From: ISMB-97 Proceedings. Copyright © 1997, AAAI (www.aaai.org). All rights reserved.

Novel Techniques for Visualising Biological Information

Alan J. Robinson and Tomas P. Flores

EMBL Outstation - The European Bioinfomratics Institute
Wellcome Trust Genome Campus, Hinxton
Cambridge, UK
alan@ebi.ac.uk, tflores@ebi.ac.uk

Abstract

The major challenge facing the bioinformatics community is the continuing increase in the number, size and complexity of biological databases with which it must contend. The goal of the research discussed herein is the development and utilisation of techniques that allow researchers to extract new and useful information from these burgeoning information resources using advanced visualisation methods and paradigms, coupled with distributed object technologies that allow communications between applications and remote databases. Visualisation has roles not only in analysis, but also in building more user-friendly interfaces, implementing methods to navigate large information spaces intuitively and powerful techniques to browse and query data. By using platform-independent object-oriented programming languages, these resources may be developed as reusable pieces of software componentry with their methods and interfaces defined fully, and then distributed through organisations such as the bioWidget Consortium. The widget and object-oriented approach is a powerful paradigm in developing new applications from existing components. Development time is reduced and greater time is spent on analysing these data, rather than in the writing of monolithic applications. More powerful applications can be constructed from components interacting in concert and offers the opportunity of a new generation of bioinformatics tools.

Introduction

As the variety, quality and quantity of information in biological databases increases at an unprecedented rate, it becomes necessary to consider how these data may be best analysed and sifted to extract valuable, previously unknown and potentially useful information. Given this potential wealth it is essential to focus on the development of new and exploitation of existing visualisation techniques in order to navigate and explore these new data successfully. The release of a combination of timely technologies has facilitated the provision of a coherent infrastructure upon which new models in visualisation may be built. These new advances include software to implement distributed object technology which provides easier access to remote databases, platform-neutral programming languages allowing "write once, use anywhere" applications and the development of libraries of reusable object-oriented software components, or widgets, from which developers may create customised applications.

The foremost objective of this paper is to describe novel techniques with which to display, navigate and understand data, and present these data in such a way that the powerful cognitive abilities of the human mind may be used to their maximum potential. Visualisation is a key element in other disciplines. Many of the visualisation tools developed in these disciplines may have application within bioinformatics. For this reason, a meta-index of Web pages has been created that catalogue sites, applications, techniques and papers that may be of interest and relevance to the bioinformatics and biological community. These awareness pages are hosted and maintained at the European Bioinformatics Institute (EBI) (Robinson, 1996).

The development of high-level languages and platform-neutral binaries, such as Tcl/Tk (Ousterhout, 1994) and Java (Sun Microsystems Inc., 1994), plus the Web as a medium for dissemination has dramatically accelerated the transition time from inception to prototype to working application. Developers may design and demonstrate a prototype application rapidly, then integrate feedback from users into the application during the development phase, rather than between releases. Applications are now widely available from their beta release and have a continuous evolution with the concept of discrete releases vanishing.

An important concept in this paper is that of reusable software componentry, or widgets. Widgets are pieces of object-oriented code that perform a specific function and are the building blocks of object-oriented programs. By bundleling widgets together, a programmer may build a new application easily and quickly.

Individual visualisation techniques are less likely to be useful in data analysis than the combination and integration of many separate tools and techniques. This approach will only scale if collections of individual components are available that may be combined easily, and then communicate with each other. This is a crucial part of the bioWidget Consortium's (bioWidget Consortium, 1996) API for bioWidgets (Crabtree, 1996).

Careful consideration should be given to the design of object-oriented widgets. By reducing the dependency of widgets on the semantic nature of the information they display or process, the greater the variety of applications and data sets to which the widgets may be applied (Searls,

1995a; Searls, 1995b). The principles of object-oriented programming were devised to foster the reuse of code, and encapsulation plays a central role in ensuring this goal. Widgets are designed to encapsulate a well-defined behaviour. Understanding of their internal implementation details is not required. Instead widgets are integrated through interfaces. By defining widgets with robust interfaces and inter-operability in mind, it becomes tractable to plug combinations of these widgets together in a coherent manner and build applications with functionality beyond that conceived initially by the widget's developer (Stein, Rozen and Goodman, 1994; Goodman, Rozen and Stein, 1995). As a consequence of this, the proportion of time a developer has to spend in prototyping and testing code is reduced significantly. By abstracting the implementation from the semantics of the data, it is then the core code, or "widget glue" that binds the widgets together, and provides the specialisation which defines the nature of the application.

'Focus+Context' Displays

An important visualisation concept is the 'focus+context' paradigm. The philosophy behind this approach is that while a user focuses upon a subset of their data, they should retain a sense of its context within the global scheme. This provides not only a means to navigate rapidly and conveniently through large, complex data sets, but also to observe both short and long range correlations and patterns in data simultaneously and their possible inter-relationship. A real world example of a 'focus+context' view is a fisheye camera lens. Pictures taken with such cameras have an extremely wide field of vision. At the centre of the image, the view is highly magnified while moving towards the periphery, objects diminish rapidly in size.

An example of such a 'focus+context' viewer for displaying data is the hyperbolic projection view which maps points into hyperbolic space and displays them as a 2-D representation. Analysis of data using hyperbolic space was developed initially as a means to layout and visualise large tree structures in an intuitive manner (Lamping and Rao, 1994; Lamping, Rao and Pirolli, 1995), allowing the user to both examine a section of tree nodes without losing their sense of position within the tree, and navigate through the tree intuitively.

