Usability of Navigation Tools for Browsing Genetic Sequences

Paul Rutherford[1], Walt Abell[1], Clare Churcher[1], Alan McKinnon[1], John McCallum[2]

[1] Department of Applied Computing, Lincoln University, PO Box 84, Lincoln, New Zealand

{ paul.rutherford, walter.abell, clare.churcher, alan.mckinnon } @lincoln.ac.nz

[2] Plant and Food Research, Private Bag 4704, Christchurch 8140, New Zealand

McCallumJ@crop.cri.nz

Abstract

Software to display DNA sequences is a crucial tool for bioinformatics research. This study examined techniques for navigating large DNA sequences via panning and zooming. This involved surveying the navigation facilities of current bioinformatics applications and performing a heuristic analysis on the most common interface controls found. Several prototypes for sequence navigation via panning and zooming were then developed and usability trials carried out, getting users to perform common sequence navigation tasks using the prototypes. The 'Connected View' design was found to be most usable for panning while the zooming results were less clear. The outcomes of this type of research can help improve bioinformatics applications so that will be more usable by the target research users.

Keywords: bioinformatics, DNA sequences, usability, user interface, navigation.

1 Introduction

The technology to display DNA sequences was developed in the mid-1970s and the volume of such data has been growing exponentially since then. Bioinformatics tools which apply computing and statistical techniques to such data are now commonly used. However, much of this software is developed or designed by scientists who typically have little formal training in user interface design issues, or by software developers who often have little understanding of the needs of researchers in the field. It is not uncommon for users of bioinformatics software to experience a steep learning curve and to be overwhelmed by the complexity of performing standard tasks.

The overall aim of this study was to evaluate different approaches for browsing DNA sequences on a computer to improve the usefulness of bioinformatics software. It applied principles of user interface design, navigation and usability to applications that allow users to navigate sequences to look for particular features or attributes. The study looked at the type of browsing capabilities and controls provided by current bioinformatics applications and used these as the basis for the design of several prototypes. The efficacy and efficiency of the prototypes as well as user preferences were determined through a usability trial.

2 Background

DNA sequences are long strings of the letters **A**, **C**, **G** and **T** which represent the nucleotides (commonly called "bases") Adenine, Cytosine, Guanine and Thymine. These letters are repeated in various combinations and can number into the thousands (or even millions) of characters in a single sequence. Clearly it is not possible to display this amount of information on a single screen. However, even sequences of a few hundred letters can still cause information overload for a user.

In addition, sequences are often annotated with a number of "features" which are segments of the DNA known to have a specific purpose. For example, Start and Stop Codons which mark the beginning and end of a sub-sequence and Exons which encode a protein product. Sequences have been displayed in various formats. A common display method is to show the sequence horizontally, with a ruler for the location of the bases and any annotations shown above and/or below the sequence. An example of this is shown in Figure 1 (Lorraine and Helt, 2002).

Figure 1: A simple display of a DNA sequence

While this provides the detailed information for a particular region of the sequence, it is often necessary to look at the annotations over a much larger region, thus requiring a less detailed view, often referred to as an overview. An example of this is shown in Figure 2. Note that the ruler and features are still visible but the base letters are not).

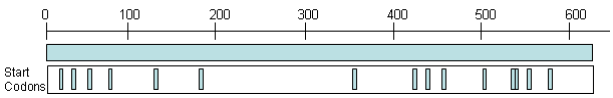

Figure 2: A sequence overview display

This project had the involvement of Crop and Food Research and was funded by the Foundation for Research, Science and Technology (FRST) Contract No C02X0203.

2.1 Information Spaces

The display of a genetic sequence is an example of an information space (Benyon and Höök. 1997). As information spaces increasingly 'go digital', there are some intrinsic characteristics that impact on their navigation. These include the lack of a "stable Euclidean geometry" (Dahlbäck, 1998), relatively unconstrained navigation (Benyon and Höök, 1997) and "a lack of explicit or implicit information that [movement is] in the right direction" (Dahlbäck, 1998). These characteristics combined with the large amount of data that can be stored digitally can contribute to users 'getting lost' which has been identified as a major problem in information spaces (Dillon et al, 1990; Spence, 1999).

Three general navigation activities in information spaces were described by Benyon and Höök (1997). These are listed in Table 1 with an explanation of what each activity is trying to achieve and an example applying to genetic sequences.

Activity	*Objective*	*Genetic Example*
Exploration	To see what objects are present and their relationships.	To investigate the number and order of features in a sequence.
Wayfinding	To browse to a specific location.	Find the location of the first base of the first exon in a sequence.
Identifying objects	To understand information about a set of features.	Find out how many exons there are between the start codon and position 2000.

Table 1: Description of navigation activities

2.2 Navigation Aids

One of the issues in user interfaces for working with large information spaces is how to allow the user to navigate without losing track of where they are in the space. It may also be necessary to carry out comparisons between sections of the data that are quite far apart. Finally it is often necessary to be able to easily switch back and forth between a detailed view and an overview of the data.

