBride, et al. *Functional genomics in hypertension*. Curr Opin Nephrol Hypertens **15**(2):145-51, 2006.
- [6] Robin D. Dowell, et al. *The Distributed Annotation System*. BMC Bioinformatics, **2**(7), 2001.
- [7] G. Fischer, et al. *Expressionview: visualization of quantitative trait loci and gene-expression data in Ensembl*. Genome Biol **4**:R477, 2003.
- [8] R. Pausch, et al. *A User Study Comparing Head-Mounted and Stationary Displays*. IEEE Symp. on Research Frontiers in VR, 41–45, 1993.
- [9] R. Dhamija and A. Perrig. *Déjà Vu: A user study using images for authentication*. USENIX Sec. Symp., 2000.
- [10] D. Pasko, May 2000. Overview of Rat Research Today. rgd.mcw.edu
- [11] Ensembl BioMart. www.ensembl.org/Multi/martview
- [12] D. P. Leader. *BugView: a browser for comparing genomes*. Bioinformatics **20**:129–130, 2004.
- [13] D. Karolchik, et al. *The UCSC Genome Browser Database*. NAR**31**, 51–54, 2003.
- [14] X. Pan, et al. *ATIDB: Arabidopsis thaliana insertion database*. NAR**31**(4), 2003.
- [15] R. Stevens, et al. *A classification of tasks in bioinformatics*. Bioinformatics:**17**(2), 2001.
- [16] Min Wu, et al. *A fisheye viewer for microarray-based gene expression data*. BMC Bioinformatics **7**:452, 2006.

- [17] L. Slaughter, et al. *Assessing users' subjective satisfaction with the Information System for Youth Services (ISYS)*. VA Tech Proc 3rd Mid-Atl. Human Factors Conf., 164-170, 1995.
- [18] Y. Yang, et al. *Integration of metabolic networks and gene expression in virtual reality*. *Bioinformatics*:**21**(18), 2005.
- [19] G. W. Furnas *Generalized Fisheye Views*. CHI, 16-23, 1986.
- [20] J. D. Fekete and C. Plaisant *Excentric Labeling: Dynamic Neighbourhood Labeling for Data Visualization*. CHI, 512-519, 1999.

F: Granularity of genomics data in genome visualisation

Asia Jakubowska
Department of
Computing Science
University of Glasgow
Glasgow, G12 8QQ
Scotland
asia@dcs.gla.ac.uk

Ela Hunt
Database Technology
Research Group
Department of
Informatics
University of Zrich
CH-8057 Zrich
ela@dcs.gla.ac.uk

Matthew Chalmers
Department of
Computing Science
University of Glasgow
Glasgow, G12 8QQ
Scotland
matthew@dcs.gla.ac.uk

ABSTRACT

Biologists collect genomic data of increasing complexity. New technologies give rise to new data types and the volume of both raw and processed data is growing fast. Biomedical researchers would like to analyze the data using user-friendly interfaces, however, the tools, computer monitors and machines have their limitations and do not always precisely show the data under investigation. We survey different genomics visualisation software including AceDB, SyntenyVista, DerBrowser, Apollo, Artemis, BugView, Ensembl, Sockeye, K-BROWSER, GBrowse, NCBI MapViewer, eQTL Explorer, and Expressionview.

This paper presents a short survey of genomic browsers and visualisation effects which were used or can be used in such applications. It presents a classification of genome browsers according to three dimensions and argues the need for a new browser which offers improved zooming functions. This leads us to introduce a new version of SyntenyVista, VisGenome, which allows the user to visualise single and comparative representations of the rat, the mouse, and the human genome at different levels of detail.

Author Keywords

genomics visualisation

INTRODUCTION

Current genomics visualisations are inadequate in many respects, as they do not allow for flexible view adjustment. We are aiming to derive general principles of data representation and visualisation usability for genomics. We would like to find a solution which will clearly present the information, including all relevant information the biologists wish to see. We study existing visualisation solutions in order to find out what features they offer, which of those correctly support data analysis, and which are not helpful. Our study will allow us

to find a better solution for data analysis which overcomes cognitive problems. We would like to discover how best to compare data coming from various sources and experiments in a biological setting.

Our work focuses on the use of visualisation to support the understanding of very large data sets. With an eye to create an universal solution, we are collaborating with biologists who use genome browsing tools, such as that we create, in their everyday work. We are aiming to solve in VisGenome both the visualisation problems and some of the database integration problems. We would like to offer a clear presentation of the data the biologists wish to see.

A BIOLOGICAL INTRODUCTION

In this section we motivate our work and introduce the concepts used in this paper. Biomedical and agricultural research is motivated in two ways. One is to acquire new knowledge and understand how living organisms function, and the other is to improve our lives. Improvement is the treatment or prevention of diseases, better diagnosis, new medications, new crops, and better understanding of the environmental impact our technologies have.

Genomics is the study of genomes and of the relationship between genomes and the way an organism functions. Each living organism has a genome which encodes information passed down from generation to generation. A bacterial genome consists of several million DNA molecules (tuberculosis genome is about 5 million long). A human or mouse has a genome of around 3 billion letters of DNA code. A genome is encoded in DNA or RNA molecules of four types (A, C, G, T for DNA). It encodes all proteins and signalling molecules needed by an organism. Only 1.5% of the human or mouse genome is translated into proteins which are the building blocks of our bodies. Chemically, they are strings of amino-acids, where each three letters of DNA correspond to one amino-acid. Proteins use an alphabet of 21 letters, and in our bodies they fold into *3-D structures* (see Fig. 2H) which may change conformation as they perform their various functions. We do not know exactly how many genes the humans have, with the current estimate being between 20 and 30 thousand. Those give rise to probably around 1 million proteins. The process of translation from DNA to protein is com-

plex, and it is important to remember that a stretch of DNA of some 30 thousand letters gives rise to a protein of some 300 letters. The parts of DNA which translate into protein are called *exons* while the parts which control the process are called *introns* or untranslated regions. Biologists want to know for each protein what gene produced it, which parts of the gene were used in this particular protein and which control regions were activated during the production process. The process of protein production is dependent on the type of cell, developmental stage, the environment, and many other factors which altogether influence the health of an organism.

Genomes of a very large number of animals are known relatively well, and are publicly available, along with genome maps which show how genes are arranged and structured. Mammalian genomes are split into around 20 chromosomes, and the set of chromosomes forms a *karyotype* (see Fig. 2A), while bacterial genomes form a circle. Groups of genes that are shared between related organisms are often collocated in the so-called *synteny groups*, and biologists study such gene groups, as there is proof for synchronised activity over groups of genes, and for similar gene functions shared between related organisms. Similar gene functions arise from similar DNA and protein sequences, and the biologists *align* (see Fig. 2I) genomes and genes to understand what sequences are shared, and what functions are common to a group of organisms.

Genes and the resulting proteins interact and form *pathways*. Such pathways stand for chemical and structural reactions which orchestrate all the processes which keep us alive. Pathways may be shared by groups of organisms but there are known cases where they diverge. Pathway visualisation tools include [32].

Very large numbers of genes have no known function, and genes responsible for common diseases like hypertension are not known. It is assumed that such diseases are controlled by a number of genes, and are under strong influence of the environment (diet, smoking, exercise). The search for disease genes uses the techniques of gene mapping, where populations of subjects are tested and a statistical correlation between a part of a chromosome, containing a number of genes, and the disease is expressed as a quantitative trait locus (QTL) [21]. The study of QTLs leads to the identification of genes which are candidate genes first, until it is proven that they are the cause of a disease. The study of QTLs is easier in animals (rat, mouse) because they are bred to be genetically identical, while human genomes have a variant DNA letter every few thousand letters for any two humans, and that is why statistical correlations are harder to make. Diseases are often studied in animals, and then the candidate human gene will be sequenced (from the blood samples gathered from patients), and subjected to further analysis which may uncover the biochemical causes of disease.

Biologists are faced with very large data sets. A QTL may contain around a hundred genes, or a few million letters of DNA code. On the other hand, single nucleotide polymorphisms (SNPs), which are individual DNA

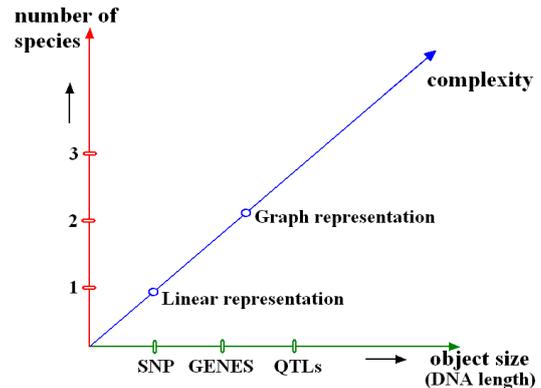


Figure 1. Genome browser classification schema.

differences, are one letter long, and need to be shown along QTLs. Visualisation is the only viable way of making this data available, as close reading of thousands of letters is not a solution. That is why genetic databases visualise data in the form of maps which show linear arrangement of genetic features. To our knowledge, the resulting visualisations have not been subjected to much scientific scrutiny, so far. They are used by thousands of scientists daily, but it is not clear how they should be designed and how well they support scientific activity. It is our aim to study this, and to deliver better visualisations which can enhance the process of scientific discovery.

USER SCENARIO

We cooperate with a number of biological research groups who work in the areas of cardiology, metabolic diseases, schizophrenia and cancer. Those researchers conduct large scale experiments using *micro array* technology. In a micro array experiment the activity of all genes is examined simultaneously. What is measured is gene expression, that is the amount of the intermediate product, produced by the DNA, and leading to the production of a protein or a gene control element. The interpretation of such experiments requires simultaneous visualisation of chromosomes, genes, micro array probes, markers, and QTLs in three species: the mouse, the human, and the rat. Additionally, SNPs which may harbour DNA mutations causing a disease need also to be shown, along blocks of SNPs shared by population groups, and called haplotypes (www.hapmap.org).

CLASSIFICATION SYSTEM

We classify genome browsers according to tree dimensions, see Fig. 1. In the first dimension (number of species) we find that genome browsers represent between one and many species. Ensembl can be used to view one species at a time but other species information can be superimposed (see Fig. 5C). K-BROWSER can show a number of species and the number is limited by the size of the web page (see Fig. 7). Multiple alignment tools can show a number of aligned sequences from different species and those sequences can also be shown as a tree, see Fig. 2I (alignment) and Fig. 2J (phylogeny).

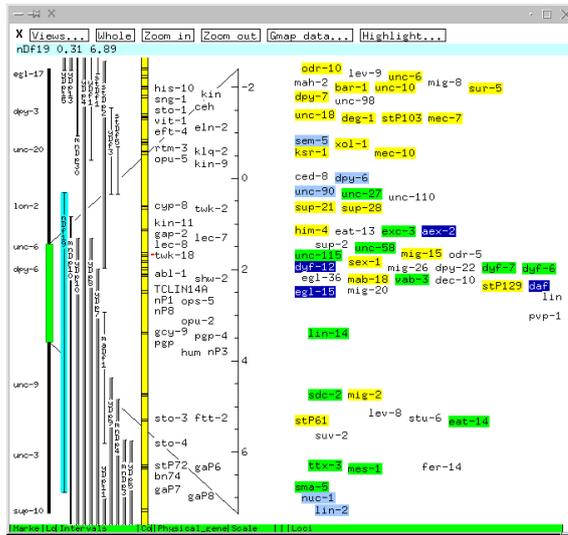


Figure 3. AceDB-representation of worm chromosome X.

The second dimension represents the size of the objects shown. The smallest objects are one DNA letter long (SNPs). In the order of increasing size one can show gene promoters, exons and introns, and other constituent parts of genes. Genes are about 20-30 thousands of DNA letters long and QTLs may contain thousands of genes. QTLs may approximate chromosome bands in size. Finally, human chromosomes are between 50 and 300 million letters of DNA long.

From the point of view of representation complexity, we classify browsers into linear and graph representations. In graph representations we can distinguish trees, networks (pathways), and 3D structures (proteins). JMol [25], see Fig. 2H, is one of the viewers showing protein 3D structure, while Treeview, Fig. 2J, offers a tree representation of a phylogeny. BugView (Fig. 2F), Ensembl (Fig. 2B) and SyntenyVista (Fig. 2D) show genome comparisons as bipartite graphs.

SURVEY OF GENOME BROWSERS

In this section we survey genomics visualisation software such as AceDB [29], SyntenyVista [1], DerBrowser [2], Apollo [3], Artemis [4], BugView [5], Ensembl [6], Sockeye [9], K-BROWSER [10], GBrowse [11], NCBIMapView [12], eQTL Explorer [14] and Expressionview [22]. We compare the systems in order to understand the problems and possible solutions to data visualisation.

AceDB

AceDB [29], see Fig. 3, is one of the first tools for genome visualisation. It offers a graphic representation which contains many objects in various colours. Colours help the researcher to identify the objects. For example, when a marker is coloured in yellow, it means that this marker has been cloned. The users can view textual details by double clicking on an object. AceDB offers simple zooming activated via zoom buttons. The viewer offers three types of sequence view: a genetic map, a physical map, and a sequence window which

shows the DNA or AA letters. All views offer pop-up menus.

SyntenyVista

SyntenyVista [1], see Fig.2D, is the first interactive representation of synteny data designed for large genomes. It shows information about the human, rat and mouse genomes, and allows us to see the relationships between genes and chromosomes in two species at a time.

SyntenyVista focuses on the visualisation of gene comparisons. The tool shows relationships between genes, syntenic groups, chromosomes and QTLs. It has features which make it more usable than other existing genome browsers. SyntenyVista shows the whole chromosome with detail and supports choosing the part which will be investigated. The view uses colour and chromosome numbering to support understanding at the starting point of the visualisation. The users can manipulate the view by using both mouse and keyboard interaction. The application (SV1) offers the option to invert the chromosomes, which was found to be useful. It offers smooth zooming which supports the visual exploration of the chromosome space. The users can keep an area of interest in focus during the zooming process. The developers have also enabled panning. The users can move the chromosome with the mouse on the gene panel, or drag the box enclosing the region of interest. The display of the genes can be scaled by using a mouse action. The second version of SyntenyVista (SV2) has the cartoon scaling feature.

SyntenyVista includes also a top panel allowing additional user interaction and presenting information. The panel displays information on genes or QTLs in response to mouse movement in the gene area. QTLs are displayed as thin lines along the chromosome axes. The panel offers options to search for a gene name or a chromosome position. A gene is then highlighted on the whole chromosome image and the gene and its counterpart in the other species blink for a few seconds.

DerBrowser

DerBrowser [2], see Fig.2E, was designed at the time of the human genome sequencing project. It is a Java applet which supports interactive visualisation of one chromosome, or of a chromosome part. It can use a local database to produce web pages showing all the information describing a given map object.

It can be used to display genes, chromosome bands (chromosome parts coloured light or dark in the karyotype pictures), markers, and can represent any object on a map [15]. It provides an illusion of smooth zooming (a slider), and supports the hiding of objects, based on object type. It can also perform search functions.

Apollo

Apollo [3], see Fig.4, is a sequence annotation viewer and editor. It allows the biologist to improve on the genomic feature descriptions derived from automated analyses and computational pipelines. It facilitates connecting to various databases and the comparison

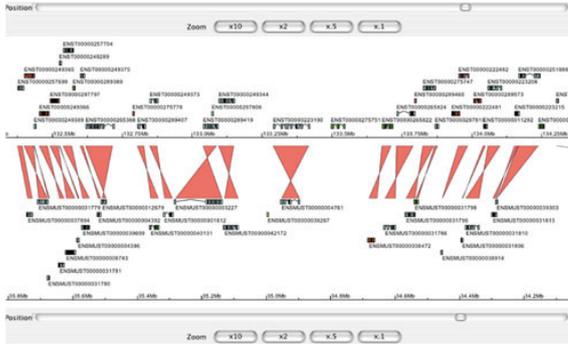


Figure 4. Chromosome comparison represented in Apollo. We can see human chromosome 20 at the top and part of mouse chromosome 2 at the bottom.

of existing annotations with other biological data. The tool offers researchers the ability to probe, manipulate and alter the interpretation of the underlying data. Within the various views offered by the package, annotations can be created, deleted, merged, split, classified and commented upon. The tool allows the view to be scaled using zoom buttons and provides a degree of semantic zooming. Some features are not displayed at low zoom levels and appear more precisely only when the user zooms in on them. The users can move to a specific position by specifying a coordinate, gene name, or short sequence string, or by using the horizontal scroll bar. Apollo can display features on two genomes at the same time. The view offers zooming and panning but it still does not present the data clearly, and the users cannot see all the relevant details.