Within bioinformatics, the most obvious application of the hyperbolic projection principle is in the display of phylogenetic information. The taxonomy database maintained by the National Center for Biotechnology Information (NCBI) reflects current phylogenetic knowledge and is, as far as is possible, a sequence-based taxonomy using published authorities. An application has been written that allows users to query and browse these data and uses a hyperbolic projection to display the resulting phylogenetic tree. The application exemplifies both the use of reusable componentry and distributed object technology in order to reduce both the development time and the number of lines of code needed to be written by a programmer. As the data are brought back from the server, they are used to construct a tree that is displayed using a hyperbolic projection in an applet window (see Figure 1). The hyperbolic projection viewer and its component widgets were written as a contribution to the bioWidget Consortium's effort to provide high-quality, reusable componentry of benefit and use to the bioinformatics and biological community (bioWidget Consortium, 1996).

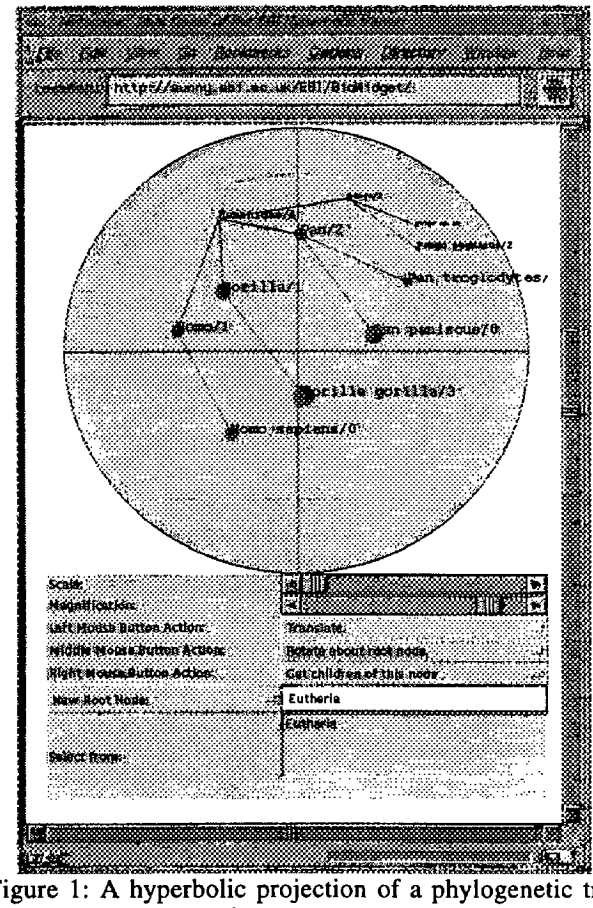

Figure 1: A hyperbolic projection of a phylogenetic tree that illustrates the 'focus+context' technique. Objects toward the perimeter of the view are reduced in size, while those in the centre are magnified. The applet may be downloaded from http://sunny.ebi.ac.uk/EBI/BioWidget/

Widgets may be divided roughly into graphical and non-graphical widgets. Graphical widgets possess methods to display information in a graphical context. Non-graphical widgets encompass other functionality, e.g. they may either process and operate on information, or provide a data model for the storage of information. The applet illustrated in Figure 1 is comprised of four separate widgets, each of which could be reused in other applications, plus the component code that binds them together. The HyperCanvas widget is a graphical widget that provides the canvas upon which the objects are drawn and an easier

front-end to the Converter non-graphical widget which maps coordinates from their hyperbolic plane coordinate system to that of the screen, and from screen to hyperbolic plane. The Tree non-graphical widget provides a mechanism to store the data structure of the tree and access and query its elements. Although the Tree widget contains methods to calculate the layout of tree structures, it is not a graphical widget as it does not include methods to write to the screen. Finally there is the TaxonomyServer non-graphical widget which provides the methods to query the remote taxonomy server and hides all the code that is necessary to access the database over the network. This allows the database to be treated as if it were a local object to be queried. The accessing of the database is done through interfaces provided by implementing CORBA (The Object Management Group and X/Open, 1992).

The architecture of the application is such that the Tree widget provides the means to store the data model of the tree which is constructed from entries retrieved by the TaxonomyServer widget. The graphical HyperCanvas displays the tree by accessing the data stored in the Tree widget through the Converter object which converts the actual positions of the tree nodes into a screen position. In this way, the graphical HyperCanvas object remains abstracted from the semantics of the data in the Tree widget. Communication between the widgets is orchestrated by the component code that binds the widgets together and provides the control interfaces.

The HyperCanvas and Converter widgets are not dependent upon semantic detail and thus may be used to project any data, provided it has some notion of its position in space. The Tree widget is more dependent on semantics and the nature of the data it describes. However the basic Tree widget may be extended to provide this functionality. This abstraction of the widget implementation from the semantics of the data facilitates greater reuse of the widgets. By reusing widgets, a developer may concentrate their efforts upon providing the application functionality, rather than implementation details of individual components.