Programs that deal with display and searching of genetic sequences suffer from the age old problem of how to show the appropriate level of detail while allowing the user to maintain the context from a larger area than can be accommodated on the screen. This problem has occurred in many application areas and various approaches have used such as distortion techniques (e.g. Fish Eye Lens and Distortion Wall) as well as 'connected views' for overview and detail.

Many of these techniques have been tested experimentally and some implemented in applications. As is often the case, the efficacy and efficiency of an approach varies depending on such factors as how well the feature is implemented, the sophistication of the end user, the type of task undertaken, and the specific application of the techniques involved.

On the other hand, standard office applications (e.g. word processors) and web applications offer somewhat standard approaches to navigation through large documents, i.e. scrolling, zooming, etc. It may be that some of these common approaches are suitable for browsing genetic sequences.

Where non-professional developers (in this case biological researchers) actually carry out application development (or play a significant role in the design), usability considerations may not be a top priority. Typically the types of users who develop bioinformatics applications are primarily interested in obtaining accurate and meaningful output (e.g. a clear diagram from part of a sequence). Features like user friendliness and appropriate interface controls may not be seen as directly contributing to the output and so not receive much attention (especially if software development is not officially part of a user's job description).

3 Purpose of the Research

This study sought to understand how navigation of genetic sequences has been included into various bioinformatics applications and experiment with various ways of offering appropriate navigation features. To this end, the study was structured as follows:

- Cataloguing of the navigation features in current bioinformatics applications that provide genetic sequence browsing. This was followed by a heuristic evaluation of the user interface controls for browsing found in the applications.
- Development of several prototypes for sequence browsing that employ the most promising user interface controls identified in the heuristic evaluation.
- Performing a usability study on the prototypes developed to determine the efficacy, efficiency and user preference for type of control.

4 Bioinformatics Applications

There is a wide range of software available to support bioinformatics research, ranging from databases for lab management to 2D and 3D visualisation of data. For the purposes of this project, software was examined that allows some form of sequence browsing.

Altogether, 20 applications were examined including many in wide use within the Bioinformatics research community, e.g. BLAST (McGinnis & Madden, 2004) for comparing new sequences to a global database and Ensembl (Hubbard et al., 2005) for accessing data from the GenBank sequence database. The appendix contains a complete list of the applications surveyed.

Each application was examined to determine:

- the number of views provided, e.g. overview and detail
- the user interface controls provided for changing the views, e.g. panning or zooming
- the 'connectedness' of the views, i.e. did changing one of the views cause a change in the other views

For example, Ensembl provides several interconnected views at different levels of detail. The interface is very 'space-intensive', sometimes requiring multiple screens to view all the information. A (cut down) example of the display is shown in Figure 3.

Ensembl views may be panned and zoomed however these transitions require the display to be refreshed. Panning is provided through buttons that move a fixed distance in a particular direction. There are also buttons for zooming as well as a control to select the zoom level.

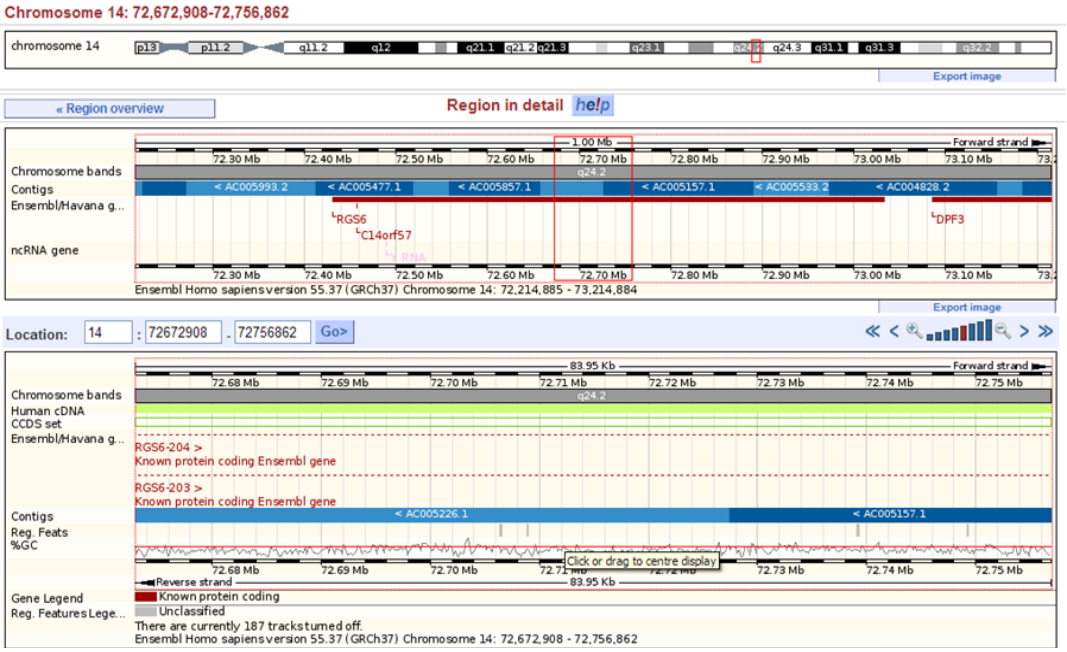

Figure 3: Sample display from Ensembl

A heuristic evaluation of each of the designs identified for panning and zooming was then undertaken using the 10 usability principles defined by Nielsen (1994). Each design was examined for issues that conflicted with one or more of the usability principles. Each issue was rated as a 'problem' (would definitely affect users) or a 'warning' (could affect users but the effect could be minimised through minor redesign). For example, Table 2 shows the evaluation for the Zoom Control in Ensembl.