Artemis

Artemis [4], see Fig. 2G, is a genome viewer and annotation tool that visualises sequence features and the results of analyses within the context of the sequence, and its translation from DNA to protein. Artemis can be used as a sequence viewer and is suitable for smaller genomes. Properties of the sequence can be plotted. Each plot allows dynamic modification of the window size used for the calculation. The sequence and plots can be zoomed together into the single base level or out for the complete genome. Artemis provides two sequence windows to view the same sequence at different zoom levels simultaneously. The tool can be run as an applet within a web browser.

BugView

BugView [5], see Fig.2F, is a comparative genome browser. It allows one to compare the arrangement of genes in two genomes, and can also be used to view individual genomes. It was written to enable comparative study of bacteria, including the comparison of bacterial strains.

The view presented by BugView is restricted to the genes, showing gene overlaps, and, where relevant, intron-exon structure, including alternative splicing. The users can scroll and zoom smoothly, and search for gene names. BugView includes support for the comparison of genes (sequence analysis) and analysis of gene alignments and other sequence features. For instance,

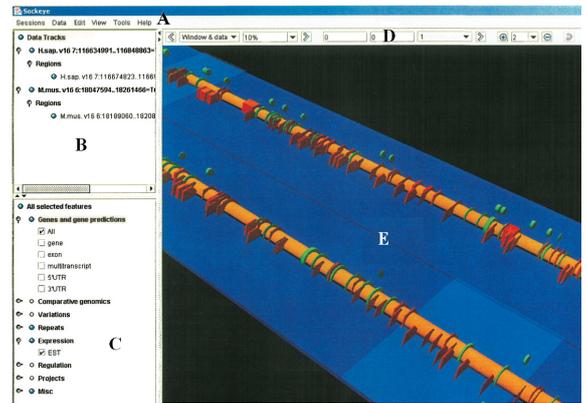


Figure 6. Sockeye chromosome visualization in 3D. We see the menu (A), the sequence track selection tree (B), the feature selection tree (C), the navigation toolbar (D), and the 3D viewport (E). The application allows the users to show/hide and obtain detailed information for loaded sequence track annotation types. In 3D viewport the users can perform analysis and annotations.

one can filter sequence alignments by specifying percentage similarity.

Ensembl

Ensembl [6], see Fig. 5, is probably one of the most popular systems for genome analysis. Ensembl database organizes biological information around the sequences of large genomes. It is an interactive Web site, a set of downloadable flat files, and a complete, portable open source software system for handling genomes. The Ensembl browser displays assembled sequences, cross-species synteny, genes, transcripts, proteins, supporting evidence, dot-plots, protein domains and gene/protein families.

The users can find 17 different views for data offered by Ensembl such as: AlignView, AnchorView, ChromoView, ContigView, CytoView, DomainView, ExonView, FastaView, GeneView, KaryoView, MapView, MarkerView, MultiContigView, ProteinView, SNPView, SyntenyView, and TransView. Different views are used to represent different kind of data. In our experiment, a number of genomic data was represented by ContigView, MultiContigView and SyntenyView. In SyntenyView a diagram of chromosomes with blocks of conserved synteny and homology matches between individual genes with syntenic blocks are shown. In ContigView, Fig. 5, a set of different views of a gene is shown, from broad chromosome context to fine nucleotide detail. These views are in separate horizontal frames, one below the other. The data presented in Ensembl is supported by labelling and searching. MultiContigView is an extension of ContigView. It allows display of genome annotation for several species. We find that because of the size of the data set, it is difficult to show all requested details on one screen. The users need to use scrolling and very often get lost in the information space.

Sockeye

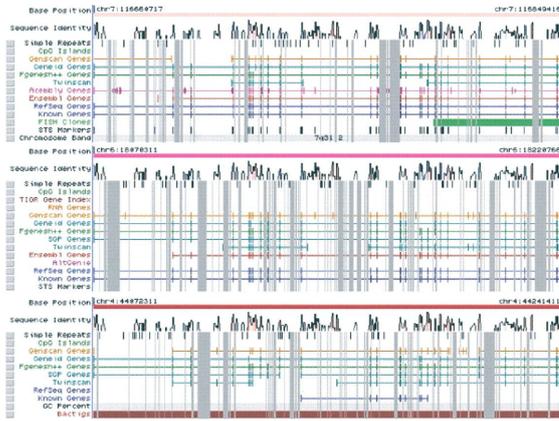


Figure 7. K-BROWSER showing the cystic fibrosis gene region (CFTR). Human, mouse and rat annotations are presented (from the top to the bottom panel). The grey bars indicate gaps (arising from insertions or deletions) in the sequences. The user can navigate using zoom buttons, gene name searching, and position jumping.

Sockeye [9], Fig. 6, uses 3D graphics and data from the Ensembl database project. A user can also import custom sequences and annotation data. Large sets of functionally linked sequences containing genes that are coexpressed, and orthologous across multiple species, can be analysed. The difference between Sockeye and other existing browsers is in the 3D environment. Each 3D model is specified in a user configurable XML format file. Sockeye integrates the process of obtaining sequence and annotation data. The application also allows a user to simultaneously visualise several different alignments and easily view their gaps. Montgomery et al. [9] stress that the 3D environment has a lot of advantages and disadvantages but only a few researchers decided to use it in their work. 3D visualisation is uncommon in genomics and researchers find it difficult to use. The developers find Sockeye to be user-friendly, but the users cannot easily see all interesting objects. The interface shows the sequence track selection tree, the feature selection tree, several navigation controls, and the 3D viewport. The users can also compare the extensive information contained across multiple genomics sequences, and zoom, pan and rotate the position of the sequence track.

K-BROWSER

K-BROWSER [10], see Fig. 7, is a comparative browser which visualises biological information at a higher level of resolution than is the case in most other tools. Its novelty is the representation of sequence similarity histograms along sequence features on several genomes. K-BROWSER was built on the foundation of the UCSC Genome Browser [24]. It can display a number of genomes overlaid with annotations and predictions, and shows the multiple alignments that describe global sequence relationships.

K-BROWSER takes as input a specific region in a genome and produces a set of images that succinctly represent the requested region and all orthologous regions in other genomes. The two critical components of

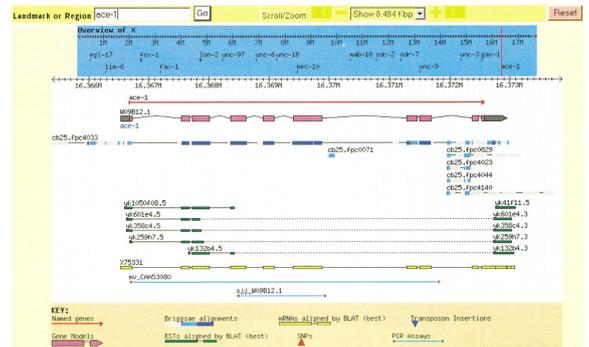


Figure 8. GBrowse. The users can type a landmark name into the text field at top. Landmarks can be gene names, clone names, accession numbers, or any other identifier configured by the administrator. Once a region is selected, it is displayed in a detailed view that summarizes annotations and other genomic features.

the application are track realignment and image generation. Track realignment is responsible for the necessary scaling of DNA lengths in the comparative views to make it consistent with the multiple alignment. Image generation takes as input a genomic region query and produces an image for every corresponding region in the multiple alignment. The tool displays a sequence conservation plot above the tracks. It allows the users to select a track according to which the conservation plot is to be coloured. The K-BROWSER can compute the percentage similarity between the root sequence and the leaf sequence in a window centred on a specified position. It allows one not only to determine if a genomic region is conserved within other genomes, but also to infer the rate at which it is evolving.

GBrowse

The Generic Genome Browser [11], see Fig. 8, is a combination of a database and interactive web pages for the manipulation and display of genome annotations. GBrowse can display an arbitrary set of features on a nucleotide or protein sequence, and can accommodate genome-scale sequences. GBrowse provides most of the features available in other browsers but was designed from the outset to be portable and extensible. It provides multiple configurable levels of zoom and two scroll speeds, and it also offers semantic zooming. The users can customize the view, the track, and the width of the image. The application allows for adding annotations to the genome. GBrowse supports also a plug-in architecture that allows third party modules to extend it.

NCBIMapViewer

The NCBI Map Viewer [12] is a Web interface used to view and search an organism's complete genome. The users can also view maps of individual chromosomes and zoom into specific regions within chromosomes to explore the genome at the sequence level. They have access to several different types of maps for different organisms. Map Viewer allows the user to view these maps graphically or in a table format. NCBI Map Viewer's graphic display is limited to features related

Genome browsers	Technology
AceDB	initially in C, later connectivity via Perl, Java or CORBA [18]
SyntenyVista	Java - Piccolo [13] and Swing [27]
DerBrowser	Java 1.02, java applet
Apollo	Java 1.2 or 1.3
Artemis	Java Application, but can be run as an applet
BugView	Java 1.1, java applet
Ensembl	MySQL [31], Perl API, and Java API, images are generated dynamically using Ensembl drawing code [7]
Sockeye	standalone application in Java, using JDK 1.4.x and Java 3D 1.3.x
UCSC GenomeBrowser	MySQL, BLAST-like Alignment Tool (BLAT) [19]
K-BROWSER	image generation component borrowed from UCSC GenomeBrowser
GBrowse	MySQL, DAS, Perl, and Apache
NCBI	Entrez System
eQTL Explorer	Java
Expressionview	Perl script derived from the Ensembl program blastview

Table 1. Technologies used to implement genome browsers.

to gene identification, although there are text links to other pages. Zooming and other visualisation features are not as sophisticated, in our opinion, as those offered in Ensembl.

eQTL Explorer and Expressionview

eQTL Explorer [14] visualises QTL data on the background of each chromosome. The chromosomes are drawn as vertical bars, and the QTLs are shown as coloured triangles. The application can display individual chromosomes in a separate view, with options to browse, zoom and export data. The tool has also a pop-up menu which provides access to annotations and cross-references to external data sources. The tool represents only a small subset of genome data.

Expressionview [22], see Fig. 2C, and eQTL Explorer, which is similar in appearance, are two applications designed specifically for the analysis of micro array experiments. Both applications show entire karyotypes and draw QTLs and genes identified in a micro array experiment alongside the chromosomes. Both applications are single-purpose, in that they do not show other biologically relevant information at the same time, for instance all the genes or SNPs.

SUPPORTING TECHNOLOGIES

The genome browsers we discuss use different technologies which offer differing levels of support for visualisation and user interaction. The newer viewers, such as SyntenyVista [1], use Piccolo [13] which allows for smooth zooming and panning. Piccolo toolkit supports the development of 2D structured graphics programs. It implements a hierarchical structure of objects and cameras, allowing the developers to manipulate objects, and the users more options in the presentation of data. SyntenyVista also uses Swing [27], which is a GUI toolkit for Java. Swing graphical user interface offers text, boxes, buttons, split-panels, and tables. The technology allows the developer to add ready-made and sometimes complex components to an application. At the other end of the spectrum we have clickable graphics generated by a sever, such as Ensembl [7]. The devel-

opers define a clickable area for graphics, and then, after user interaction, images are generated.

Visualisation software often requires the user to modify her software environment. The users need to have a specific version of Java and adjust security settings if they want to use an application based on Piccolo or Swing. There is also a very important limitation because of available memory and CPU speed on the users' machine. Some genome browsers, especially the ones which use the newest technology, expect a lot of memory. There is a trade-off currently between portability and visualisation. Most portable viewers use simple, server-side technology, and offer little in terms of view adjustment. On the other hand, powerful browsers written in Java need better hardware and need to be set-up but offer improved data analysis support. Most of the browsers we describe connect to a database, while some rely on flat files. Ensembl and GBrowse, for instance, support the addition of new data sources via the Distributed Annotation Service (DAS) protocol [30]. DAS is an open source standard supporting the sharing of genomic annotations on the web.

TESTING

In cooperation with the biologist groups, we tested all the described genome browsers in order to find which one supports the interpretation of their experiments (see <http://www.dcs.gla.ac.uk/~asia/work.html>). We found that none of the browsers fully supports user requirements we identified. Using AceDB with Ensembl data is not feasible, and would make us inherit the limited zooming support offered by the AceDB maps. On the other hand, it would have been possible to add new data from the lab easily, and add new data types. We found that DerBrowser's functionality does not fulfill users' expectations. We expected that the user should be able to move around the data columns and to zoom smoothly and precisely. Beyond a certain point, we could not zoom in any further and we could not see the genes and micro array probes in detail. We could not compare two genomes either. It was also impossible to add new features because of the old version of Java the application uses.

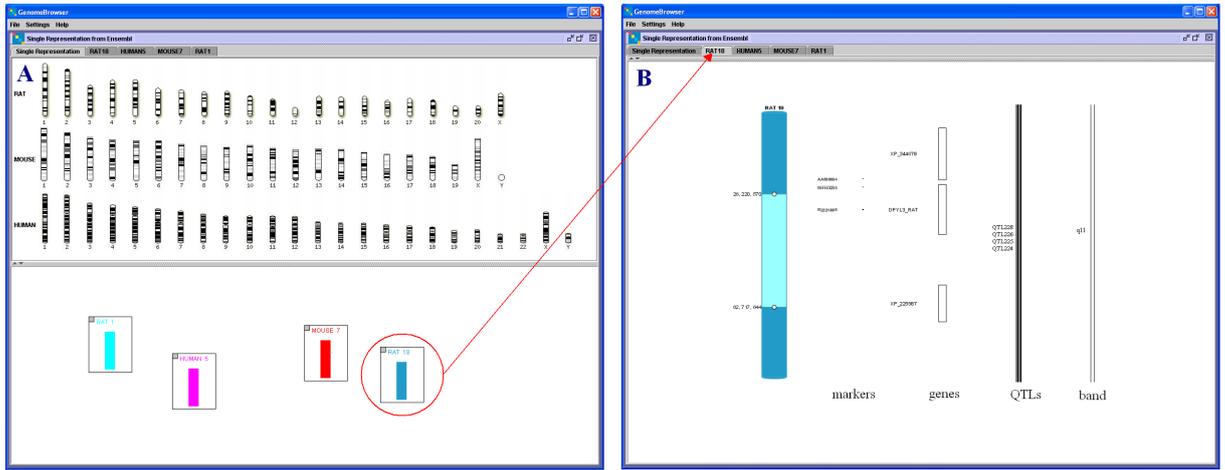


Figure 9. VisGenome offers two new views. View A shows chromosomes from three species (the mouse, the rat and the human) in the upper part of the display and the chromosomes for which data has been retrieved from Ensembl in the lower part. View B contains an overview and detail for the rat chromosome 18.