The Display of Large Information Spaces - Information Murals

A drawback of the hyperbolic projection 'focus+context' approach is that the coordinate space of the points is distorted during the projection into hyperbolic space. A user might wish to examine the complete data set without focusing in upon a particular portion in order to view the spatial relationship, e.g. clustering or distance, between points. This will be hampered by the distortion of the data space. A technique is required to examine large data sets that may contain more data points than there are screen pixels available. Information murals (Jerding and Stasko, 1995a; Jerding and Stasko, 1995b; Jerding and Stasko, 1996) address this challenge by using graphics aliasing and anti-aliasing techniques. At its simplest, an information mural treats a screen pixel as a histogram bin into which data points are collected. The intensity of each pixel when displayed is proportional to the number of points that have fallen within its bin. This technique adds extra information to scatter plots since clusters of pixels containing many data points will appear brighter than similarly sized clusters made up of fewer points.

To illustrate the application of information murals, Figure 2 shows two scatter plots displaying the percentage sum of guanine and cytosine content along a DNA sequence 0.6 megabases long. Regions rich in guanine and cytosine are usually taken to indicate the possible location of genes. The percentage sum was calculated every 50 bases by averaging over the next 200 bases. Thus the informational mural is attempting to describe over 12,000 data points in the display area. For comparison, a conventional monochrome plot where a pixel is drawn for every data point is shown below the plot using an information mural (N.B. A blue to red colour scheme was used in the actual plot so as to increase the contrast between pixels). From the information mural depiction, it can be ascertained which peaks correspond to a longer length of the DNA showing a high combined guanine and cytosine content, and not just localised, or spurious, points. The former will occur as bright points, and the latter as dimmer objects. This information is missing from the conventional scatter plot. Using an information mural has increased the information content of the display without complicating it significantly.

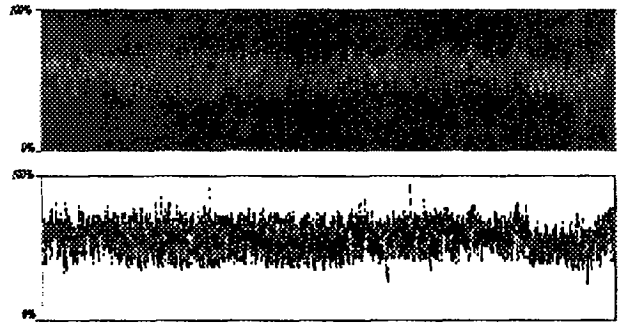

Figure 2: Two different displays of the combined guanine and cytosine percentage content along a DNA sequence 0.6 megabases long. a) The information presented using an information mural b) The same information in a normal scatter plot.

The code to generate the information murals was written in Java and is thus object-oriented and readily reusable. Furthermore its methods are designed not to be dependent upon the semantic nature of the data. The information mural widget has already been used in a range of varied applications, e.g. the display of feature maps of long DNA sequences from the EMBL Nucleotide Sequence database as well as to show average base composition. The applications have also included zoomable information murals that allow users to magnify and examine interesting

regions in more detail. Information murals are a technique that should have a wide application in many fields and not just bioinformatics. The technique may be extended to 3-D projections using voxels, the 3-D equivalent of pixels, where clusters of points in space would appear as bright "constellations".

Visualising Semantic Information - Semantic Lenses and Filters

The data that are to be analysed in bioinformatics are often rich in semantics, e.g. a single SWISS-PROT or EMBL Nucleotide Sequence entry contains large amounts of information. This necessitates techniques which will allow users to filter data and only examine those data, or part of them that are of interest. This requires methods that operate upon the semantics of these data and hence involves designing widgets which, rather than being general, are instead easily extensible to support all possible types of behaviour. Just as ordinary lenses or filters can magnify and change the appearance of objects beneath them, so an ideal tool would be a lens or filter which would focus or filter information intrinsic to the data points beneath them. Such tools are called semantic lenses and filters (Bier, et al., 1993; Stone, Fishkin and Bier, 1994) and in an entirely analogous fashion to regular lenses, semantic lenses may be overlaid to produce quite complicated lenses, or queries. To a certain extent, semantic lenses incorporate the 'focus+context' ideology.

Figure 3 shows the implementation of some semantic lenses operating on data points within a graph. The basic lens has been written to be fully extensible, allowing developers to design and implement specialised behaviour for their lenses in a straightforward manner. Furthermore, lenses do not have to be simple data filters, they may also operate upon the underlying data. Thus lenses could be conceived of that implement gene prediction behaviour. If several of these lenses, each implementing different algorithms are overlaid and passed over a DNA sequence, then a consensus view may be displayed that highlights those regions most likely to correspond to genes.

The current semantic lenses were implemented in Java during an effort to clarify the presentation of the relationship between a group of protein entries from the SWISS-PROT database. The data were homology scores from BLAST sequence comparisons between pairs of proteins from a set of forty-nine proteins. These scores were then used in a geometric minimisation such that those proteins showing greater homology to each other were placed closer together in space.

Originally, the data points of Figure 3 had been shown as points in 3-D space (see Figure 4), however this scene turned out to be both difficult and slow to navigate with users getting lost because of the unsuitability of computer screens in showing 3-D data and the loss of context when a viewer zoomed into a scene. So a 2-D approach was tried using semantic lenses (see Figure 3). The images of Figure 3 show that the applet uses a 'focus+context' type philosophy, also called 'pan and zoom'. In the top level window all the data points are shown with the output of the lens within it directed to a daughter window that shows an enlarged view of these data. It is however easy to maintain how the zoomed view relates to the overall scene because the position of the daughter lens is shown clearly in the parent. Figure 3 also illustrates how lenses in the daughter window may be overlaid; one lens shows the name of the organism that the protein originated from, the other shows the SWISS-PROT entry name of the protein. Where these lenses are overlaid, both names are shown for underlying points.