Type	*Principle*	*Explanation*
Problems	Internal model	No consistent way to interpret the scale; could associate small bars with more detail or interpret as showing less detail (i.e. overview).
	User control	Only a limited set of levels available.
	Recognition	Difficult to label buttons to indicate detail and overview. Design relies on recall and/or complex labelling.
Warnings	Standards	Non-standard design but familiarity with buttons may compensate.
	Recognition	Labelling important so users will recognise purpose of each button.

Table 2: Issues for Ensembl Zoom Control

In addition to bioinformatics applications, the designs of a few common applications (e.g. Acrobat Reader and Google Maps) that provide panning and/or zooming were also evaluated. This was done to consider whether user interface designs from common software could be useful for sequence browsing.

A summary of the designs from current applications and the problems and warnings produced by the evaluation is shown in Tables 3 and 4 (ordered from least to most problems/warnings).

As can be seen from the panning list, the most common design uses scroll bars. This is probably due to perceived user familiarity with this common control. The use of this control is also fairly well understood as evidenced by the low number of issues in the heuristic evaluation.

Design	*Configuration*	*Occurrences*	*Problems*	*Warnings*
Scroll bar	Horizontal	14	1	1
Connected view	Overview displayed above detail	8	1	4
Buttons	Two or four buttons, horizontal	4	2	3
Hand tool	Drag view in either direction	2	4	2
Circular map	Small circular overview	2	4	3
No panning		1	-	-

Table 3: Evaluation of panning designs

Design	*Configuration*	*Occurrences*	*Problems*	*Warnings*
Slider	Horizontal or vertical	2	1	2
Buttons	2-4 buttons to alter zoom or one button to toggle between overview and detail	6	2	3
On-view slider	Slider is superimposed on view	1	2	4
Select level	Choose from several zoom levels	2	3	2
Magnifying glass	Use mouse buttons to set zoom level	1	4	1
Marquee Tool	Select region to zoom	2	4	1
Dynamic Zoom	Drag mouse in 'zoom mode'	1	5	1
No zooming		8	-	-

Table 4: Evaluation of zooming designs

The results for the zooming designs were less clear cut, especially as many of the applications did not provide any method to adjust the detail level. Of those that did, the most common control was buttons of varying types. It is interesting that sliders did not feature in more applications as these had relatively few issues and operate similarly to scroll bars.

5 The Prototypes

Prototypes that provided panning and zooming of sequences were developed in Flash. These were based on the top three designs of each type from the heuristic evaluation. Figure 4 shows a screenshot of the prototype with a Connected View control for panning.

The sequence display was created from screenshots of the detail window of the Artemis 7 application (Rutherford et al., 2000). For the zooming prototypes, the images were manipulated to provide various levels of detail and code was included to provide smooth transitions between levels. Details from the Artemis overview window were used to construct a display of sequence features used in the Connected View control.

The controls used in the panning designs were all oriented horizontally:

Panning Buttons Pan left or right at two different speeds or go to the start/end of the sequence.

Scroll Bar Pan by clicking the arrow keys or in the tray or by dragging the thumb.

Connected View Pan by clicking the arrow keys or in the tray or by dragging the thumb. The tray shows a sequence overview with the main features highlighted.

The controls for zooming were vertically oriented but the prototypes also included a horizontal scroll bar for tasks that required panning.

Zoom Buttons

Click + to zoom in, – to zoom out.

Zoom Slider

Drag slider down to zoom in, up to zoom out. Can also click + and – buttons to zoom.

On-view Slider

Drag the slider to zoom in or out. Slider will follow mouse pointer. View will pan if mouse is moved to left or right edge of view.

6 Usability Trials

The prototypes were incorporated into an overall application for the trials. The application contained an introduction to the display and terminology used followed by sections presenting and testing each prototype design.