We found that Ensembl is not appropriate for our users, as it does not support the comparison of QTLs and their gene content, due to the limited flexibility of view manipulation. The same is true for the NCBI Entrez system, Gbrowse and K-BROWSER. We also looked at the maps offered by the Rat Genome Database [8] and saw that the data required by the biologists was not shown. SyntenyVista solves only some of the visualisation problems. It supports the comparison of two genomes, but it does not display all the relevant data (micro array probes, SNPs, markers, etc.). We experimented also with Apollo's user interface which is meant to be intuitive. We found it harder to use than other interfaces, and found that the display was not clear and zooming was not satisfactory. We also found that BugView was easy to use, but, unfortunately, shows only a subset of data that the biologists want to visualise. This is similar to the situation we encountered in SyntenyVista.

eQTL explorer and Expressionview show only a subset of data, and the users cannot compare known genes or SNPs with data represented by the visualisation. Completely different are Ensembl and NCBI Map Viewer, which read data directly from a huge database and show as much data as it is possible. The users easily get lost in such interfaces, as the data is shown in several screens which do not fit simultaneously on the computer screen. This limitation is the result of the lack of support for image manipulation within web browsers. The images shown by Ensembl and NCBI never fit on one screen and we found that disorientating. In Ensembl the display of synteny, see Fig. 2B, does not present much detail and can not be used for micro array data analysis. The MultiContigView is much less legible than SyntenyVista, and does not offer smooth navigation. We also examined Sockeye and found the 3D view to be confusing. This was mostly due to poor labelling and possibly visual occlusion.

DESIGN OF VISGENOME

We developed a new version of SyntenyVista, VisGenome. The software extends SyntenyVista with new features, and allows for the addition of new data ty-

pes to the display, and will be able to satisfy user requirements fully. The data are presented vertically. The application loads the data from Ensembl. It welcomes the user with a view of all rat, mouse and human chromosomes. Then, after choosing a chromosome of interest, the user sees it in the bottom window. After the user selects the chromosome by clicking on it, a new view with detailed data about the chromosome is created. This solution allows the users to see in one place what data was downloaded from Ensembl as well as the detailed information on the chromosome, including bands, markers, QTLs and genes. After choosing a chromosome the users can manipulate the view by mouse and keyboard interaction. We offer smooth zooming which supports the visual exploration of the chromosome space. The users, in the same way as in SyntenyVista, can keep an area of interest in focus during the zooming process. We implemented zooming and panning using Piccolo [13]. The users can choose the chromosome region of the interest by dragging the box enclosing the region or typing in the coordinates in the top info panel. Then only the data in the selected area is displayed. The solution allows us to keep the context, the users can navigate the data and all the time they know exactly in which region of the chromosome the data is situated.

The new genome browser, see Fig. 9, shows bands, markers, QTLs and genes in a single representation. It can show any data types specified by a query sent to the Ensembl database. We are currently adding the display of SNPs, and will also add haplotype blocks and protein expression results, and allow the user to adjust the display to suit their information needs.

USER TEST

We will carry out a user test with 10 users, in two settings, cardiovascular research and schizophrenia. The users will be performing the following tasks. First they will read in the data from their latest micro array experiment, stored in a spreadsheet. The visualisation system will show an overview of chromosomes to which the new results relate, similar to that seen in Express-

sionview (Fig. 2C), where both the QTLs and differentially expressed genes will be shown superimposed on a karyotype picture. Then the biologists will select the longest of the QTLs in which they are interested, and verify that they can see the QTL, the genes, and the micro array results. Micro array results will be coloured in two colours, one for genes showing increased expression, the other for the genes showing reduced expression. The view will also display results imported from another micro array experiment, from external published data selected by the biologist, for comparison.

The following will be measured: total time required to perform the visual assessment of the new experiment; time to examine one QTL in detail; and number of mouse and keyboard actions executed. Additionally, a survey will be used to get user impressions on the legibility of the display, aesthetic appeal, and the subjective ease of use.

CONCLUSION

Visualisation of genome comparisons is an important research tool in biology and medicine. There are a variety of genome browsers which in practice should perform the same function - show the chromosomes of some species in detail. The differences in the view and also in functionality of the tools for genome browsing motivated us to create a classification of genome browsers.

Our future plans include more experiments with the users to check which of the tools' properties are welcome and which are less user-friendly. We would like to test not only VisGenome but also different tools, with biologist groups we cooperate with, to find the most intuitive visualisation technique. We are going to continue our work with VisGenome, which shows genome data at different level of details. We believe, that biologists still require new methods to visualise genomic data.

ACKNOWLEDGMENTS

We thank Prof. A. Dominiczak, BHF Glasgow Cardiovascular Research Centre and the Wellcome Trust Cardiovascular Functional Genomics Consortium for their collaboration and Prof. K. Dittrich at the University of Zurich for hospitality.

REFERENCES

- Hunt, E. et al. The visual language of synten. *OMICS* 8(4), (2004), 289-305.
- Leser, U. et al. IXDB, an X chromosome integrated database. *NAR* 26(1), (1997), 108-111.
- Lewis, S. E. Apollo: a sequence annotation editor. *Genome Biology*, (2002).
- Rutherford, K. et al. Artemis: sequence visualization and annotation. *Bioinformatics* 16(10), (2000), 944-945.
- Leader, D. P. BugView: a browser for comparing genomes. *Bioinformatics* 20, (2004), 129-130.
- Ensembl database. <http://www.ensembl.org>.
- Hubbard, T. et al. Ensembl 2005. *Nucleic Acids Res.* 2005 Jan 1;33 Database issue:D447-D453.
- Rat Genome Database (RGD). <http://rgd.mcw.edu>.
- Montgomery, S. B. et al. Sockeye: A 3D Environment for Comparative Genomics. *Submitted Genome Research*, (2003).
- Chakrabarti, K. and Pachter, L. Visualization of multiple genome annotations and alignments with the K-BROWSER. *Genome Research*, (2004).
- Stein, L. D. et al. The genetic genome browser: a building block for a model organism system database. *Genome Research* 12(10), (2002).
- Online Mendelian Inheritance in Man. <http://www.ncbi.nlm.nih.gov/>.
- Piccolo Toolkit. <http://www.cs.umd.edu/hcil/piccolo/>.
- Mueller, M. et al. eQTL Explorer: integrated mining of combined genetic linkage and experiments. *Bioinformatics* 22(4), (2006), 509-511.
- The human chromosome 21 database. <http://chr21.molgen.mpg.de/>.
- NCBI Entrez. <http://www.ncbi.nih.gov/Entrez>.
- British Heart Foundation Blood Pressure Group. <http://www.medther.gla.ac.uk/bhf/index.htm>.
- CORBA. <http://www.corba.com/>.
- Kent, W. J. BLATthe BLAST-like alignment tool. *Genome Res.*, 12, (2002), 656-664.
- Cinema. <http://umber.sbs.man.ac.uk/dbbrowser/CINEMA2.1/>.
- Hubner, N. et al. Integrated transcriptional profiling and linkage analysis for identification of genes underlying disease. *Nat. Genet.*, 37, (2005), 243-253.
- Fischer, G. et al. Expressionview: visualization of quantitative trait loci and gene-expression data in Ensembl. *Genome Biology*, 4, (2003).
- MGI. <http://www.informatics.jax.org>.
- Karolchik, D. et al. The UCSC Genome Browser Database. *Nucleic Acids Res.* 31: , (2003), 51-54.
- Jmol. <http://jmol.sourceforge.net>.
- Medical Research Council. <http://www.mrc.ac.uk>.
- Swing. <http://java.sun.com/products/jfc/>.
- Page, R. D. M. TREEVIEW: An application to display phylogenetic trees on personal computers. *Computer Applications in the Biosciences* 12, (1996), 357-358.
- Durbin, R. and Mieg, J. T. A C. elegans Database. Documentation, code and data available from anonymous FTP servers at lirmm.lirmm.fr, cele.mrc-lmb.cam.ac.uk and ncbi.nlm.nih.gov, (1991-).
- Dowell, R. D. et al. The distributed annotation system. *BMC Bioinformatics*, 2:7, (2001).
- MySQL. <http://www.mysql.com>.
- Metabolic Pathways. <http://www.lirmm.fr/~fjourdan/mainE.html>.

G: VisGenome User Manual, under construction

<http://www.dcs.gla.ac.uk/~asia/VisGenome/VisGenomeManual.pdf>

Joanna Jakubowska and Ela Hunt
asia@dcs.gla.ac.uk, elahunt@inf.ethz.ch

1 Installation

The easiest way to start VisGenome is by selecting one of the two WebStart buttons at <http://www.dcs.gla.ac.uk/~asia/VisGenome/>, described in Section 1.1. Alternatively, one can download the files, as described in Section 1.2.

1.1 WebStart

At www.dcs.gla.ac.uk/~asia/VisGenome you can find Java Web Start versions of VisGenome for file and database versions. To run it just click on the link at the web page.

1.2 Installation

1. You can find the application jar at www.dcs.gla.ac.uk/~asia/VisGenome or www.dcs.gla.ac.uk/~asia/VisGenome/index_files/jars.

At www.dcs.gla.ac.uk/~asia/VisGenome/index_files/jars you find three files: `VisGenome_DatabaseVersion_1.5_OneJar.jar`, `VisGenome_FileVersion_1.5.jar`, and `data.zip`. These files correspond to two versions, and a file containing all data for the MM, HS and RN that VisGenome uses. We recommend the use of `VisGenome_DatabaseVersion_1.5_OneJar.jar` which connects to Ensembl and takes data from the database. For impatient users we recommend `VisGenome_FileVersion_1.5.jar` which has downloaded data from Ensembl v. 42 and is faster than the database version of VisGenome. If you prefer file version of VisGenome you have to remember to download `data.zip` and unpack it in the same directory as the jar.

2. You should save the file and make sure that the file is saved as `VisGenome_DatabaseVersion_1.5_OneJar.jar` (`VisGenome_FileVersion_1.5.jar`), and not `VisGenome_DatabaseVersion_1.5_OneJar.zip` (`VisGenome_FileVersion_1.5.zip`). If you saved it as `VisGenome_DatabaseVersion_1.5_OneJar.zip` (`VisGenome_FileVersion_1.5.zip`), please rename it to `VisGenome_DatabaseVersion_1.5_OneJar.jar` (`VisGenome_FileVersion_1.5.jar`), for instance at command prompt by executing `copy VisGenome_DatabaseVersion_1.5_OneJar.zip VisGenome_DatabaseVersion_1.5_OneJar.jar`.

1.3 Application startup

Click on `VisGenome_DatabaseVersion_1.5_OneJar.jar` (`VisGenome_FileVersion_1.5.jar`) or go to the directory where you saved `VisGenome_DatabaseVersion_1.5_OneJar.jar` (`VisGenome_FileVersion_1.5.jar`) and execute the command `java -jar VisGenome_DatabaseVersion_1.5_OneJar.jar` or the command `java -jar -Xms256m -Xmx256m VisGenome_DatabaseVersion_1.5_OneJar.jar`.

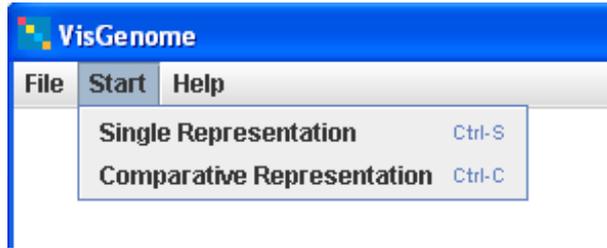


Figure 1: VisGenome, menu, representation choice: single or comparative.

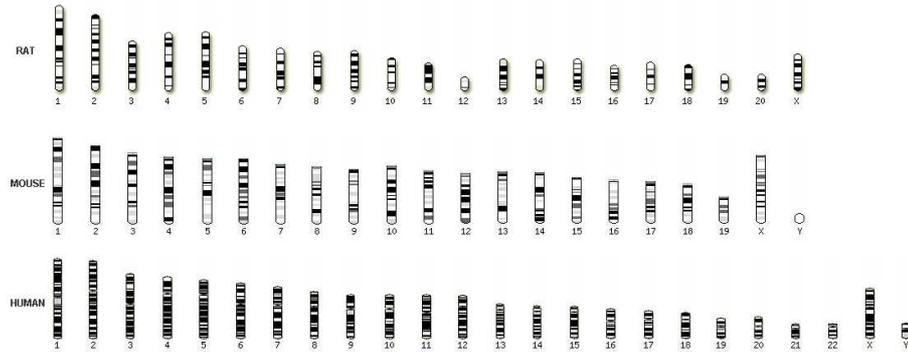


Figure 2: VisGenome, Single Representation, chromosomes appear when you select the option Single Representation.

`Jar.jar` (for file version appropriately: `java -jar VisGenome_FileVersion_1.5.jar` or the command `java -jar -Xms256m -Xmx256m VisGenome_FileVersion_1.5.jar`). The directives `-Xms256m -Xmx256m` depend on your machine and reserve more memory.

To run Java Web Start versions of VisGenome for file or database versions just click on the link at the web page.

VisGenome opens a browser which allows you to choose a Single or a Comparative Representation. Please select one of those options, see Figure 1. Alternatively, use `Ctrl+S` for Single Representation or `Ctrl+C` for Comparative Representation. This invokes a connection to Ensembl and may take some time. You can see the User Manual by choosing *Help* in VisGenome menu, see Figure 1.

2 Single Representation

In the Single Representation choose the chromosome by clicking on the karyotype picture in the main panel, see Figure 2. Then you see a progress bar and a new view with details for the chosen chromosome. You can select as many chromosomes as you want, subject to memory restrictions on your machine. For each selected chromosome a separate tab with the data is created. Figure 4 shows tabs *Single Representation*, *Rat18*, *Rat11*, and *MouseY*. You can switch between the tabs.

Each tab shows the main view with detailed data about the chosen chromosome (the chromosome with bands, Affy Probe Sets, Genes, Markers and QTLs) and at the top - Further Info,

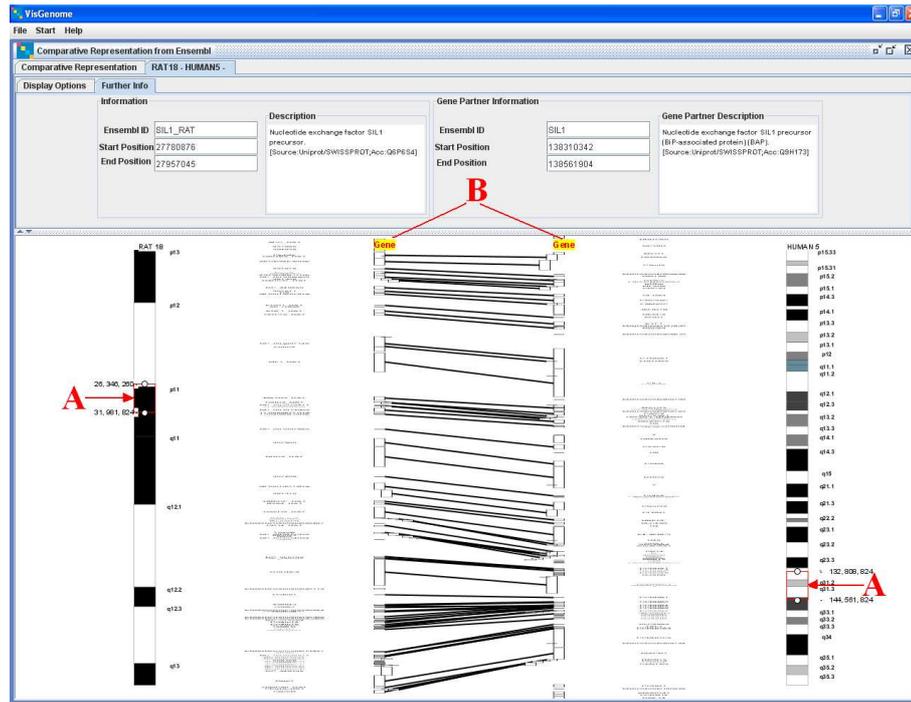


Figure 5: VisGenome, Comparative Representation, homologies between the rat chromosome 18 and the human chromosome 5. A points to red squares marking the chromosome regions for which the data is shown. B points to column labels.

see Figure 4 and 3. In Single Representation in Further Info you can enter the coordinates for start and end of the chosen chromosome region (then you have to press *Set Region Position* to see the results). When you move the mouse in the main view and scroll along the genes you can see additional information in the information panel, such as EnsemblID, Start Position, End Position, and Description. The information is about the gene which the mouse is positioned over.