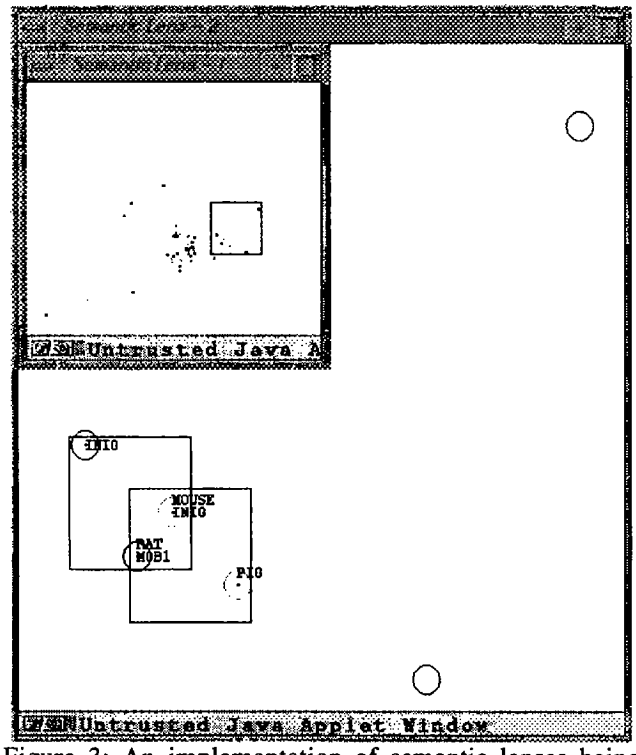

Figure 3: An implementation of semantic lenses being applied to data points corresponding to SWISS-PROT entries. The parent window contains a single lens, the output of which is directed to the daughter window which contains two further semantic lenses. (See text for details). An example applet implementation of semantic lenses may be downloaded from http://industry.ebi.ac.uk/~alan/SemanticLenses/

Using the 2-D 'pan and zoom' parent and daughter window approach, plus the power of semantic lenses, both navigation and comprehension of the data becomes more intuitive. These are extremely important considerations in the design of any user interface or application. There is no reason why semantic lenses could not be applied to 1-D data, e.g. DNA and protein sequences, to 3-D data where the lens is either a 2-D window operating on data behind it, or a 3-D shape that acts upon points contained within it, or a lens that is placed over other widgets, e.g. information murals. For the last option to be feasible, communication

between widgets is essential which implies they must have a common interface to handle events between them. Such an interface is part of the bioWidget API.

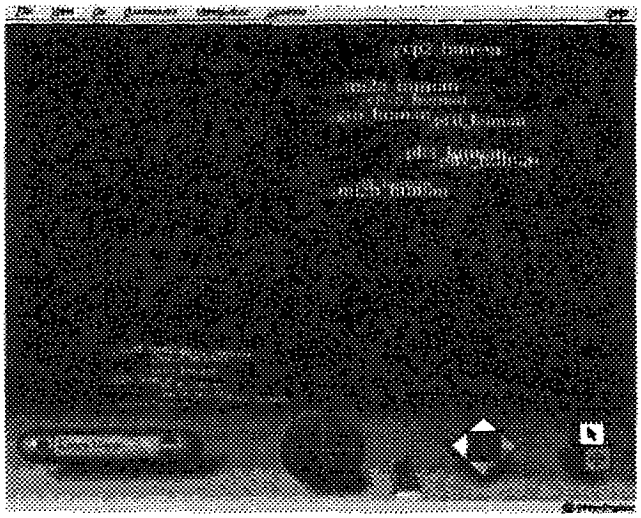

Figure 4: A 3-D non-linear map of the SWISS-PROT entries from Figure 3 shown using VRML. This may be downloaded from http://industry.ebi.ac.uk/~alan/VRML/

This simple example shows how lenses have been implemented that deal with points whose semantic information relates to SWISS-PROT entries by extending the simple semantic lens behaviour widget. Thus using this Java application, it is straightforward to visualise and analyse which proteins are grouped by homology and pass lenses, which behave as queries or filters, over points which are observed to cluster in the scatter plot so as to uncover finer detail and relations. The Java widget that does the geometric optimisation of the positions of the proteins, the semantic lens widget and its extensible behaviour widgets, plus the basic widgets used to describe data operated upon by the lenses are to be submitted to the bioWidget Consortium.

Exploring Other Fields - "Reusable Applications"

Bioinformatics is not the only discipline that has had to contend with extremely large and complex data sets. In fact it is a relatively new problem to bioinformatics and it is prudent to examine research in other fields that have had similar challenges to resolve. By surveying the capabilities of applications already available for data visualisation, it can be determined if they include techniques that may be of relevance within the bioinformatics and the biological sciences. This offers the possibility of ascertaining rapidly if it is worthwhile developing the technique further by demonstrating it to potential end-users. If it is considered useful, then the options are to either re-implement the techniques as widgets which allows their reuse in other applications, or to use the application itself (which may require its further development).