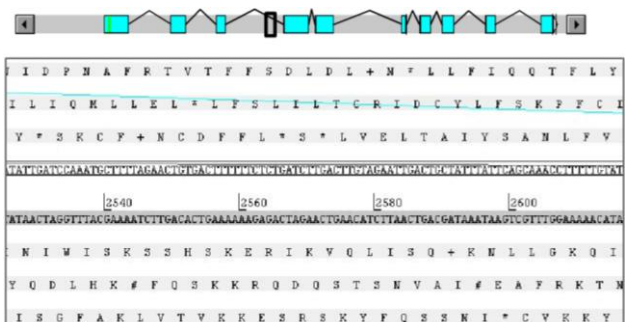

Figure 4: Connected View prototype

There were three different tasks for the user to perform with each design:

Task	*Example*
Find a Feature	Find the location (number) of the first base of the first exon.
Go to Location	Find the four bases on the sequence from location 2000.
Identify features	Find the number of features between the first exon and position 2000.

The same tasks were repeated for each prototype design but the locations and data were varied. Tasks were presented at the top of the screen (with an answer box to fill in) and the prototype showing the sequence and relevant controls was displayed below this. Pre-testing was carried out to refine the application and the terminology used.

Participants were recruited from biological research staff and students working at Lincoln University and the nearby Crown Research Institutes. The only pre-requisite was having had some prior experience of working with genetic sequences on a computer. Human Ethics Committee approval was obtained before participants were approached. A total of seven participants were involved.

At the start of each trial, the participant was briefed by the researcher reading from a usability script. The researcher then started the trial application and observed the participants as they worked on the tasks, making notes on a pre-printed observer sheet. The application also recorded the mouse actions and timings to a file and Camtasia was used to record the screen display and user interaction for further analysis. Each trial was scheduled to last for an hour.

7 Results and Discussion

There were several items which were evaluated for each prototype, some based on the data recorded by the application and some on observations and discussion with participants.

Efficacy	Were users able to get the correct answers for tasks?
Efficiency	How much time/effort was required for each task?
Usage	How did users actually use the controls?
Preference	Which designs did users prefer after completing the trial?

As there were only seven participants in the study, a formal statistical analysis was not undertaken. Instead a descriptive approach was used to analyse the results.

7.1 Panning Designs

All tasks for all prototypes were completed and correct answers given by all participants (100% efficacy). There seemed to be little confusion about how to use each control, possibly due to the relative familiarity of the designs chosen.

To determine the efficiency of use, the amount of time that a participant worked on a task and the number of mouse actions used were compared. One participant was excluded from this analysis as they took significantly more time and mouse actions to complete tasks than the other participants. Figure 5 shows the average time and mouse actions for the other six participants.

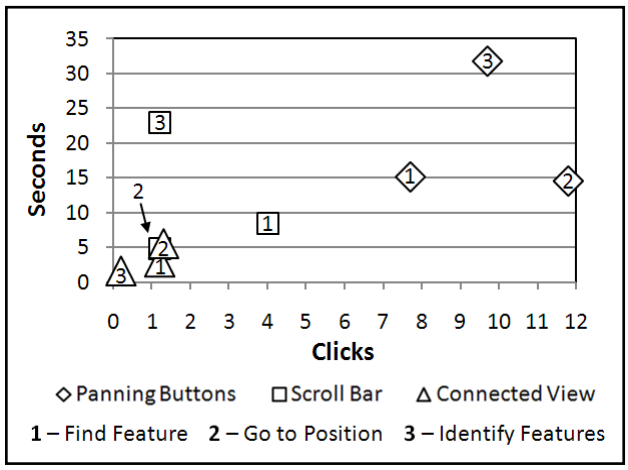

Figure 5: Average Number of Seconds and Clicks for Panning Tasks

Note that for the Connected View, the Find Feature task could be completed by simply inspecting the tray (which displays an overview of the sequence features) without having to click the mouse. Hence the average number of mouse clicks is less than one in this case. It is possibly not surprising that the Connected View was the clear winner in both minimising the amount of time and number of mouse actions required to complete tasks.

Many participants commented on the benefit of the additional information provided by the overview display embedded in the Connected View control. One participant said "it's good [because] you can see what you're coming up to, or go straight to where you want to go". Several participants suggested showing "location indicators" in the overview as these would have assisted in the Go to Position tasks. One participant also suggested the addition of 'Go to Start' and 'Go to End' buttons to the design.

The Panning Buttons were the least efficient approach while the Scroll Bar provided mixed results. To understand why these controls performed so poorly, further analysis of the actual usage of these controls was undertaken. Figure 6 shows a typical example of the use of the Panning Buttons in the Find Feature task.

As can be seen, the user began the task by immediately going to the start of the sequence (the view for each task started somewhere in the middle of the sequence). After pausing (possibly to reorient themselves), they panned right 'fast' (double arrowhead button) three times, overshooting the location for which they were searching. This required them to backtrack, using progressively shorter bursts of movement to ensure they did not overshoot again. Participants had mixed reactions to this design, some describing it as "good" while another called it "annoying".

Figure 7 shows a typical example of the use of the Scroll Bar control for the Find Feature task. Here the user completed the task with one continuous drag action. First they moved to the start of the sequence in two motions. This was completed quite slowly; perhaps they were checking the features of the sequence as it scrolled by. After reaching the start of the sequence, they paused and then quickly panned right through the sequence, overshooting the feature and then backtracking. Despite some inefficiencies in usage, participants described this design as "responsive" and "easier to use" than the Panning Buttons.