Labelling. You can switch the labelling mode for Affymetrix Probe Sets and Markers by *Toggle* in Labelling section. Two labelling modes are provided: either all labels are shown, or only the label for the topmost object is displayed. Try using the 'Toggle' button to experiment with the labeling option for microarray probes and markers, to see which suits you best. When you move the mouse in the main view and scroll along the Affy Probe Sets or Markers, you see that a line connecting a selected object and one of its labels is highlighted, or, when you toggle, the lines disappear and only some names are visible.

3 Comparative Representation

In the Comparative Representation choose two chromosomes from two different species by clicking on them. Similar to Single Representation, you can choose as many pairs of chromosomes as you want, subject to memory restrictions on your machine. Each view in the Comparative Representation offers two tabs with additional info. You can set the range of chromosomes displayed in the tab *Display Options*, see Figure 6, or you can see additional information about genes in the tab *Further Info*, see Figure 7.



Figure 6: VisGenome, Comparative Representation, Display Options Panel which enables setting the chromosome area for viewing.

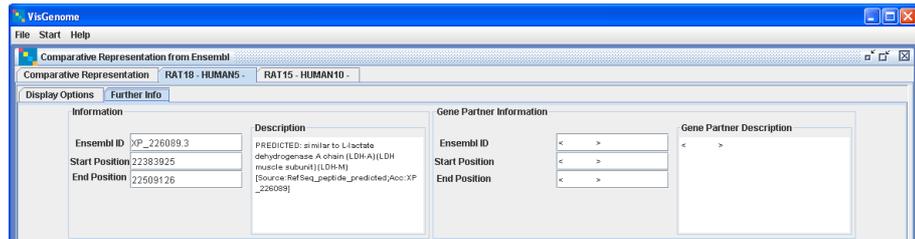


Figure 7: VisGenome, Comparative Representation, Panel with Further Info, showing additional information about homologous genes. If a gene has a homologue, both *Information* and *Gene Partner Information* are filled in. If it has no partner, information appears in one of the boxes, depending on the mouse position in the comparative view.

4 Zooming, panning and access to Ensembl pages

4.1 Zooming

Zooming is manipulated by the **right mouse button**, by positioning the mouse pointer over the genes or other objects which are to be manipulated, and dragging the mouse with the button pressed down to the right (zoom in) or to the left (zoom out). This zooming technique allows you to keep an area of interest in focus during interaction with the data, see Figure 8.

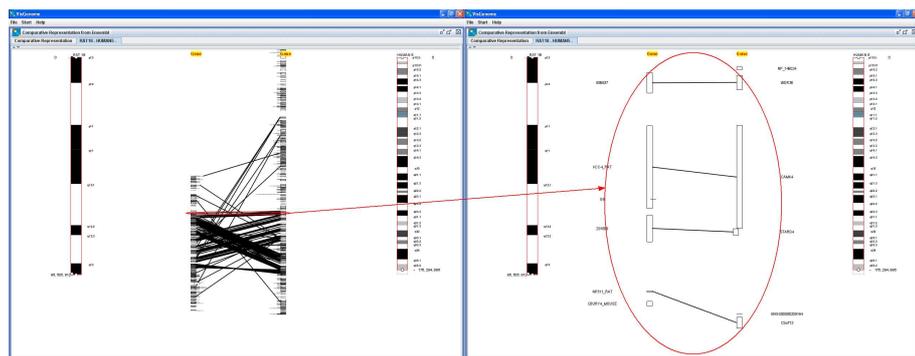


Figure 8: VisGenome: the same area of the presented data zoomed out (left part of the figure) and zoomed in (right part of the figure).

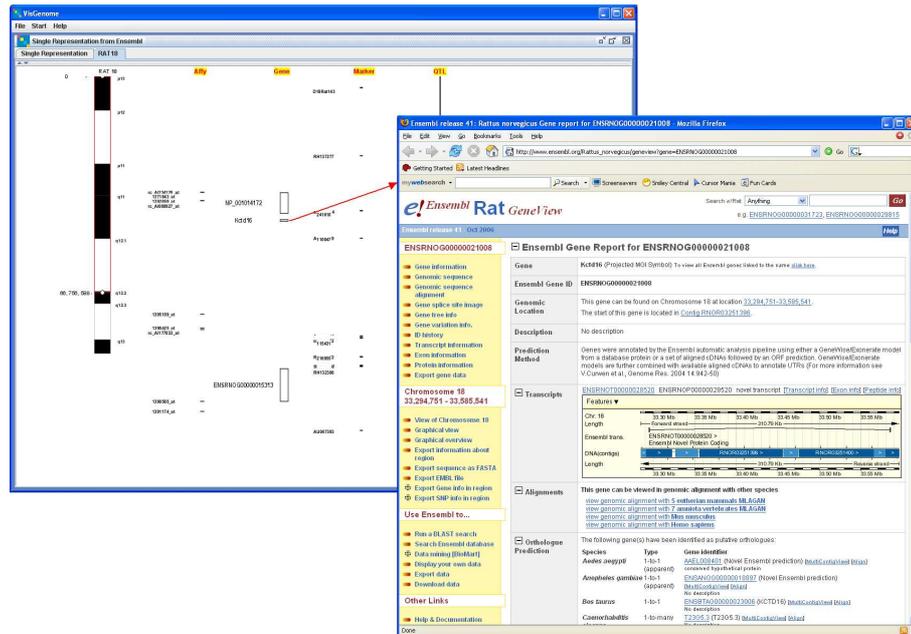


Figure 9: VisGenome linking to a selected gene in Ensembl.

4.2 Panning

VisGenome offers panning. To pan, please position the mouse pointer over the karyotype image or over the genes and press the **left mouse button** while moving the mouse up or down. You can also extend a single boundary of the box on the karyotype picture and reposition it by pressing the left mouse button and pulling the boundary up or down.

4.3 Invocation of Ensembl web pages

VisGenome can query supporting data from Ensembl. To see the web page for a selected gene, please click the left mouse button (this is changed in the second version of VisGenome where you should press Shift and click) on a gene of interest in VisGenome and an Ensembl web page will appear, see Figure 9. In this release the functionality is implemented only for genes in both Single and Comparative Representations.

4.4 The red square

Both Single and Comparative Representations offer the red square which allows the users to mark the area of the interest, see Figure 3A and Figure 5A. You can move the red square along the chromosome and make it larger or smaller either by mouse manipulation (click at the top or bottom of the square and stretch it) or by entering the coordinates in the info panel using the option *Region Range*). Only data for the red square is displayed, see Figure 3 or Figure 5. This means that when you zoom or pan in the main view you see all or some of the data from the red square. If you want to see more, you have to mark a larger region. Figure 5 shows all data from the red square, zoomed out. Figure 3 shows only part of the data from the red square, but by zooming or panning you get to see all data marked by the red square. You will not see data outside the coordinates marked by the square.

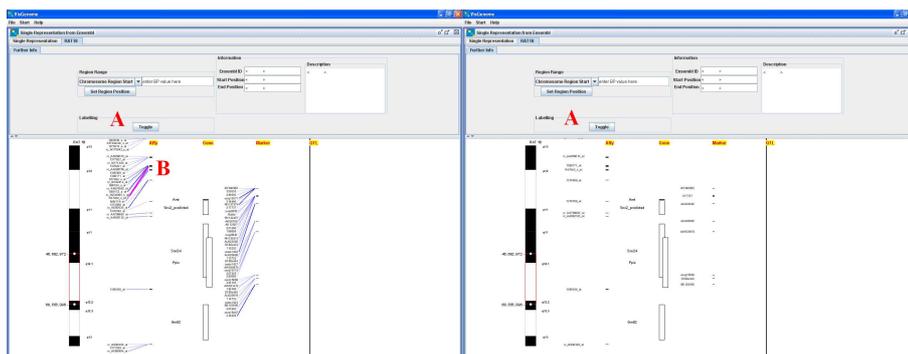


Figure 10: VisGenome: the same area of the presented data with all available labels for Affy Probe Sets and Markers (left part of the figure) and selected labels for Affy Probe Sets and Markers (right part of the figure). A points to button which switches the mode between selected labels and the all labels. When all labels appear, they have blue links which connected an element with a label. As well when an user move through the elements he sees that the line with a label is highlighted (B).

4.5 Labelling

Toggle button in Labelling section in additional info for Single Representation is responsible for appropriately displaying labels for Affy Probe Sets and Markers. After choosing Single Representation a user sees a view with all labels for Affy Probe Sets and Markers, see Figure 10 - left view. In this mode one can move through the columns with Affy Probe Sets or Markers and then a line connecting an element with its label (only one label is selected by our algorithm) is highlighted. In this mode all labels are displayed. It means that if a number of elements have the same coordinates, for each of them a label is displayed. The *Toggle* button allows you to switch mode. After pressing it you see only selected labels for Affy Probe Sets and Markers. A label is situated exactly opposite an element it describes. In this mode some of the elements are without labels, depending on the number of elements is in the neighborhood. In this mode as at most one label is displayed for each visible element. The solution allows us to display labels without overlaps, and increases legibility.

5 Orthologue predictions

The homologies presented in VisGenome are orthologue predictions. We retrieve from Ensembl all orthologue predictions, including `ortholog_one2one`, `apparent_ortholog_one2one`, `ortholog_many2many`.

After choosing one chromosome in the Comparative Representation you see chromosomes which have homologies with the chosen chromosome, see Figure 11.

6 Focus on

Both Single and Comparative Representations offer focus on to make the focal element large enough so that its name can be read. Focus on moves the element to the centre of the view, and

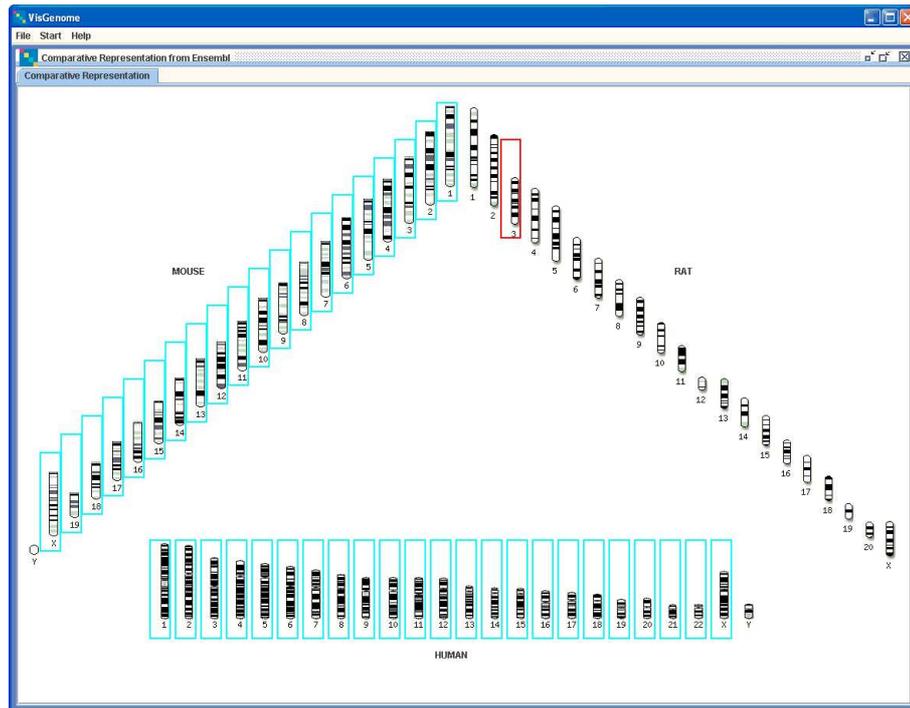


Figure 11: The rat chromosome 3 is chosen and other chromosomes which have homology with the rat chromosome 3 are surrounded by blue boxes.

marks its boundaries in red. To focus on, please click the left mouse button on an element (gene, marker, QTL, or microarray probe) of interest in VisGenome. In a Single Representation all neighbouring elements in the view become proportionally larger in all columns. In a Comparative Representation only elements in the chromosome containing the chosen element are changed, and all elements on the other chromosome maintain the same size.

7 Colour

You change colour for each of the elements by clicking on the object while pressing Alt. The default colour choice view is displayed and you can change the colour of the marked element.

8 Scaling

Scaling button in Scaling section in additional info for Single Representation is responsible for scaling all data. After choosing Single Representation you see a view with all data in original size, see Figure 12 - left view. The *Scaling* button allows you to switch mode. After pressing it you see all data scaled in relationship to genes. All genes are the same size and all other data size is modified, see Figure 12 - right view.

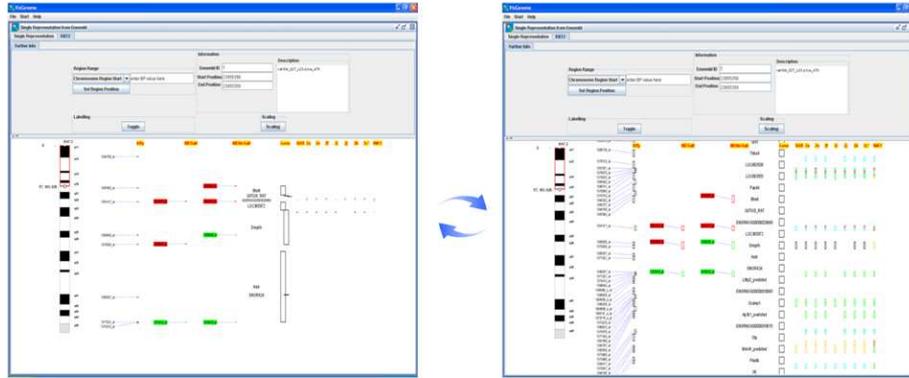


Figure 12: The data is scaled by the scaling algorithm which makes all genes the same size and other data size depends on genes.

9 Known problems

The biggest known problem is the speed of access to data from Ensembl. The application times out after 20 minutes if data was not fetched. This is true of the database version. To remedy this, we also offer the file version which uses the data.zip data file. An improved solution is being developed.

To increase the memory assigned to the application you may try starting VisGenome with larger parameters for `-Xms` and `-Xmx`. For instance, if your machine has 2 GB RAM, you can probably reserve most of that memory for VisGenome by invoking the application via: `java -Xms100m -Xmx1800m -jar VisGenome.jar`.

Some Mac machines do not like the database versions for one jar for Java 1.4 and Java 1.5. Therefore if you have a problem with running `VisGenome_DatabaseVersion_OneJar.jar` at your machine you have to unjar it and run from command line as: `java -cp main/VisGenome_DatabaseVersion.jar:lib/mysql-connector-java-3.1.8-bin.jar UserInterface.VisGenome`.

10 Programmatic reconfiguration to use other databases

All settings responsible for database configuration are in the class `GB_SR_DataEnsembl`. Further advice is available from the authors.

11 User study

We conducted a user study comparing VisGenome to Ensembl. A full account of the study is available as a technical report (http://www.dcs.gla.ac.uk/publications/PAPERS/8510/VG&Ens_TechRep.pdf) "Usability of VisGenome and Ensembl - A User Study". The user study is also published in DILS'08 proceedings - "VisGenome and Ensembl: Usability of Integrated Genome Maps". We also conducted second mixed paradigm user study which is in preparation - "Mixed Paradigm User Study".