By way of an example, the program MapViz, which is part of the MineSet data mining suite of packages produced by Silicon Graphics Inc. (SGI) (Silicon Graphics Inc., 1996), is used normally to display the geographical and spatial relationships of data, e.g. the spending per household in States of the USA would be shown as a map of the USA with the height of each state corresponding to expenditure. However the principle may also be applied to genomic data, i.e. the geographic map may be replaced by a genomic map. This effort was a collaboration between the EBI and SGI. SGI converted the raw data provided by the EBI into a suitable format for integration into the MapViz application.

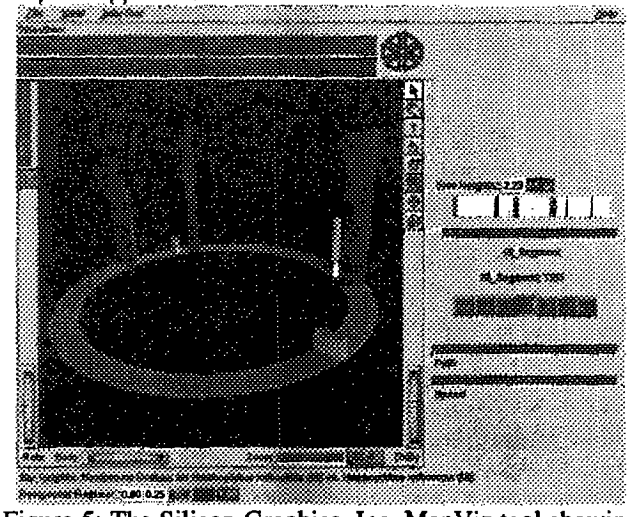

Figure 5: The Silicon Graphics, Inc. MapViz tool showing a representation of the Haemophilus influenzae circular chromosome. Peaks correspond to regions that are similar to the current query sequence.

Figures 5 and 6 show screen shots of the application MapViz being used to view and identify regions of similarity between the Haemophilus influenzae and Mycoplasma genitalium circular chromosomes. The circular chromosomes are displayed as rings upon which information relating to genome position is superimposed. The data originated from two sets of calculations. In the first set, Haemophilus influenzae was compared with itself by dividing the genome into stretches of DNA 1000 bases in length and using FASTA to return a score describing the similarity between this sequence and other 1000 base length sequences around the rest of the genome. In the second instance, these 1000 base length sequences from Haemophilus influenzae were compared against the Mycoplasma genitalium genome. Thus two sets of multiple FASTA queries were run amounting to a significant quantity of data. Figure 5 shows MapViz with the self-comparison of Haemophilus influenzae for one of the 1000 base sequences where the height of peaks around the chromosome correspond to the degree of similarity between the query sequence and that region of the genome.

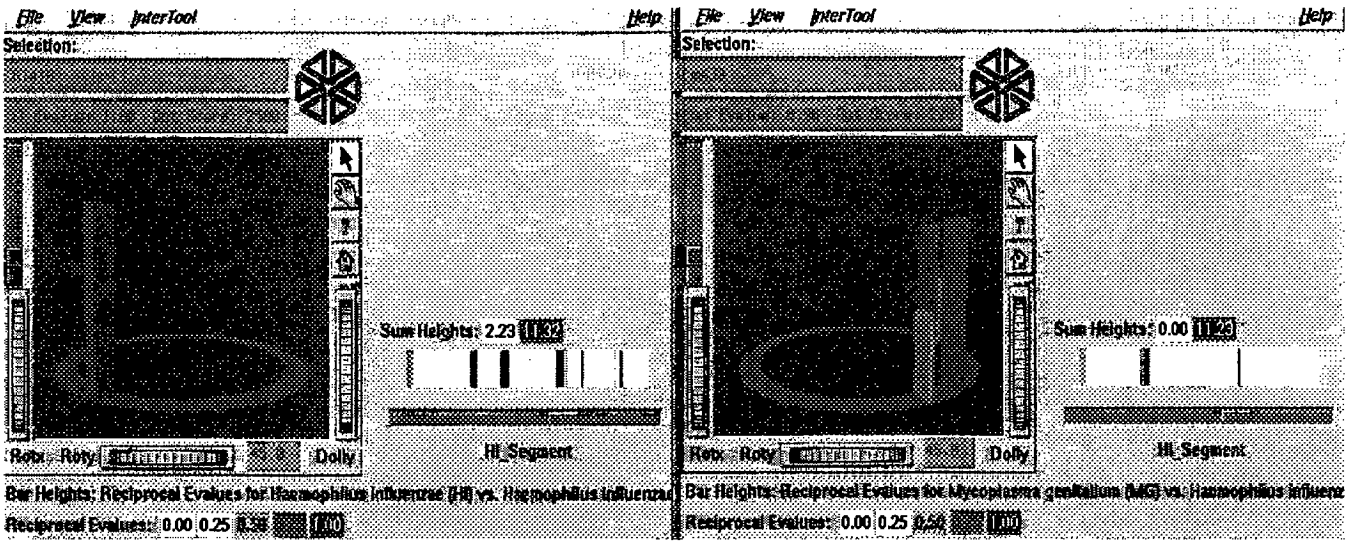

Figure 6: The Silicon Graphics, Inc. MapViz tool showing a representation of the Mycoplasma genitalium circular chromosome (right) that has been synchronised with that of the Haemophilus influenzae chromosome (left). It can be seen that 3 regions on the Mycoplasma genitalium chromosome have similarity to the current query sequence of Haemophilus influenzae. (See text for details).