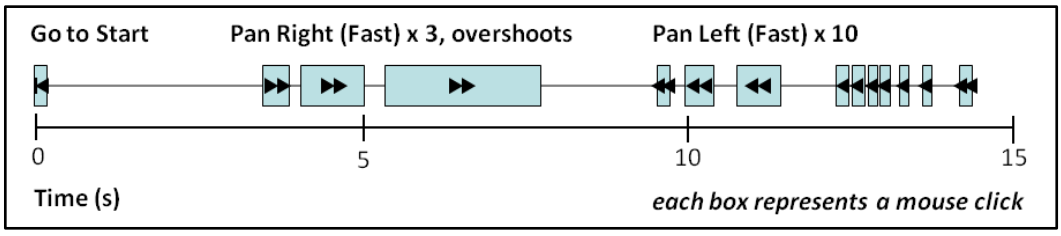

Figure 6: Example of Find Feature task using Panning Buttons

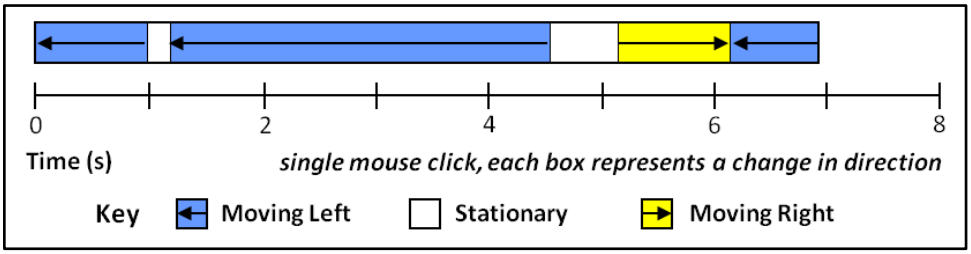

Figure 7: Example of Find Feature task using Scroll Bar

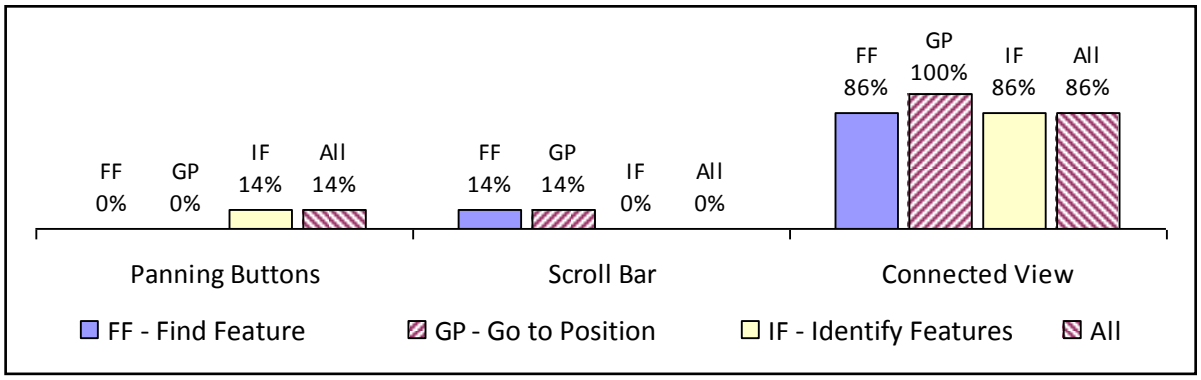

Figure 8: Percentage of participants preferring each panning control by task

Not surprisingly, participants overwhelmingly said they preferred the Connected View control for all tasks and overall (see Figure 8). One participant could not choose between the Scroll Bar and Connected View for the Go to Position task (both preferences have been included in the chart).

7.2 Zooming Designs

The results for the zooming designs were not so straight forward as for the panning ones. For one thing, participants were not restricted to only using zooming as a scroll bar was included in each design to allow panning (and the On-view Slider also performed panning). In addition, the tasks for the zooming section of the trial were the same as those tested in the panning section (but with different sequence locations). Participants were free to use zooming or not to complete the tasks. Figure 9 shows the percentage of the seven participants who did not use the supplied zooming control for each task. It is probably not surprising that zooming was least used in the Go to Position task as this involved finding a specific (numeric) location in the sequence.