H: Ethics Committee Form

1. DESCRIBE THE BASIC PURPOSES OF THE PROPOSED RESEARCH.
The proposed research aims to identify the best features in the most popular genome browsers. Between two and four genome browsers will be examined. At the beginning we will experiment with Ensembl and VisGenome tools. Participants – biological researchers who use such tools in their daily work - will be asked to use the browsers to visualise data from their experiments. Then an interview will be conducted with each of the participants about the usefulness of visualisation techniques used in it.
2. INDICATE WHO IS FUNDING THE RESEARCH (IF COMMERCIALY FUNDED, ENSURE THAT PARTICIPANTS ARE INFORMED).
PhD studentship awarded by the MRC to Dr Ela Hunt and Dr Matthew Chalmers of DCS. The work will be carried out by a PhD student Joanna Jakubowska.
3. DESCRIBE THE DESIGN OF YOUR EXPERIMENT (E.G. CONDITIONS, NUMBER OF PARTICIPANTS, PROCEDURE AND EQUIPMENT WHERE APPROPRIATE) see BPS §2 & §8.
A number of experiments will be carried out. The first one will use Ensembl and Vis Genome, and follow up experiments may involve other software tools.
First Experiment: Find the most user-friendly features in Ensembl and VisGenome
The experiment will examine the visualisation techniques used in genome browsers – Ensembl and VisGenome. User training will take 10 minutes, experiment itself 20 minutes and the interview 20 minutes, with 10 minutes left for any questions raised during the experiment. The experiment will take a maximum of 1h and 5 participants will take part. The experiment will tell us which visualisation techniques are useful in biological research. The participants will download data from files into the visualisation tool and carry out data interpretation
We will measure the following:
 - time required by the user
 - mouse clicks count
 - degree of task completionDuring the interview we will gather the following information.
 - the appropriateness of information representation (colours, font, layouts, and effects such as zooming, scrolling or panning)
 - the usefulness of the tools
 - ease of use**Further Experiments**
The above experiment will be repeated in the future for different genome browsers, and for an improved version of the tool we are developing (Vis Genome)
Procedure for all experiments
The procedure to be followed for all experiments is:
 - participants will be asked to read and sign a consent form
 - participants will undergo training on tool functionality
 - participants will be asked to download their data to the tools

- participants will be asked to find and show some specified data: a region of chromosome, gene or QTL
- participants will be asked to download the data from Ensembl for different species and compare with their data
- participants will be allowed to find an item of particular interest to their research
- the experiment will be recorded for future analysis (the software will be instrumented to measure mouse clicks and time between them)
- there will be interview with participants about their impression of the tool, if it was easy or difficult to use
- there will be short questionnaire about visualisation techniques used in the tools

4. DESCRIBE HOW THE PROCEDURES AFFECT THE PARTICIPANTS.

The participant will be performing the following actions:

- Reading and signing the consent form
- Reading biological data in specified genome browser
- Using mouse and keyboard to answer questions
- Identifying biological data used in the experiments

These activities present no risk, as they are daily performed by all potential participants.

5. STATE WHAT IN YOUR OPINION ARE THE ETHICAL ISSUES INVOLVED IN THE PROPOSAL (see BPS All Sections).

- Participants should understand that they themselves are not being tested.
- As a few of the participants may be students, they should be informed that their performance is not linked to their university marks.
- Participants should be informed that all data collected is collected in confidence, and is stored in an anonymised form.
- Participants should be informed that they may withdraw from the experiment at any time without prejudice, and that any data already recorded will be overlooked.

6. SPECIFY THE NATURE OF THE PARTICIPANTS. INDICATE IF THE RESEARCH INVOLVES CHILDREN OR THOSE WITH MENTAL DISABILITIES OR HANDICAP (see BPS §3) IF SO, EXPLAIN THE STEPS TAKEN TO OBTAIN PERMISSION FROM L.E.A.s, HEADTEACHERS, PARENTS, ETC...

No

7. STATE IF PAYMENT WILL BE MADE TO SUBJECT.

No

8. DESCRIBE THE PROCEDURES FOR ADVERTISING, FOR RECRUITING PARTICIPANTS, AND FOR OBTAINING CONSENT FROM PARTICIPANTS (see BPS §3).

A consent form, providing an overview of the experiment and the data that will be collected, its means of storage and the purposes of its use will be given to the participants. In addition, this sheet will also provide contact details of the experimenter, and give the participant the option to receive the summarised results of the data after it has been analysed.

9. STATE WHETHER THE PROPOSAL IS IN ACCORD WITH THE BPS CODE OF CONDUCT (see BPS All Sections).
Yes
10. DESCRIBE HOW THE PARTICIPANTS' ANONYMITY AND CONFIDENTIALITY WILL BE MAINTAINED (see BPS §7).
Each Participant will be allocated a participant number, however, no record of the mapping from participant number to participant will be retained.
11. DATE ON WHICH PROJECT WILL BEGIN.
August 2006
12. LOCATION AT WHICH THE PROJECT WILL BE CARRIED OUT.
Room F132
8-17 Lilybank Gardens
University of Glasgow
Glasgow G12 8QQ
13. DESCRIBE HOW PARTICIPANTS WILL BE DEBRIEFED AT THE END OF THE EXPERIMENT (THIS MUST INCLUDE THE OPPORTUNITY TO CONTACT THE EXPERIMENTER - OR SUPERVISOR - FOR FEEDBACK ON THE GENERAL OUTCOME OF THE EXPERIMENT) (see BPS §5 & §10).
Each participant will be verbally debriefed and asked about their opinions of the experiment, and any further comments they have. At this time participants will be reminded of the consent form and (if they have not already done so) can select the option for getting feedback from the experiment. They will also be reminded on how to contact the experimenter.
14. ATTACH PARTICIPANT INFORMATION FORM AND CONSENT FORM (see BPS § 3 & 6).
As part of the automated process, these files are required in order to proceed. If you do not have one of these files, submit an empty text file and sort the matter with the ethics committee.
(Please make sure that the files you are submitting can be readable by the members of the committee. Do not send compressed files as these are OS dependent. Best file formats are MSWord or plain text).

I: Participant Consent Form: Genome Visualisation

This experiment aims to identify the best features in two genome browsers – Ensembl and VisGenome. You will be asked to use the browsers to visualise the data from your experiments or some other publicly available data. Then you will be interviewed about the usefulness of visualisation techniques you tested.

Before the experiment you will be shown how the genome browsers work. In the experiment itself you will load or query data and see it visualised on the screen. You also will be asked to carry out a few (up to 5) short tasks involving the analysis of your data (searching and visualising some items).

After the experiment you will be asked to complete a short questionnaire to identify how useful are visualisation techniques were. You will be also asked for suggestions on how to develop further the tools.

During the experiment the computer will record usage data in the background (mouse clicks, mouse movement, and text typed in). The data will be stored anonymously using an ID number, rather than your name or any number that could identify you. All results will be held in strict confidence, ensuring full privacy of all participants. There will be no record kept that will allow your results to be tracked back to you, or used in any context other than this visualisation research. All data will be held securely in a password protected computer system.

The data will be analysed to identify which visualisation effects are poorly implemented in biological visualisation tools. The results of this analysis may be published in appropriate scientific journals and conferences.

A feedback sheet will be sent to all participants who request it, after the data has been analysed.

Your participation in this experiment will have no effect on your marks for any subject at this, or any other university, and the fact that you participated will not be known to anyone other than yourself and the experimenter.

You may withdraw from the experiment at anytime without prejudice, and any data already recorded will be overlooked.

If you have any further questions regarding this experiment, please contact:

Asia Jakubowska
Department of Computing Science
17 Lilybank Gardens
Glasgow G12 8QQ
e-mail: asia@dcs.gla.ac.uk
tel.: +44 141 330 4256 (ext. 0985)

I have read this information sheet, and agree to take part in this experiment:

Name: _____

Signature: _____

Date: _____

I would like to receive a summary sheet of the experimental findings

E-mail address: _____

J: Participant Consent Form: Genome Visualisation

This experiment aims to identify the best features in VisGenome and other tools used by biologists in their work. You will be asked to use the tools to visualise the data from your experiments or some other publicly available data for 2 weeks. During the time you use VisGenome, it will log all your activity in VisGenome. The person who conducts the experiment will spend with you about 2 hours in total observing your work and recording your activity. After the 2 weeks, you will be interviewed about the usefulness of visualisation techniques you used.

Before the experiment you will be shown how VisGenome works. In the experiment itself you will load or query data and see it visualised on the screen. You also will be asked to use VisGenome at least 5 times during the 2 weeks period.

After the experiment you will be asked to complete a short questionnaire to identify how useful visualisation techniques were. You will also be asked for suggestions on how to develop further the tools.

During the experiment VisGenome will log your activity in the tool (mouse clicks, mouse movement, and text typed in). The data will be stored anonymously using an ID number, rather than your name or any number that could identify you. All results will be held in strict confidence, ensuring full privacy of all participants. There will be no record kept that will allow your results to be tracked back to you, or used in any context other than this visualisation research. All data will be held securely in a password protected computer system.

The data will be analysed to identify which visualisation effects are poorly implemented in biological visualisation tools. The results of this analysis may be published in appropriate scientific journals and conferences.

A feedback sheet will be sent to all participants who request it, after the data has been analysed.

Your participation in this experiment will have no effect on your marks for any subject at this, or any other university, and the fact that you participated will not be known to anyone other than yourself and the experimenter.

You may withdraw from the experiment at anytime without prejudice, and any data already recorded will be overlooked.

If you have any further questions regarding this experiment, please contact:

Asia Jakubowska
Department of Computing Science
17 Lilybank Gardens
Glasgow G12 8QQ
e-mail: asia@dcs.gla.ac.uk
tel.: +44 141 330 4256 (ext. 0985)

I have read this information sheet, and agree to take part in this experiment:

Name: _____

Signature: _____

Date: _____

I would like to receive a summary sheet of the experimental findings

E-mail address: _____

K: Questionnaire (Initial Quantitative User Study)

1. How often do you use a computer during your work?

Never |-----|-----|-----|-----| All the time

2. How often do you use a genome browser during your daily work?

Never |-----|-----|-----|-----| All the time

3. If you do use a genome browser, please give the name of the one you use most frequently.

.....

Ensembl questions:

4. What do you like about the Ensembl?

.....
.....
.....

5. What do you dislike about Ensembl?

.....
.....
.....

6. How often do you use Ensembl in your daily work?

Never |-----|-----|-----|-----| All the time

VisGenome questions:

7. What do you like about VisGenome?

.....
.....
.....

8. What do you dislike about VisGenome?

.....
.....
.....

Visual techniques questions:

- 9. Do you think the fisheye visualisation technique is useful? YES / NO
- 10. Do you like it? YES / NO
- 11. Do you think excentric labelling, as used in Ensembl, for example, is useful? YES / NO
- 12. Do you like it? YES / NO
- 13. Do you use panning? YES / NO
- 14. Do you like it? YES / NO
- 15. Do you use zooming? YES / NO
- 16. Do you like it? YES / NO
- 17. Is zooming via buttons, for example in Ensembl, better than via mouse action? YES / NO
- 18. Do you use scroll bars in, for example in Artemis or UCSC Browser? YES / NO
- 19. Which other visualisation techniques did you meet in biological tools?
.....
- 20. Which of the techniques in VisGenome and Ensembl seem to be helpful?
.....
- 21. Are the colours in the visualisation meaningful for you? YES / NO
- 22. If you do use the colours at all in the visualisation, please say how you use them?
.....
.....
.....

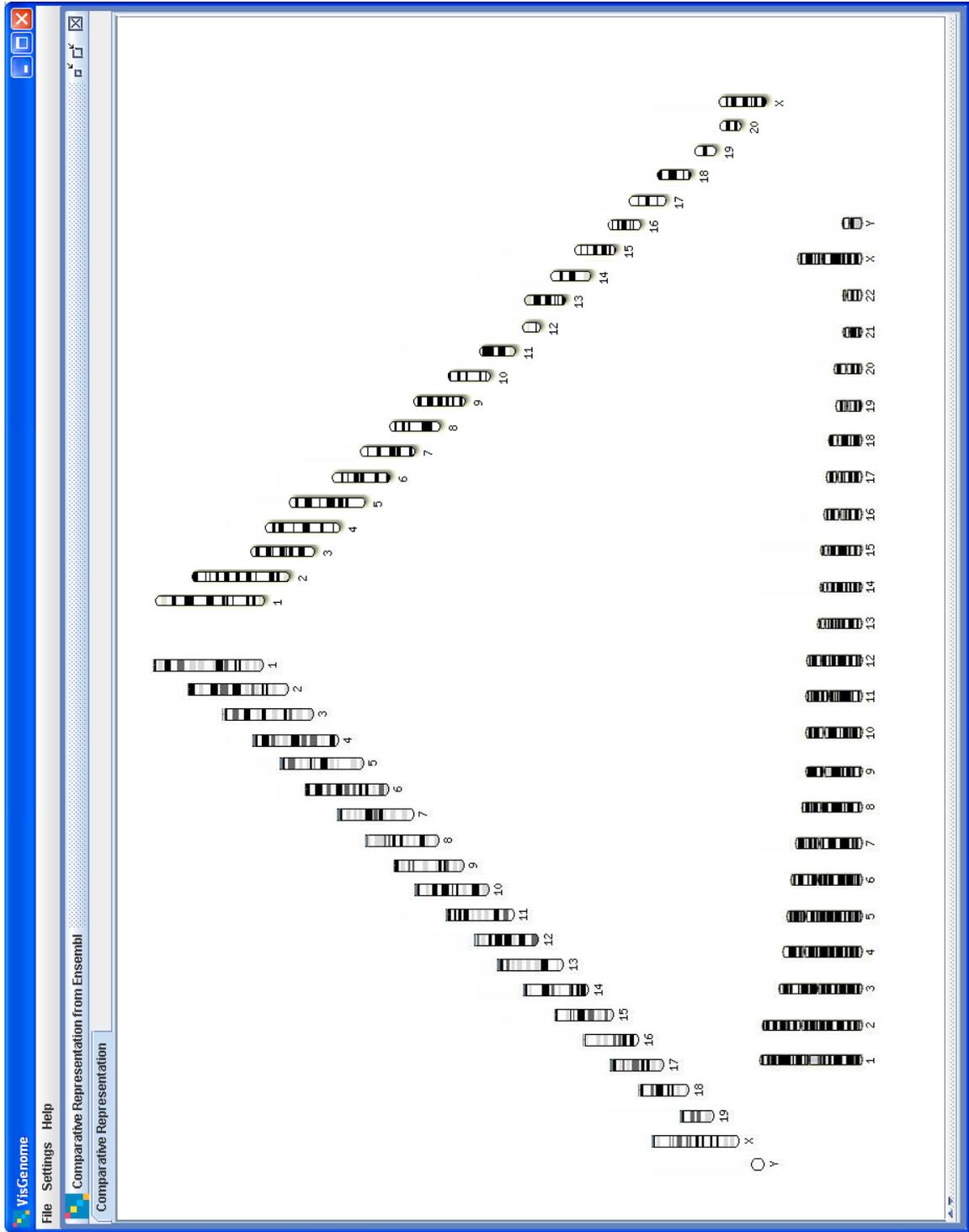
23. You were given the three versions of chromosome presentation at the beginning of the experiment. Please say which of them you prefer. Why do you prefer it?