Thus Figure 5 shows that two other regions in the chromosome have significant similarity with the current query sequence. A scroll bar is used to move around the circular genome and hence change the current query sequence. During this movement, regions showing similarity to this query sequence appear as peaks, the height of which corresponds to their similarity to the query sequence.

Figure 6 shows the Mycoplasma genitalium circular chromosome. One of the features of MapViz is the ability to tie connected data sets together. Thus the Mycoplasma genitalium view may be synchronised with the Haemophilus influenzae view so that as the user moves around the Haemophilus genitalium chromosome, regions that are similar to the current query sequence in both Haemophilus influenzae and Mycoplasma genitalium chromosome, appear as peaks. Hence it is easy to observe regions that are common between the two genomes. This is an example of where communication between widgets is essential and leads to extra functionality. Another feature of the MapViz tool is the ability to "drill-down" into peaks. Thus the user may examine finer detail, or call up further information by selecting a peak. The application may be extended to include the possibility of using novel, or multiple probes to query the underlying data, or extra information may be included in the visual display by colouring peaks differently according to a property of the sequence, e.g. the percentage guanine plus cytosine content. This would allow the application to show not only which sequences are similar, but also if they are potential genes.

Work is ongoing at Silicon Graphics Inc. in developing and refining the MineSet application for use in bioinformatics. MapViz is a case in which an existing program has been shown to have useful application within bioinformatics; an area in which it had not been envisioned it would have scope. During demonstrations of MapViz to audiences as a genome navigation tool, opinion has been polarised sharply about its potential. But these early demonstrations have also lead to many suggestions on usful features which should be included and so accelerated its development into a useful tool. Furthermore, significantly less time and effort have been expended in tuning MapViz to test the potential of its techniques in bioinformatics, than in its previous development for other areas in which it is successful. Thus if it transpires that a program is unsuitable for the proposed application, little effort has been lost. However if an application shows potential, then a new but previously proven software application becomes available to scientists, and software developers discover a new market.

Data Integration through Visualisation

The cross-pollination of data and ideas between scientific disciplines may have an important impact on data presentation and interpretation. Fields may not be fully aware of how parallel research in other areas uses different procedures to study similar data or how it might impact their own data. Yet new methodologies may significantly enrich the data of both parties. Many fields are now converging on the analysis of biological data from different angles, but with similar aims. If these data and approaches are combined, then the returns may be even greater for all parties.

An example case involves the work on the tumour suppressor gene and protein, p53 (Culotta and Koshland,