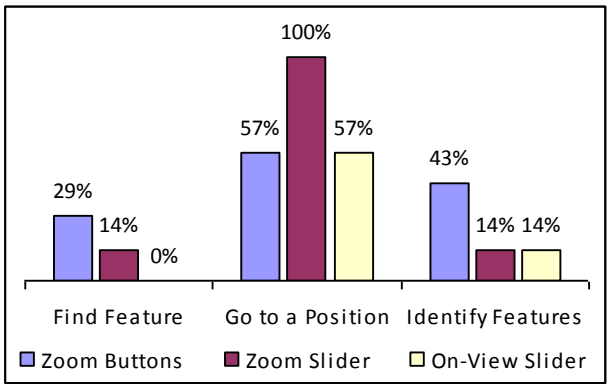

Figure 9: Percentage of participants NOT using zooming controls in the zooming tasks

Seven answers to the task questions (a third) were incorrect in the zooming section (as opposed to none in the panning section). This is surprising especially given that the tasks were essentially the same in both sections. Most of the errors were minor, e.g. an obvious data entry error or miscounting of the number of features. However, some of the errors may have been due to the way the program displayed the sequence which caused some distortion of the text when zooming was used. There was also a bug in the code for the On-view Slider which caused its panning behaviour to be inconsistent when used with the panning scroll bar. This affected one participant's responses but did not appear to impact on other participants.

To analyse the efficiency of use, the timing data was separated into tasks where only the panning scroll bar was used versus where the zooming control was used as well. Figure 10 shows the average number of seconds required to complete each task for both situations. As for the panning results, the times for the participant who took significantly longer have been excluded. Note that the times must be interpreted cautiously as some represent data from only one or two participants.

In almost all tasks, the efficiency of using the scroll bar alone was better or the same as also using the zooming control. Indeed only in the Identify Features task using the Zoom Slider (where all but one participant used the control) was use of the zooming control noticeably faster than panning alone.

The overall advantage of the 'panning only' approach may be explained by some participants commenting that they had not previously used software that provided zooming of sequence displays. Also, the behaviour of the prototype controls was not always what participants anticipated. For example, several said that they expected the centre of the zoomed image to be in the centre of the view but the prototype did not always do this accurately.

Those using the zoom controls did so in various ways but a typical approach for the Find Feature task was to zoom out, pan to find the feature, and then zoom in on the feature in one or two movements. Figure 11 illustrates an example of this approach.

Figure 12 shows the percentage of participants preferring each zoom control for the various tasks. These figures should also be treated cautiously because not all participants used the zoom control in every task. It can be seen that some designs were preferred by users who did not actually use them to do the task (but the controls were demonstrated and explained to each participant).

Some participants described the Zoom Slider as "more straightforward" than the On-view Slider. The one participant preferring Zoom Buttons overall said that if the Zoom Slider had been displayed horizontally rather than vertically, it would have been their equal preference.

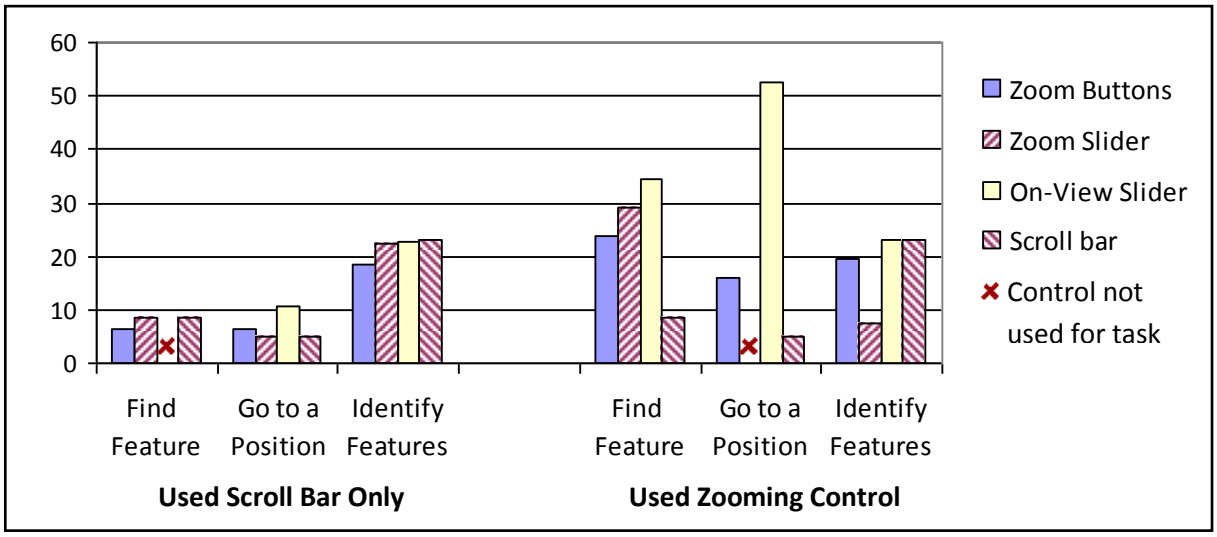

Figure 10: Average seconds to complete zooming tasks with and without use of zooming controls