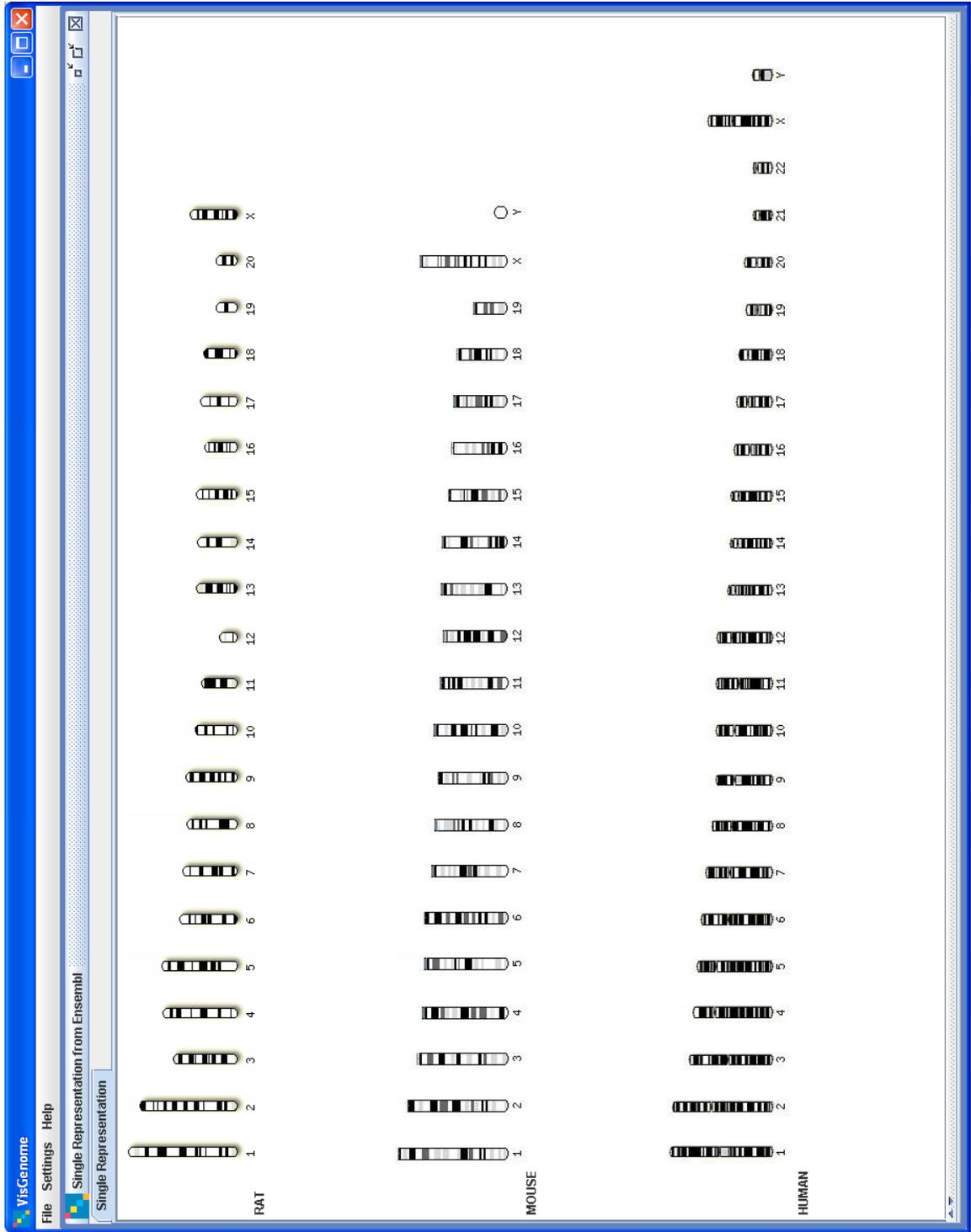
.....
.....
.....

24. Is it important for you to have any additional information about the genes such as the presented in VisGenome in PanelInfo? What would you like to see on it?

.....
.....
.....

Thank you for your help.







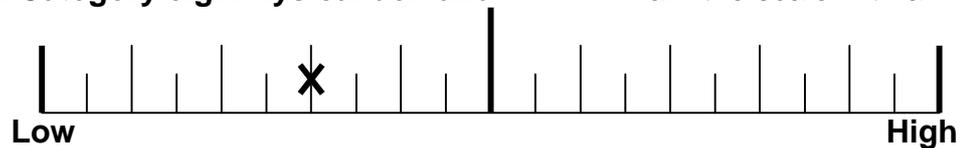
L: Workload Tests

I would now like to examine the “workload” you experienced during the experiment.

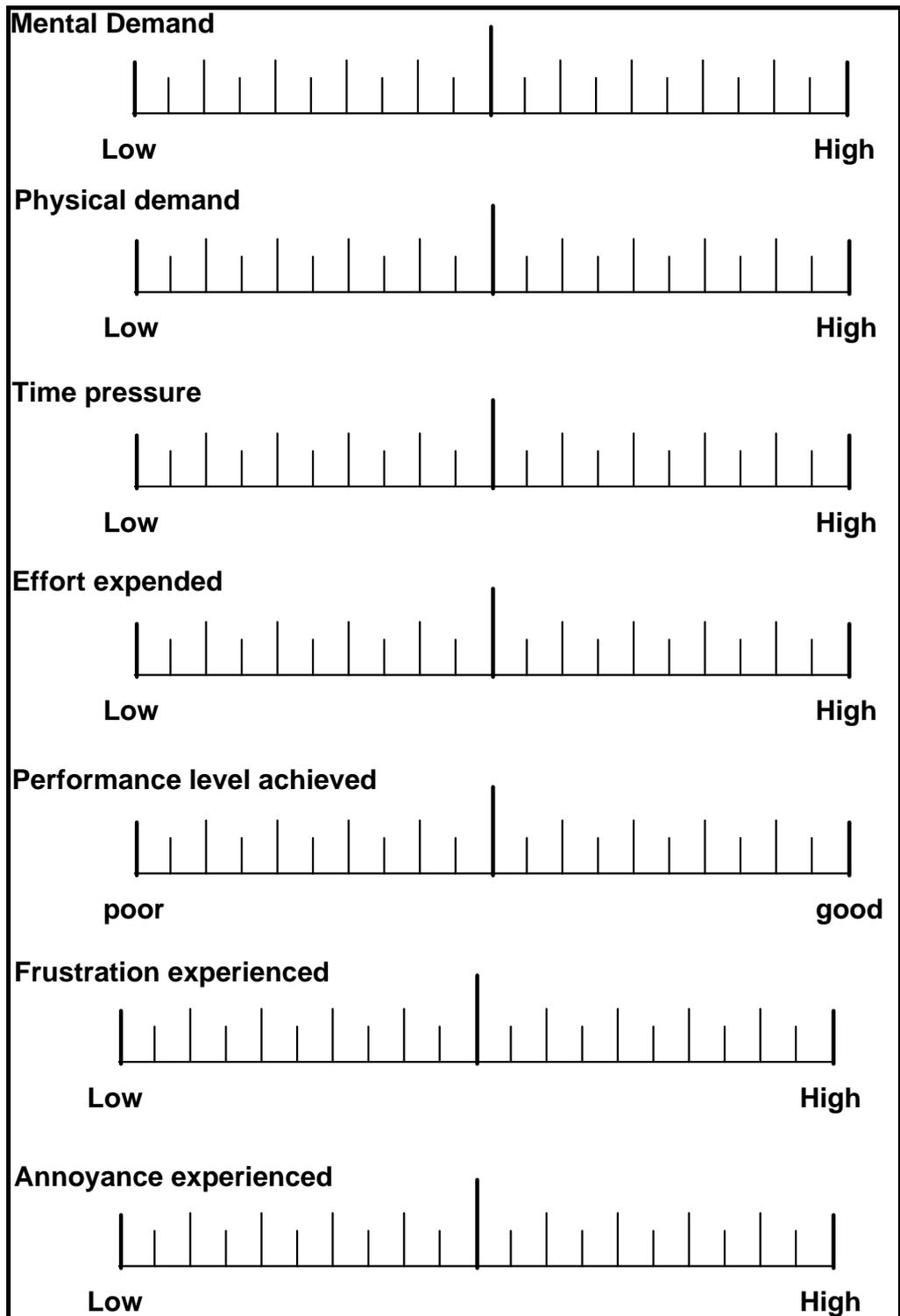
The table below explains the seven categories we are using to assess different aspects of workload. Please read the definitions of the scales carefully. If you have a question about any of the scales in the table please ask me about it as it is important that they be clear to you. You may keep the descriptions with you while completing the scales.

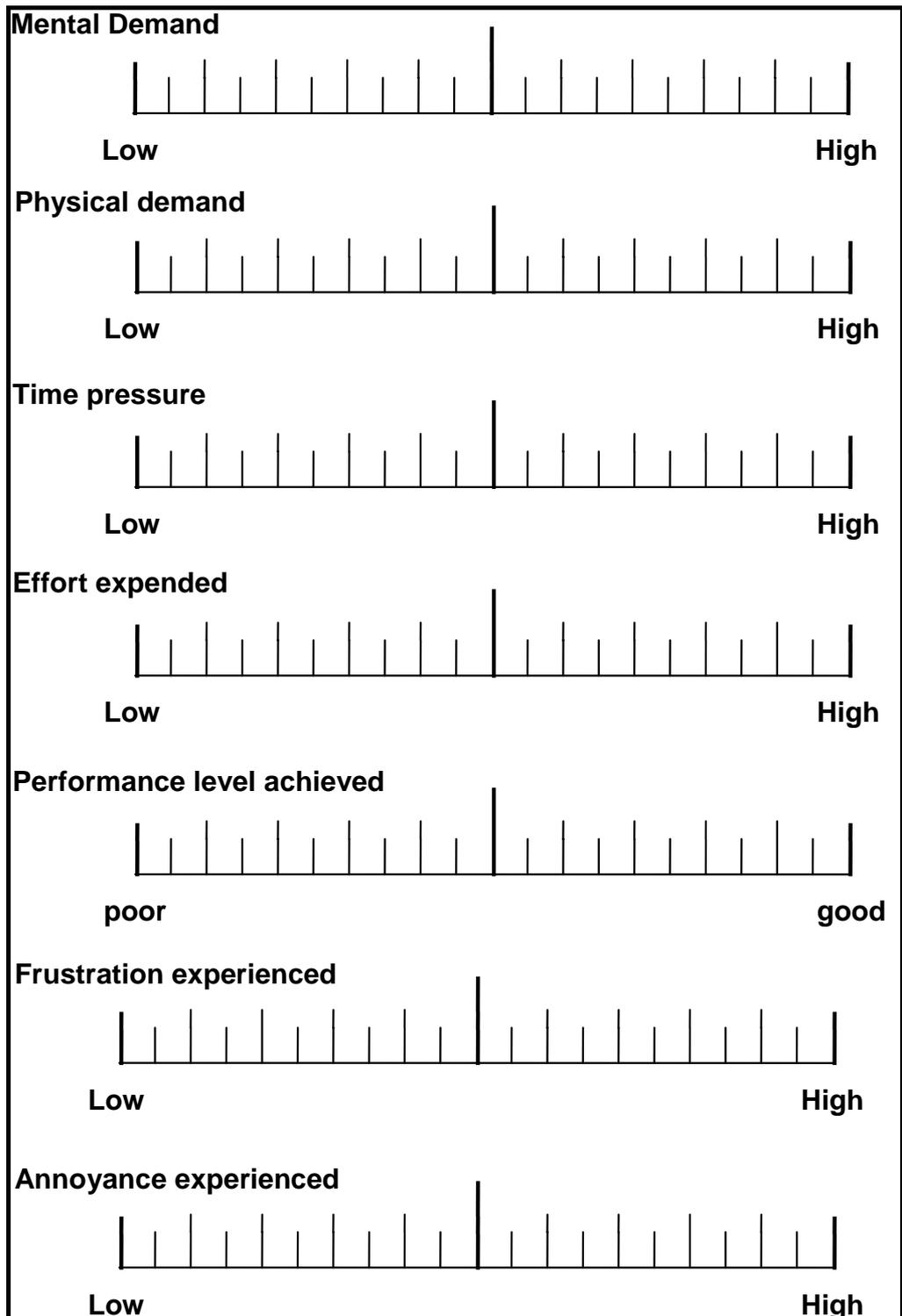
On the following page each category is assigned a linear scale with a description at each end. Please put a cross on one of the vertical bars for each category, at the point on the scale which matches your experience (see example below). Please consider your responses carefully and consider each scale individually. Your ratings will play an important role in the evaluation being conducted, thus your active participation is essential to the success of this experiment, and is greatly appreciated.

Example: Category e.g. Physical demand mark the scale with an ‘x’



Rating Scale Definitions		
Title	Endpoints	Description
Mental Demand	<i>Low/High</i>	How much mental, visual and tactile activity was required? (e.g. thinking, deciding, calculating, feeling)
Physical Demand	<i>Low/High</i>	How physically demanding did you find this experiment? (e.g. did it cause any pain or fatigue, or was the physical demand minimal?)
Time Pressure	<i>Low/High</i>	How much time pressure did you feel because of the rate at which things occurred or the time limit imposed on the task? (e.g. slow, leisurely, rapid, frantic)
Effort Expended	<i>Low/High</i>	How hard did you work (mentally and physically) to accomplish your level of performance?
Performance Level Achieved	<i>Poor/Good</i>	How successful do you think you were in doing the task set by the experimenter? How satisfied were you with your performance?
Frustration Experienced	<i>Low/High</i>	How much frustration did you experience? (e.g. were you relaxed, content, stressed, irritated, discouraged)
Annoyance Experienced	<i>Low/High</i>	How annoying did you find the mouse manipulations used in the experiment? e.g. pleasant, un-intuitive, uncomfortable, intuitive?





Subject _____

VisGenome vs Ensembl

Preference



M: Diary (Mixed Paradigm User Study)

I use VisGenome:	POSITIVE (if VisGenome helps me somehow in my work, done tasks)	NEGATIVE (something in VisGenome what I do not like, what disturbs me)	other suggestions
10.09.2007 <input type="checkbox"/>			
11.09.2007 <input type="checkbox"/>			
12.09.2007 <input type="checkbox"/>			
13.09.2007 <input type="checkbox"/>			
14.09.2007 <input type="checkbox"/>			
17.09.2007 <input type="checkbox"/>			
18.09.2007 <input type="checkbox"/>			
19.09.2007 <input type="checkbox"/>			
20.09.2007 <input type="checkbox"/>			
21.09.2007 <input type="checkbox"/>			

N: Interview Form (Mixed Paradigm User Study)

tools	I used the VT	How frequently do I use the Visualisation Techniques	Data used	Was it useful	Did I succeed	comments
VisGenome	<input type="checkbox"/> zoom <input type="checkbox"/> pan <input type="checkbox"/> scaling <input type="checkbox"/> panel info <input type="checkbox"/> labelling <input type="checkbox"/> <input type="checkbox"/>	1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/>		Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	
Ensembl	<input type="checkbox"/> zoom <input type="checkbox"/> scrolling <input type="checkbox"/> pop up menu <input type="checkbox"/> <input type="checkbox"/>	1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/>		Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	
Excel	<input type="checkbox"/> scrolling <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/>		Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	
Acquisition	<input type="checkbox"/> zooming <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/>		Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	
Rat tail blood pressure determination	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/>		Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	

Microsoft Word/other text editor	<input type="checkbox"/> scrolling <input type="checkbox"/> searching <input type="checkbox"/> <input type="checkbox"/>	1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/>		Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	
Outlook/other e-mail browser	<input type="checkbox"/> searching <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/>		Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	
Internet Browser	<input type="checkbox"/> scrolling <input type="checkbox"/> searching <input type="checkbox"/> <input type="checkbox"/>	1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/>		Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	
	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/>		Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	
	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/>		Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	
	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/>		Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/> Partially <input type="checkbox"/>	

- 1 – daily
- 2 – weekly
- 3 – monthly
- 4 – rarely, a few times per year
- 5 – never

O: Statistical Methods

1 Statistical Methods

In order to interpret experimental results, it is vital to use and understand the appropriate statistical methods for analysing data. Ellis and Dix [2] state that researchers quote either the exact number or convert it into “an apparently over-precise percentage” during result presentation. The authors also point out that people find statistics to be difficult. In order to ensure that the analysis and interpretations of our data were correct, all statistics calculations presented in this thesis were carried out in consultation with a statistician from the BHF Cardiovascular Research Group. We had only 15 participants in our initial quantitative user study and 5 participants in the second user study. This is quite a small number of users from the statistical point of view, and because of this, we could not analyse all the data as we had hoped. Some statistical tests require a higher number of users. Therefore, after consultation with the statistician, we chose statistical tests that could be applied to our data and could show us any significant results from the statistical point of view.