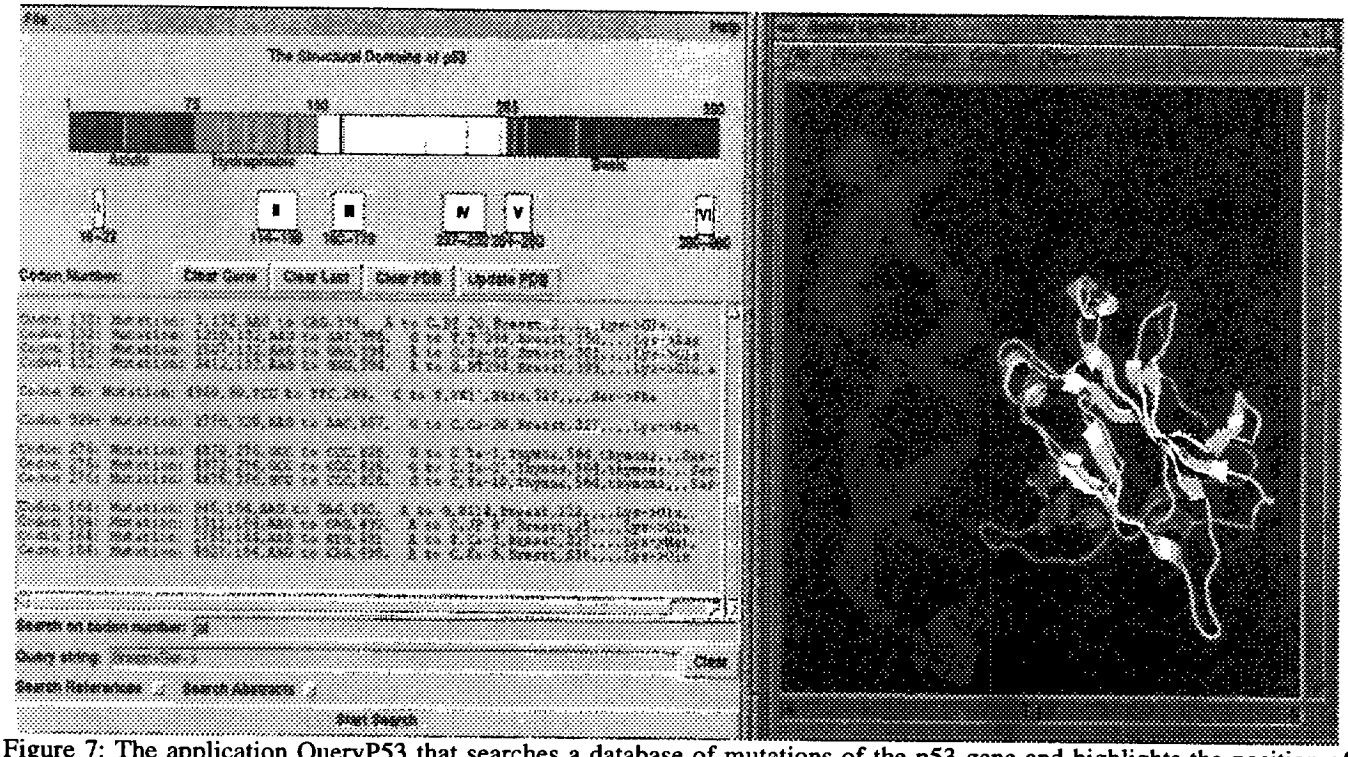

Figure 7: The application QueryP53 that searches a database of mutations of the p53 gene and highlights the position of mutations on both the 3-D structure of the p53 protein binding to DNA (right) and a linear schematic of the gene (left).

1993; Wong and Gruber, 1994). Human mutations databases are destined to become evermore important as technology allows the rapid screening of variants (Schumaker, 1996). It is observed that the p53 tumour suppressor gene is mutated in over half the human tumours reported (Hollstein, et al., 1991; Levine, Momand and Finlay, 1991; Greenblat, et al., 1994). Because of the obvious importance of p53 in cancer, many groups are working on many different aspects of p53. A vital piece of work has been the collation of a database containing over 6000 entries describing mutations of the p53 gene (Hollstein, et al., 1995). This database includes information on the nature and position of the base, triplet and amino acid mutation that has occurred, plus information on the cancer type that was observed, cell line details and relevant environmental aspects (e.g. was the subject a smoker). Other groups have worked on resolving the structure of p53 and there are now crystal structures of p53 binding to DNA available in the public domain. If the information from these separate resources can be combined, a precedent is established for how other groups may also look at data integration within their field.

To this end, the application QueryP53 (see Figure 7) was developed as a demonstration program that integrated a search program on the database of the p53 gene mutations with the 3-D structure of the p53 protein binding to DNA. Initially the user is shown a linear colour-coded schematic of the gene and two text boxes, into one of which search terms may be entered. The position of mutations corresponding to successful search terms are then highlighted along the length of the gene. These lines may be "clicked on" causing full information about the mutation to be displayed within the main text window. This gene view of the problem is the one more familiar to geneticists. However, this linear gene display is lacking in structural detail pertaining to the p53 protein. The application allows the user to display the 3-D structure of the published crystallographic structure of p53 binding to DNA (Cho, et al., 1994) which is colour-coded in the same manner as the linear gene view. As searches are executed, the position of mutations are shown on the linear p53 gene and may also be highlighted on the p53 protein structure, shown in the molecular structure viewer, if that mutation effects a triplet that is expressed as an amino acid in the protein.

Such an approach may allow one to infer how a mutation may effect the functioning of the protein. Mutations that are seen to occur within the body of the protein probably effect the folding of the protein and hence its function, while those occurring on loops interacting with the DNA probably effect the reading of the DNA. A screen snap shot of the application is shown in Figure 7.

Thus the mutation data from the database is enriched by integrating it with the 3-D structural information and offers new insight into how the mutations may effect the protein's function. The molecular structure becomes a platform upon which data may be projected that relates absolutely to the context of the data and this is a powerful paradigm; extra information has been added to the visual display simply by the way it has been presented.

Discussion

The visualisation techniques and tools described in this paper aim to provide alternative techniques with which to understand and manipulate data. An interesting concept that may be readily implemented due to the widget approach is to combine the techniques described above. Thus semantic lenses and filters could be used on information murals; a 3-D semantic lens connected to the p53 mutation database could be used on the 3-D molecular structure of p53; the information mural approach could be used within the hyperbolic projection viewer. Such functionality implies that the interface for any widget must include the means for it to communicate with other widgets while at the same time balancing its generic and specific nature.

The widget and object-oriented approach is a powerful paradigm in developing new and novel applications from existing components which may be plugged together and applied to a large number of varied problems. Development time may then be spent on analysing the data, rather than writing monolithic applications from first principles. With the advent of Java, which removes the platform dependence of compiled code, and organisations such as the bioWidget Consortium, a plethora of quality and reliable tools and widgets should become available that will prove an invaluable arsenal in the scientists tool kit for analysing and manipulating data. Furthermore with the availability of technologies for the provision of distributed objects, databases on remote servers are now as readily accessible as local objects. Data becomes more freely accessible and negates the necessity to download large databases from servers as flat files with the subsequent need to then develop code to parse them.

Conclusion

Scientific visualisation is more than just "pretty graphics". If visualisation is to be useful it must shed new light on data and lead to previously unknown, but valid and useful conclusions, provide means to manipulate and browse data more easily, or provide intuitive user interfaces that increase productivity.

The demonstration techniques and applications discussed above are intended to show that there is already a large and rich body of work that the relatively new science of bioinformatics may adapt and utilise on the flow of data that is now emerging. Bioinformatics is not the first discipline to be struck by a tidal wave of information and it can only profit by learning from other fields that have encountered such situations already.

Acknowledgments

The authors would like to thank; Timothy Slidel, Jeroen Coppieters and Christopher Dodge of the EBI for their invaluable support in implementing the CORBA and IDL interfaces necessary for access to the taxonomy database used by the hyperbolic projection viewer; Pam Bremer and Huruna Cofer of SGI for their collaboration on the MapViz application and finally members of the bioWidget Consortium, and in particular Gregg Helt, for their feedback and discussion on the hyperbolic projection widget.