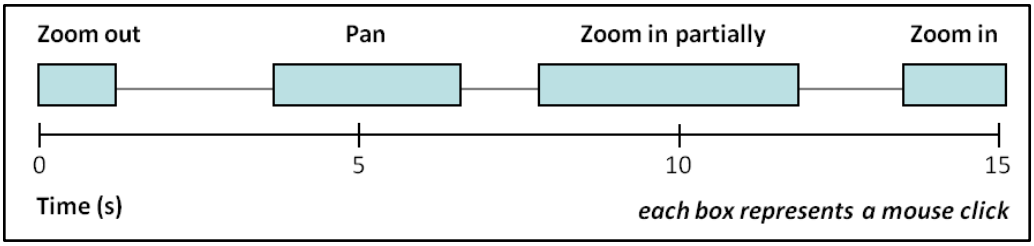

Figure 11: Example of "Find Feature" task using Zoom Slider and scrolling

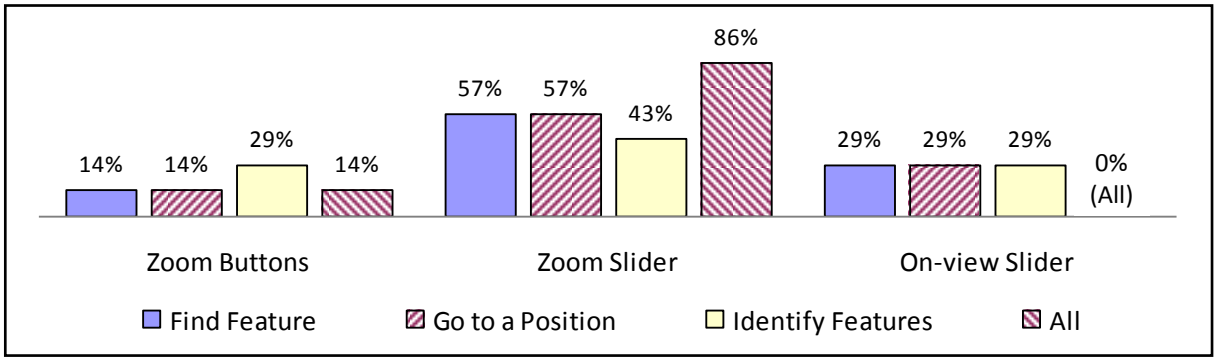

Figure 12: Percentage of participants preferring each zooming control by task

8 Conclusions

This study evaluated a number of common designs in bioinformatics software to browse genetic sequences. Based on this, a set of prototypes to provide panning and zooming were constructed and a usability trial performed. The panning results were unambiguous with the Connected View being the most efficient as well as most preferred control. It would clearly be useful for developers of sequence browsers to consider some form of this control for navigation.

The zooming results were less clear cut but illustrated the tendency for users to use features with which they are already familiar. In this case, it meant a number of tasks were completed by panning using the scroll bar and with no use of the supplied zooming control. The use of zooming is much less prevalent in existing bioinformatics applications and this may account for its low use in the trial. In addition, the tasks required were relatively straight forward (and the same as those tested with the panning controls). There were also some glitches in the operation of the zooming controls which may have put some participants off. It would be instructive to design tasks that would more obviously benefit from zooming and repeat this section of the trial (with improved versions of the controls) to see what impact this has on users' approaches.

The controls tested in this study could form the 'building blocks' of full sequence browsing software. Future work could look at how to add additional facilities for real life browsing tasks, e.g. to compare sets of features from different parts of a sequence. In addition,

there are often a number of parallel 'tracks' of features and annotations attached to a sequence. It would be useful to consider how to adapt the Connected View to be able to show a variety of features.

Finally these prototypes attempted to provide 'smooth' panning and zooming displays. This is different to the majority of current bioinformatics applications which tend to redisplay the whole screen, particularly when changing the level of detail displayed. If would be interesting to test whether smooth displays would better enable users to maintain context and orientation within a sequence.

Bioinformatics software is evolving (and the number of applications increasing) at a rapid rate. Since this study was carried out, newer versions of some of the applications evaluated have been released. In most cases, they have new features for particular sorts of analyses. In a few cases, the user interface has been improved by the addition of better labelling or more predictable behaviour. It is essential that usability issues are key design criteria for bioinformatics software if it is to be of maximum value to researchers who are increasingly reliant on it.

9 Appendix

Bioinformatics applications examined in this study.

1. APIC (Bisson & Garreau, 1995)
2. Apollo (Lewis et al., 2002)
3. Artemis (K. Rutherford et al., 2000)
4. BLAST (McGinnis & Madden, 2004)
5. ChARMView (Myers, Chen, & Troyanskaya, 2005)
6. DNAMAN (Woffelman, 2004)
7. Ensembl (Hubbard et al., 2002)
8. GAP (Bonfield, Smith, & Staden, 1995
9. GeneViTo (Vernikos et al., 2003)
10. Genotator Browser (Harris, 1997)
11. Gestalt (Glusman & Lancet, 2000)
12. MEGA (Kumar, Nei, Dudley, & Tamura, 2008)
13. NCBI Map Viewer (Wheeler et al., 2005)
14. NEBcutter (Vincze, Posfai, & Roberts, 2003)
15. Primer3 WWW Interface (Rozen & Skaletsky, 2000)
16. RegulonDB (Salgado et al., 2001)
17. SeqScape (Applied Biosystems, 2004)
18. Sequencher (Gene Codes Corporation, 2003)
19. SeqVista (Hu et al., 2003)
20. UCSC Browser (Karolchik et al., 2002)

10 References

Applied Biosystems. (2004). SeqScape 2.5.