We use the expression **statistical significance** to mean that $p < 0.05$ (p is short for p -value). A p -value is a measure of how much evidence we have against the null hypothesis. The null hypothesis, traditionally represented by the symbol H_0 , represents the hypothesis of no change or no effect. In our first user study (Chapter 6) we compare VisGenome and Ensembl and in task T3 more users succeeded in VisGenome than in Ensembl - the user success for VisGenome was significantly greater for task T3 with $p = 0.0078$. This p -value tells us that the chance of observing this data (or data more extreme than this) is 0.78% (or approximately 1 in 128), assuming that the null hypothesis is true. Here the null hypothesis is that users would be equally likely to succeed with VisGenome as with Ensembl. The results are considered statistically significant when $p < 0.05$ (1 in 20 chances of falsely rejecting the null hypothesis). Clearly, the experimental results could be non-significant. However we set this chance of falsely rejecting the null hypothesis at the arbitrary small value of 1 in 20. It means that they could be generated by random chance or just that the participants taking part in an experiment had “good/bad” day.

1.1 Quantitative Data

	1 sample	2 sample
parametric	t-test	two-sample t-test
non-parametric	Wilcoxon signed rank test sign test	Mann-Whitney test

Table 1: The classification of used statistical tests for measurement quantitative data. The tests can be divided into two groups: parametric and non-parametric tests. We applied them for 1 sample or 2 samples.

During the analysis of both our user studies we used some common statistical terms such as median, mean, normal distribution, standard deviation, or 95% CI. We also used some more complicated tests such as the sign test, the t-tests, the Mann-Whitney test, and the Wilcoxon signed rank test, see Table 1. During the analysis of our first user study we show some data by using the nonparametric Kaplan-Meier survival plot (see Figure 3). We introduce the terminology in this section.

The **sample** [1] is a set of data collected for an experiment.

The **population** [1] is the set (often infinite) of all possible individuals we would have sampled. Statistics aims to answer questions about populations using sampled data. In general we use the sample mean, median, variance, and standard deviation to estimate the population mean, median, variance, and standard deviation.

The **median** [1] is defined as the number separating the higher half of a sample, a population, or a probability distribution, from the lower half. If there is an even number of elements, the median is not unique, so one often takes the mean of the two middle values. For example, 8 is the median for the list of numbers: 1, 4, 8, 20, and 97; $14 = \frac{8+20}{2}$ is the median for the list of numbers: 1, 4, 8, 20, 97, and 110.

The **mean** [1] used in the thesis is an arithmetic mean (\bar{x}) which is calculated by summarising all numbers from a list of numbers and then dividing the number by the number of items in the list. For our two examples presented before we have means $\bar{x} = \frac{1+4+8+20+97}{5} = 26$ and $\bar{x} = \frac{1+4+8+20+97+110}{6} = 40$. A general formula for calculation arithmetic mean is as follows:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

In our statistical calculations we also used term **variance** (σ^2 or $\text{Var}(X)$) [1]. The variance is one measure of statistical dispersion, averaging the squared distance of possible values from the expected value (the mean). The general formula for calculation of variance is:

$$\text{Var}(X) = E[(X - \mu)^2] \quad (2)$$

where $\mu = E(X)$ is the expected value (mean) and X is our sample.

In the situation when we have arithmetic mean, the variance is calculated as follows:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (3)$$

However, we can easily calculate the variance as equal to the mean of the squares minus the square of the mean ($\sigma^2 = \frac{1}{n} \sum_{i=1}^n x_i^2 - (\frac{1}{n} \sum_{i=1}^n x_i)^2 = \frac{1}{n} \sum_{i=1}^n x_i^2 - \bar{x}^2$). For example, if we take our example of list of the numbers 1, 4, 8, 20, and 97 then the mean of the squares is $1978 = \frac{1^2+4^2+8^2+20^2+97^2}{5}$. The square of the mean is $\bar{x}^2 = 26^2 = 676$. Therefore, the sample variance is $\sigma^2 = 1978 - 676 = 1302$.

Since the variance is in squared units, its square root is often quoted instead and is known as the **standard deviation** [1]:

$$\sigma = \sqrt{\text{Var}(X)} \quad (4)$$

Standard deviation (σ)¹ provides a good measure of variability. It measures how widely values

¹We can calculate standard deviation for our example of list of numbers: 1, 4, 8, 20, and 97, and it is $\sigma = \sqrt{1302} = 36.08$. A large standard deviation indicates that the data points are far from the mean and a small standard deviation indicates that they are clustered closely around the mean. For example, each of the three data sets {0, 0, 14, 14}, {0, 6, 8, 14} and {6, 6, 8, 8} has a mean of 7. Their standard deviations are 7, 5, and 1, respectively.

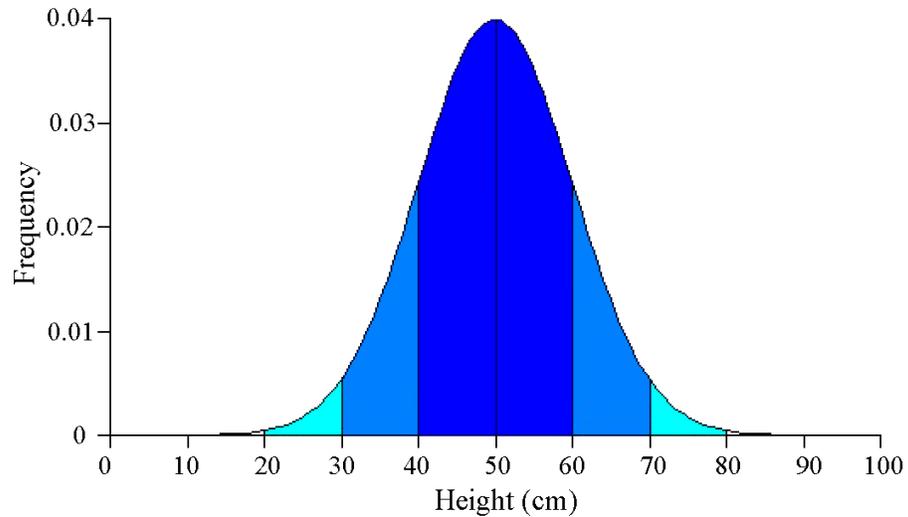


Figure 1: A normal distribution curve taken from [5]. It is often called the bell curve because the graph of its probability density resembles a bell. “The mean is 50 cm and the standard deviation is 10 cm. The dark blue area (the mean plus or minus one standard deviation, 40-60 cm) contains 68% of the total area under the curve. If we include the mid-blue area too (all measurements within 2 standard deviations of the mean, i.e. 30-70 cm)” [5], this contains 95% of all the measurements (2 standard deviations).

are dispersed² from the average.

The **normal distribution** [1], see Figure 1, is also called the Gaussian distribution and it is very important in statistics. The distribution is defined by two parameters: the mean and variance. All normal distributions are symmetric and have bell-shaped density curves with a single peak. The standard normal distribution is the normal distribution with a mean of zero and a variance of one.

It is also worth mentioning the **standard error** [1] of the mean (SE), which is the standard deviation divided by the square root of the sample size:

$$SE = \frac{\sigma}{\sqrt{n}} \quad (5)$$

In our example we have $SE = \frac{36.08}{\sqrt{5}} \approx 16.14$.

A few times, during representing our experimental results, we used the expression “95% C.I.”. A **confidence interval** (C.I.) is “an estimated range of values which is likely to include an unknown population parameter³, the estimated range being calculated from a given set of sample data” [6]. A C.I. is usually calculated so that it contains the true value of the population parameter with 95% confidence interval. It could be also produced at 90%, 99%, or 99.9% confidence intervals for the unknown parameter. It is based on three elements:

- the sample estimate (for example the sample mean= \bar{x}),

²Dispersion is the difference between the actual value and the average value. The larger the differences between the individual values and the average value, the higher the standard deviation will be and the higher the volatility.

³A population parameter is a characteristic of a probability distribution. Examples would be the mean and the variance. We estimate the population parameters different using sample values, e.g. we use the sample mean to estimate the population mean.

- the standard error (SE) of the estimate,
- the desired width of the confidence interval (e.g., the 95% confidence interval or the 99% confidence interval).

In large samples, a C.I. is defined by the following formula:

$$95\%C.I. = \bar{x} \pm (z * SE) \tag{6}$$

which means that 95%C.I. is FROM $\bar{x} - (z * SE)$ TO $\bar{x} + (z * SE)$ where z is the z -score for the particular confidence interval of interest. For example, if we want the 95% C.I. the value of z would be 2 (the value 2 comes from our understanding of the normal curve), then the areas between plus and minus 2 standard deviations will contain the population parameter in 95% of the cases, if the means are normally distributed (see Figure 1 and the area between 30 and 70 cm). If the sample size is smaller than 100, say, then it is better to use a t-test confidence interval. Here z is replaced by t which is slightly bigger according to the sample size.

Statistical methods for quantitative data can be divided into parametric and nonparametric. Parametric methods assume the data follow a normal distribution. Nonparametric methods “require fewer assumptions about a population or probability distribution and are applicable in a wider range of situations” [4].

Where both parametric and nonparametric methods can be used, statisticians usually recommend the use of parametric methods as parametric methods tend to provide better precision [4].

In the chapters concerned with our experimental results we used statistical tests (see Table 1) to find out whether a particular hypothesis can be supported, or needs to be rejected. There are a number of statistical tests and techniques. However, we chose only those that could give us any significant results and could be applied to our data (some of them are appropriate only for discrete or continuous data, some of them expect a number of samples). We used Minitab for all statistical calculations presented in the thesis.

For the questionnaire results from the initial quantitative user study we used the 2-sample **t-test** [1]. The test assesses whether the means of two groups are statistically different from each other. It compares the difference between two means in relation to the variation in the data, see Figure 2. Minitab also provides confidence intervals for these 2 sample t-test and the paired t-test calculations⁴.

In the second user study we used the **Mann-Whitney test** (also known as Wilcoxon-Mann-Whitney test) - the non-parametric equivalent of the 2-sample t-test. The Mann-Whitney tests two independent samples of numerical or ordinal values [1]. These samples do not need to contain the same number of observations. We used the test for all data from our second experiment’s log files as these data were not normally distributed.

In our first user study we used the **Wilcoxon signed rank test** (also known as Wilcoxon matched pairs test) [1] for time measurements in task T2. It is a non-parametric test used when a paired t-test is not appropriate and it tests the median difference in paired data. Each individual in the sample generates two paired data values, one from first measurement and one from second measurement (in our situation one from VisGenome and one from Ensembl). Differences between the paired data values are used to test for a difference between the two populations [4].

During our initial quantitative user study we used the nonparametric **Kaplan-Meier survival plot**, see Figure 3. This method is used for analysing survival data. It is very popular in

⁴For two paired sets of n measured values (in our initial quantitative user study - results from $n=15$ users for VisGenome and Ensembl), the paired t-test determines whether they differ from each other in a significant way under the assumptions that the paired differences are independent and identically normally distributed.

difference between group means

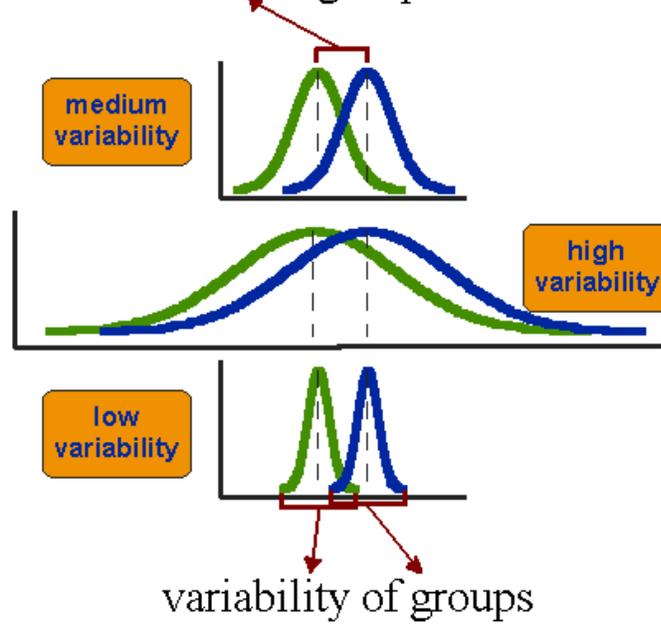


Figure 2: Three scenarios for differences between means. The three graphs present the same difference between the means, but they have different variability of the data groups. The figure is adapted from [7]. Note that the sample size needed to find significant differences will be greatest, the larger the variability of the data is (e.g. it will largest for high variability scenario).

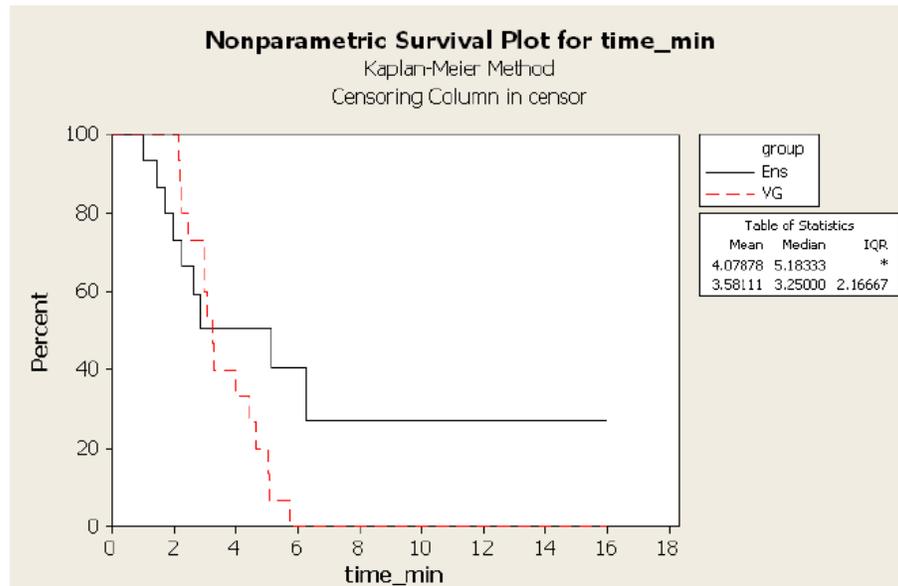


Figure 3: The example of the nonparametric Kaplan-Meier survival plot. Survival plot for time in task T2 conducted during the first experiment, see Chapter 6.

medical research, where the patients receive medicaments which help them (they survive) or in some situations the patients die [1]. In the experiment, “dying” is, metaphorically, finishing in a task in VisGenome or Ensembl. During the analysis, we used the **sign test** as a small sample alternative to McNemar’s test [1] (p.247), in order to compare task completion using VisGenome and Ensembl. The sign test is used to test the null hypothesis that positive and negative results are equally likely. It can also be used to test a hypothesis about a median, because the hypothesis that a median equals 7 (for example), is the hypothesis that equal numbers of cases fall above (positive results) and below (negative results) 7. In our first experimental analysis we used the sign test for median. To perform the test, we simply counted the numbers of cases above and below the hypothesised median (ignoring any cases that exactly equal the hypothesised median), and then calculated the result.

1.2 Categorical Data

We used the **1-proportion test** to test categorical data. This is a hypothesis test of a population proportion [1], and examines the population proportion using information from one sample and comparing it to a target value. We used it during our first experiment to find out users’ preferences: whether they preferred VisGenome or Ensembl.

During our first user study we wanted to use **McNemar’s test**, which is a categorical data method used for the comparison of proportions from paired samples. Q. McNemar introduced this test in 1947 [3], using it on “ 2×2 contingency tables⁵ with a dichotomous⁶ trait with matched pairs of subjects”. However, the test as it stands has a condition which disqualified our data. The condition is $s+t > 10$ where s is the number of successes in the environment A and failures in the environment B and t is the number of successes in B and failures in A. In our first study s is the number of users who succeeded in VisGenome and not in Ensembl and t is the number of participants who succeeded in Ensembl and not in VisGenome, calculated for each task separately. For example, in task T2 we have $s = 9$ and $t = 0$, i.e. 9 participants succeeded in VisGenome and did not succeed in Ensembl, and there was no user who succeeded in Ensembl and did not succeed in VisGenome. In our study $s+t$ was never > 10 , and so we used the two-sided sign test (where 0=both/neither successful; 1=VG success but Ens not; -1=Ens success but VG not). The two-sided sign test is usually used for quantitative data, but in special situations, when we make fewer assumption about a population (we divided all data into three groups) we can use it for categorical data as an alternative to McNemar’s test [1].