References

Bier, E. A., M. C. Stone, K. Pier, W. Buxton and T. D. DeRose; 1993; Toolglass and Magic Lenses: The See-Through Interface. In *Proceedings of SIGGRAPH '93* (Annaheim, CA, August), *Computer Graphics* Annual Conference Series, ACM, 1993. pp.73-80.

bioWidget Consortium; 1996; Home page; http://goodman.jax.org/projects/biowidget/

Cho, Y., S. Gorina, P. D. Jeffrey and N. P. Pavletich; 1994; Crystal Structure of a p53 Tumour Suppressor-DNA Complex: Understanding Tumorigenic Mutations; *Science* 265:346.

Culotta, E. and D. E. Koshland; 1993; *Science* 262:1958-1959.

Crabtree, J., S. Fisher, G. Helt and M. Gibson; 1996; bioWidget Core Architecture. Draft API; http://agave.humgen.upenn.edu/bioWidgetsJava/draft_api.html

Greenblat, M. S., W. P. Bennett, M. Hollstein and C. C. Harris; 1994; *Cancer Res.* 54:4855.

Goodman, N., S. Rozen and L. D. Stein; 1995; The importance of standards and componentry in meeting the genome informatics challenges of the next five years. In *Proceedings of the Meeting on the Interconnection of Molecular Biology Databases*, July 1995, Cambridge, UK. http://www.ai.sri.com/people/pkarp/mimbd/95/abstracts/goodman.html

Hollstein, M., D. Sidransky, B. Vogelstein and C. C. Harris; 1991; p53 mutations in human cancers; *Science* 253:49-53.

Hollstein, M., C. Rice, M. S. Greenblatt, T. Soussi, R. Fuchs, T. Sorlie, E. Hovig, B. Smith-Sorensen, R. Montesano and C. C. Harris; 1995; Database of p53 Somatic Mutations in Human Tumours and Cell Lines; Submitted to *Nucleic Acids Res.*

Jerding, D. J. and J. T. Stasko; 1995a; The Information Mural: A technique for displaying and navigating large information spaces; In *Proceedings of the IEEE Visualixation '95 Symposium on Information Visualization*, pp. 43-50, Atlanta, GA, October 1995.

Jerding, D. J. and J. T. Stasko; 1995b; Using Information Murals in visualization applications; In *Proceedings of the*

1995 Symposium on User Interface Software and Technology (Demonstration), pp. 73-74, Pittsburgh, PA, November 1995.

Jerding, D. F. and J. T. Stasko. 1996; The Information Mural; Technical Report GIT-GVU-96-xx, Graphics, Visualization, and Usability Center, Georgia Institute of Technology.

Lamping, J. and R. Rao; 1994; Laying Out and Visualizing Large Trees using a Hyperbolic Space; *In Proceedings of the ACM Symposium on User Interface Software and Technology*. November 1994. ACM Press, pp. 13-14.

Lamping, J., R. Rao and P. Pirolli; 1995; A Focus+Context Technique based on Hyperbolic Geometry for Visualizing Large Hierarchies. *In Proceedings of the ACM SIGCHI '95 Conference on Human Factors in Computer Systems*. May 1995.

Levine, A. J., J. Momand and C. A. Finlay; The p53 tumour supressor gene; *Nature* 351:453-456.

The Object Management Group and X/Open; 1992; The Common Object Request Broker: Architecture and Specification, Version 1.1. 1992. John Wiley and Sons.

Ousterhout, K. K.; 1994; Tcl and the Tk Toolkit; Addison-Wesley Publishing Company, Massachusetts.

Robinson, A. J.; 1996; The European Bioinformatics Institute's Visualisation Awareness Pages. http://industry.ebi.ac.uk/~alan/VisSupp/VisAware/

Schumaker, J. M.; 1996; A Chip-based Methodology for Identifying DNA Sequence Alterations; In *Proceedings of the Eighth International Genome Sequence and Analysis Conference*, October 5-8, Hilton Head, SC.

Searls, D. B.; 1995a; Sharable Interface Componentry. MIMBD'95 Position Paper. http://www.cbil.upenn.edu/~dsearls/MIMBD.html

Searls, D. B.; 1995b; bioTk: Componentry for Genome Informatics Graphical User Interfaces; *Gene* 163:1-16; http://www.cbil.upenn.edu/~dsearls/bioTk_paper/paper.html

Silicon Graphics Inc.; 1996; The MineSet data mining application; http://www.sgi.com/Products/software/MineSet

Stein, L. D., S. Rozen and N. Goodman; 1994; The case for componentry in genome information systems; In *Proceedings of the Meeting on the Interconnection of Molecular Biology Databases*, August 1994, Stanford, CA; http://www.ai.sri.com/people/pkarp/mimbd/94/abstracts/wi.html

Stone, M. C., K. Fishkin and E. A. Bier. The Movable Filter as a User Interface Tool. In *Proceedings of CHI '94*. (Boston, MA, April 24-28) ACM, New York. pp. 306-312.

Sun Microsystems Inc.; 1994; The Java Language. A White Paper. http://www.javasoft.com/

Wong, M. and J. Gruber; 1994; *J. Natl. Cancer Inst.* 86:177-182.