Benyon, D., & Höök, K. (1997): Navigation in information spaces: supporting the individual. *Proceedings of INTERACT'97, London: Chapman and Hall*, 39-46.

Bisson, G., & Garreau, A. (1995). APIC -- A generic interface for sequencing projects. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*.

Bonfield, J. K., Smith, K. F., & Staden, R. (1995). A new DNA sequence assembly program. *Nucleic Acids Research, 23*(24), 4992-4999.

Dahlbäck, N. (1998): On spaces and navigation in and out of the computer. *Exploring Navigation; Towards a Framework for Design and Evaluation of Navigation in Electronic Spaces. Technical Report T, 98*, 15-29.

Dillon, A., Richardson, J., & McKnight, C. (1990): Navigation in Hypertext: a critical review of the concept. In D. Diaper, D. Gilmore, G. Cockton & B. Shackel (Eds.), *Human-Computer Interaction-INTERACT'90* (pp. 587-592). North Holland: Amsterdam.

Gene Codes Corporation. (2003). Sequencher 4.2: Ann Arbor, Michigan.

Glusman, G., & Lancet, D. (2000). GESTALT: a workbench for automatic integration and visualization of large-scale genomic sequence analyses. *Bioinformatics, 16*(5), 482-483.

Harris, N. L. (1997). Genotator: A Workbench for Sequence Annotation. *Genome Research, 7*(7), 754-762.

Hu, Z., Frith, M., Niu, T., & Weng, Z. (2003). SeqVISTA: A graphical tool for sequence feature visualization and comparison. *BMC Bioinformatics, 4*(1).

Hubbard, T., Barker, D., Birney, E., Cameron, G., Chen, Y., Clark, L., et al. (2002): The Ensembl genome database project. *Nucleic Acids Research, 30*(1), 38-41.

Karolchik, D., Baertsch, R., Diekhans, M., Furey, T. S., Hinrichs, A., Lu, Y. T., et al. (2002). The UCSC Genome Browser Database. *Nucleic Acids Research, 2003*(31), 1.

Kumar, S., Nei, M., Dudley, J., & Tamura, K. (2008). MEGA: A biologist-centric software for evolutionary analysis of DNA and protein sequences. *Briefings in Bioinformatics, 9*(4), 299.

Lewis, S. E., Searle, S. M. J., Harris, N., Gibson, M., Iyer, V., Richter, J., et al. (2002). Apollo: a sequence annotation editor. *Genome Biology, 3*(12).

Loraine, A. E., & Helt, G. A. (2002): Visualizing the genome: techniques for presenting human genome data and annotations. *BMC Bioinformatics, 3*(19).

McGinnis, S., & Madden, T. L. (2004): BLAST: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Research, 32* (Web Server Issue), W20.

Myers, C. L., Chen, X., & Troyanskaya, O. G. (2005). Visualization-based discovery and analysis of genomic aberrations in microarray data. *BMC Bioinformatics 6*, 146.

Nielsen, J. (1994): Heuristic Evaluation. In J. Nielsen & R. Mack (Eds.), *Usability Inspection Methods* (pp. 25-62): John Wiley and Sons, Inc.

Rozen, S., & Skaletsky, H. (2000). Primer3 on the WWW for general users and for biologist programmers. *Methods Mol Biol, 132*(3), 365-386.

Rutherford, K., Parkhill, J., Crook, J., Horsnell, T., Rice, P., Rajandream, M.-A., et al. (2000): Artemis: sequence visualization and annotation. *Bioinformatics, 16*(10), 944-945.

Salgado, H., Santos-Zavaleta, A., Gama-Castro, S., Millán-Zárate, D., Díaz-Peredo, E., Sánchez-Solano, F., et al. (2001). RegulonDB (version 3.2): transcriptional regulation and operon organization in Escherichia coli K-12. *Nucleic Acids Research, 29*(1), 72-74.

Spence, R. (1999): A Framework For Navigation. *International Journal of Human-Computer Studies, 51*, 919-945.

Vernikos, G. S., Gkogkas, C. G., Promponas, V. J., & Hamodrakas, S. J. (2003). GeneViTo: Visualizing gene-product functional and structural features in genomic datasets. *BMC Bioinformatics, 4*(53).

Vincze, T., Posfai, J., & Roberts, R. J. (2003). NEBcutter: a program to cleave DNA with restriction enzymes. *Nucleic Acids Research, 31*(13), 3688-3691.

Wheeler, D. L., Barrett, T., Benson, D. A., Bryant, S. H., Canese, K., Church, D. M., et al. (2005). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research, 33*, D39.

Woffelman, C. (2004). DNAMAN for Windows, Version 5.2.10: Lynon Biosoft, Institute of Molecular Plant Sciences, Leiden University, Netherlands.