⁵A contingency table is used to record and analyse the relationship between two or more categorical variables.

⁶A dichotomy is any splitting of a whole into exactly two non-overlapping parts.

References

- [1] J. M. Bland. An Introduction To Medical Statistics, Third Edition, OUP, Oxford, 2000.
- [2] G. Ellis and A. Dix. An explorative analysis of user evaluation studies in information visualization. In Proc. of BELIV 2006.
- [3] Q. McNemar. Note on the sampling error of the differences between correlated proportions of percentages. *Psychometrika* **12**, 153-157, 1947.
- [4] Encyclopedia Britanica Online. Academic Edition: <http://www.britannica.co.uk>.
- [5] Normal distribution: <http://www.steve.gb.com/science/statistics.html>.
- [6] Statistics Glossary by V. J. Easton and J. H. McColl's: http://www.cas.lancs.ac.uk/glossary_v1.1/main.html.
- [7] The t-test: http://www.socialresearchmethods.net/kb/stat_t.php.

P: Raw Data from Log Files for Mixed Paradigm User Study

1 Raw Data from Log Files

We analysed video and voice recordings, the questionnaires, the diaries, the log files and hand-written notes on my observations during the mixed paradigm user study. In this appendix we present the data from the log files.

In the tables, we show how many times the users carried out a function - see Tables 1, 3, 5, 7, and 9. We also show the duration of each function (in seconds), see Tables 2, 4, 6 8, and 10. The measurement is shown for each function and user. We also show which data is used during the experimental period, see Tables 11, 12, 13, 14, and 15. We do this separately for each log file. In log files we recorded the start time of each function. The duration of each function was calculated as the start time for the next function minus the start time for the current one. I was not present when the subjects created all the log files, however, according to my observation, they often stop to speak with co-workers. Some also left the application running for a whole night until the next day. Therefore I made the assumption that if the function duration is longer than 10 minutes, it should be shortened to 10 minutes. This is in accord with my observations. In all tables in this section we used the following convention: in the first column we show log file number in the order of creation time, in the next columns we show functions or species. We used abbreviations listed below.

SR - single representation

CR - comparative representation

C - colour change (colour an object)

DR - drag region (on the chromosome icon)

FO - focus on

LL - labelling (switch between two labelling modes - all labels or selected labels)

LK - link to Ensembl

P - pan

PS - pan session - pans reduced to sessions

S - scaling (CartoonPlus)

RS - set region (set chromosome region for navigation using info panel)

Z - zoom

ZS - zoom session - zooms reduced to sessions

H - human

M - mouse

R - rat

Each of the functions was explained in detail in Chapters 5 and 7 (VisGenome and VisGenome - Extension).

Tables 1, 3, 5, 7, and 9 (one per user) present how many times the user carried out a function, for *colouring*, *dragging region*, *focusing on*, *labelling*, *linking*, *panning*, *scaling*, *setting region*, and *zooming*.

Tables 2, 4, 6 8, and 10 (one per user) show how long a function was carried out (in seconds).

The measurement is presented separately for single (SR) and comparative representations (CR). Tables 1, 3, 5, 7, and 9 have more columns than Tables 2, 4, 6 8, and 10. This results from two ways of representing of *zooming* and *panning*. First, we counted the number of functions, each iteration step and each session. Second, during analysing times for each of *panning* and *zooming*, we did not need the difference between iteration steps and sessions.

The last five tables in this section: Tables 11, 12, 13, 14, and 15 refer to chromosomes and animals. They show what chromosome from which species was viewed during the experiment.

Table 1: User **D** - count of all functions for each session.

	CR	SR	SR-C	CR-DR	SR-DR	SR-FO	SR-LL	SR-LK	SR-P	SR-PS	SR-S	SR-RS	CR-Z	CR-ZS	SR-Z	SR-ZS
1	0	1	0	0	0	6	2	0	376	4	2	1	0	0	84	2
2	1	1	1	9	0	4	2	2	134	8	2	0	68	2	262	10
3	1	1	0	0	6	12	7	2	9491	28	2	2	0	0	1777	21
4	0	1	0	0	0	0	0	0	301	2	0	0	0	0	137	1
5	0	2	0	0	0	28	3	6	7959	37	2	0	0	0	979	28

Table 2: User **D** - duration of all functions for each session (in seconds).

	CR	SR	SR-C	CR-DR	SR-DR	SR-FO	SR-LL	SR-LK	SR-P	SR-S	SR-RS	CR-Z	SR-Z
1	0	13	0	0	0	20	30	0	127	90	27	0	10
2	18	5	4	3	0	5	10	21	42	10	0	3	43
3	30	130	0	0	53	343	42	601	1338	12	33	0	308
4	0	6	0	0	0	0	0	0	16	0	0	0	13
5	0	46	0	0	0	168	15	1050	740	12	0	0	137

Table 3: User **W** - count of all functions for each session.

	CR	SR	CR-C	SR-C	CR-DR	SR-DR	CR-FO	SR-FO	SR-LI	CR-LK	SR-LK	CR-P	CR-PS	SR-P	SR-PS	CR-S	SR-S	SR-RS	CR-Z	CR-ZS	SR-Z	SR-ZS
1	1	1	0	0	11	7	0	0	0	0	0	494	6	231	6	2	0	0	1670	6	617	9
2	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	30	0	7	0	1	1	0	4	2	7	2	2	2	0	944	7	58	5
4	0	4	0	1	0	4	0	4	4	0	2	0	0	577	0	0	8	6	0	0	1362	30
5	0	3	0	0	0	0	0	6	2	0	1	0	0	352	0	0	1	2	0	0	197	14
6	0	1	0	0	0	4	0	0	0	0	0	0	0	37	0	0	1	0	0	0	65	2
7	0	1	0	0	0	0	0	0	1	0	0	0	0	63	0	0	1	0	0	0	152	7
8	0	1	0	0	0	0	0	1	4	0	1	0	0	145	0	0	1	3	0	0	388	4

Table 4: User **W** - duration of all functions for each session (in seconds).

	CR	SR	CR-C	SR-C	CR-DR	SR-DR	CR-FO	SR-FO	SR-LI	CR-LK	SR-LK	CR-P	CR-P	SR-P	CR-S	SR-S	SR-RS	CR-Z	SR-Z
1	15	8	0	0	5	31	0	0	0	0	0	53	85	22	0	0	0	75	203
2	28	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	15	8	3	0	38	0	30	0	41	21	0	3	64	131	83	0	0	180	13
4	0	23	0	8	0	10	0	601	35	0	92	0	1014	0	110	160	0	1500	1500
5	0	61	0	0	0	0	0	50	1126	0	1	0	108	0	209	74	0	158	158
6	0	10	0	0	0	251	0	0	0	0	0	0	1293	0	40	0	0	14	14
7	0	25	0	0	0	0	0	0	97	0	0	0	129	0	20	0	0	470	470
8	0	7	0	0	0	0	0	1	7	0	0	0	21	0	600	45	0	1222	1222

Table 5: User **J** - count of all functions for each session.

	CR	SR	CR-C	CR-DR	SR-DR	CR-FO	SR-FO	SR-LI	CR-LK	CR-P	CR-PS	SR-P	SR-PS	CR-S	SR-S	SR-RS	CR-Z	CR-ZS	SR-Z	SR-ZS
1	3	1	1	0	0	12	26	2	2	370	5	4579	19	2	2	0	1388	9	1149	19
2	0	1	0	0	8	0	0	2	0	0	0	5439	6	0	3	2	0	0	18	1
3	0	1	0	0	7	0	0	0	0	0	0	11816	9	0	0	0	0	0	336	7
4	0	1	0	0	35	0	0	0	0	0	0	14657	14	0	0	0	0	0	551	11
5	1	0	0	81	0	20	0	0	0	3067	16	0	0	0	0	0	2662	15	0	0

Table 6: User **J** - duration of all functions for each session (in seconds).

	CR	SR	CR-C	CR-DR	SR-DR	CR-FO	SR-FO	SR-LL	CR-LK	CR-P	SR-P	CR-S	SR-S	SR-RS	CR-Z	SR-Z
1	118	7	12	0	52	92	373	86	43	166	3358	31	516	0	112	700
2	0	10	0	0	138	0	0	12	0	0	473	0	8	53	0	7
3	0	5	0	0	97	0	0	0	0	0	1442	0	0	0	0	124
4	0	4	0	0	137	0	0	0	0	0	968	0	0	0	0	61
5	25	0	0	22	0	97	0	0	0	140	0	0	0	0	144	0

Table 7: User M - count of all functions for each session.

	CR	SR	CR-C	SR-DR	CR-FO	SR-FO	SR-LL	CR-LK	SR-LK	CR-P	CR-PS	SR-P	SR-PS	CR-S	SR-S	SR-RS	CR-Z	CR-ZS	SR-Z	SR-ZS	
1	1	1	1	1	0	27	0	4	3	0	127	5	872	13	4	6	0	502	9	300	12
2	0	1	0	0	0	1	0	1	0	0	0	0	1274	20	0	9	3	0	242	12	
3	0	1	0	0	0	0	3	0	0	0	0	0	139	3	0	1	2	0	0	0	
4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	1	0	1	0	3	7	0	3	0	0	0	1954	35	0	0	13	0	1068	32	
6	0	1	0	0	0	0	4	0	0	0	0	0	474	8	0	6	2	0	176	4	

Table 8: User M - duration of all functions for each session (in seconds).

	CR	SR	CR-C	SR-DR	CR-FO	SR-FO	SR-LL	CR-LK	SR-LK	CR-P	SR-P	CR-S	SR-S	SR-RS	CR-Z	SR-Z
1	92	8	17	0	100	0	68	44	0	46	1109	82	100	0	97	139
2	0	12	0	0	0	0	39	0	0	0	1565	0	446	40	0	245
3	0	9	0	0	0	0	37	0	0	0	1698	0	25	64	0	0
4	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	5	0	2	0	0	233	0	413	0	1094	0	0	82	0	282
6	0	5	0	0	0	0	52	0	0	0	419	0	247	18	0	132

Table 9: User **L** - count of all functions for each session.

	CR	SR	SR-C	CR-DR	SR-DR	CR-FO	SR-FO	SR-LL	CR-LK	SR-LK	CR-P	CR-PS	SR-P	SR-PS	CR-S	SR-S	SR-RS	CR-Z	CR-ZS	SR-Z	SR-ZS
1	2	2	0	0	5	0	0	0	0	0	1068	4	345	3	0	0	0	14	1	39	1
2	0	1	0	0	2	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
3	0	1	0	0	2	0	7	1	0	0	0	0	189	7	0	3	0	0	0	65	6
4	0	1	1	0	0	0	27	0	0	12	0	0	71	10	0	1	6	0	0	50	5
5	2	0	0	0	0	11	0	0	0	0	1011	12	0	0	0	0	0	306	10	0	0
6	1	1	0	0	0	46	20	0	6	8	1007	19	32	4	1	3	4	780	12	195	3
7	2	2	0	0	0	3	2	0	0	2	95	3	375	8	0	1	0	174	2	49	4

Table 10: User **L** - duration of all functions for each session (in seconds).

	CR	SR	SR-C	CR-DR	SR-DR	CR-FO	SR-FO	SR-LL	CR-LK	SR-LK	CR-P	SR-P	CR-S	SR-S	SR-RS	CR-Z	SR-Z
1	22	7	0	8	106	0	0	0	0	0	659	132	0	0	0	1	5
2	0	10	0	0	10	0	0	0	0	0	0	1	0	0	0	0	0
3	0	5	0	0	12	0	9	10	0	0	0	66	0	45	0	0	21
4	0	7	4	0	0	0	70	0	0	413	0	125	0	19	104	0	28
5	60	0	0	663	0	11	0	0	0	0	72	0	0	0	0	58	0
6	125	11	0	0	0	57	25	0	44	28	604	63	4	19	70	61	137
7	78	11	0	0	0	222	0	0	0	41	58	235	0	45	0	82	25

Table 11: User **D** - species and chromosomes viewed. User **D** created 5 log files. In log 1 she used the rat chromosome 2 in the single representation. In log 2 - the rat chromosome 2 in the single representation and comparisons between the human chromosome 1 and the rat chromosome 2, and between the human chromosome Y and the rat chromosome 12 were made. The user carried out the experiment with the rat chromosome 3 and looked at a comparison between the rat chromosome 3 and the rat chromosome 2 (log 3). She made a mistake here, because there are no homologous genes between chromosomes from the same species. The user also viewed the rat chromosome 2 (log 4) and the rat chromosome 3 (log 5) in the single representation.

log	CR-H1	CR-HY	CR-R12	CR-R2	CR-R3	SR-R2	SR-R3
1	0	0	0	0	0	1	0
2	1	1	1	1	0	1	0
3	0	0	0	1	1	0	1
4	0	0	0	0	0	1	0
5	0	0	0	0	0	0	1

Table 12: User **W** - species and chromosomes viewed. User **W** created 8 log files, and looked at comparisons between the human chromosome 7 and the rat chromosome 2 (logs 1 and 3). He also made two mistakes and tried to find homologous genes between the rat chromosome 2 and the rat chromosome 3, and between the rat chromosome 2 and the rat chromosome X in second log file. The log files 1-6 show that the user was particularly interested in the rat chromosome 2 (log file 4 shows that he looked 4 times at the chromosome). The participant also looked at the rat chromosome 10 (log 8), the rat chromosome 5 (log 4), and the rat chromosome 3 (log 7).

log	CR-H7	CR-R2	CR-R3	CR-RX	SR-R2	SR-R3	SR-R5	SR-R10
1	1	1	0	0	1	0	0	0
2	0	2	1	1	1	0	0	0
3	1	1	0	0	1	0	0	0
4	0	0	0	0	4	0	1	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	1	0	0	0
7	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	1

Table 13: User **J** - species and chromosomes viewed. User **J** used the single representation only for viewing the rat chromosome 2 (logs 1-4). In log 1 he also compared the rat chromosome 3 and the mouse chromosome 8, and made one mistake when trying to look for a comparison between the rat chromosome 3 and the rat chromosome X. In session 5 he viewed comparisons between the human chromosome 5 and the rat chromosome 2, the human chromosome 3 and the rat chromosome X, and the human chromosome X and the rat chromosome X.

log	CR-H3	CR-H5	CR-HX	CR-M8	CR-R2	CR-R3	CR-RX	CR-R7	SR-R2
1	0	0	0	1	0	2	0	1	1
2	0	0	0	0	0	0	0	0	1
3	0	0	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	1
5	1	1	1	0	1	0	2	0	0

Table 14: User **M** - species and chromosomes viewed. User **M** was interested only in the rat chromosome 2, and the comparison between the rat chromosome 2 and the mouse chromosome 3.

log	CR-M3	CR-R2	SR-R2
1	1	1	1
2	0	0	1
3	0	0	1
4	0	0	1
5	0	0	2
6	0	0	1

Table 15: User **L** - species and chromosomes viewed. User **L** was the only one interested in the human chromosomes 2 and 7. He also looked at the comparisons between the human chromosome 7 and the rat chromosome 2, the human chromosome 2 and the rat chromosome 14, the human chromosome 7 and the rat chromosome 14, and between the human chromosome 7 and the rat chromosome 7. The user made one mistake in log 5, trying to look at a comparison between the human chromosome 7 and the human chromosome 10.

log	CR-H2	CR-H7	CR-H10	CR-R14	CR-R2	CR-R7	SR-H2	SR-H7
1	0	3	0	3	0	0	2	0
2	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	1	0
5	1	2	1	2	0	0	0	0
6	0	1	0	0	1	0	0	1
7	0	1	0	0	0	1	1	1